

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Collaborative System of Wildfire Detection Using Multiple Intelligent Wireless Sensors Networks

Thadeu Brito



Doctoral Program in Electrical and Computer Engineering

Supervisor: Paulo Costa

Co-supervisor: José Lima

Co-supervisor: Ana I. Pereira

September 26, 2025

Collaborative System of Wildfire Detection Using Multiple Intelligent Wireless Sensors Networks

Thadeu Brito

Doctoral Program in Electrical and Computer Engineering

Approved by:

President: Prof. Dr. José Nuno Moura Marques Fidalgo, Faculty of Engineering of the University of Porto, Portugal

Referee: Prof. Dr. Laura Belli, Department of Engineering and Architecture at the University of Parma, Italy

Referee: Prof. Dr. Raphaël Couturier, Computer Science Department of University Institute of Technology of North Franche-Comté of University of Franche-Comté, France

Referee: Prof. Dr. António Luís Gomes Valente, Department of Engineering of the University of Trás-os-Montes and Alto Douro, Portugal

Referee: Prof. Dr. Armando Jorge Miranda de Sousa, Faculty of Engineering of the University of Porto, Portugal

Supervisor: Prof. Dr. Paulo José Cerqueira Gomes da Costa, Faculty of Engineering of the University of Porto, Portugal

September 26, 2025

I dedicate this thesis to my mother, Mariza Cristina Silvia de Brito. This work is, in fact, a reflection of her, for without her, none of this would have been possible. Thank you, *MÃE*, for being the foundation of my life and making me who I am!

This simple work I dedicate to you is far from what I have received. This thesis is a small tribute to everything you gave me: the life.

Abstract

Wildfires significantly threaten ecosystems, economies, and human lives, particularly in Portugal, where climatic and geographic conditions exacerbate fire risks. This thesis presents the development of a novel, integrated wildfire monitoring system based on Wireless Sensor Networks (WSN), employing low-cost, scalable hardware and advanced Machine Learning (ML) techniques. The research addresses the urgent need for efficient, reliable, and sustainable solutions for wildfire detection and management, combining innovative technologies and practical implementations.

The proposed system integrates low-cost sensor modules, robust communication infrastructures using LoRaWAN, and a cloud-based data processing framework. The sensor modules, designed with temperature, humidity, and infrared sensors, were optimized for energy efficiency and long-term operation through the development of the DROWSIER filtering algorithm, achieving extended battery life. By implementing this filter in the firmware, the modules achieved 324 days of uninterrupted operation.

Controlled and supervised fire experiments were conducted to validate the system. The results indicate that the WSN reliably detects fire ignition events across different distances (approximately from 4m to 121 m) and multiple time intervals (ranging from 1 min to 31 min). Advanced ML models, including RWR, ARIMA, RNN, and TimesFM, were employed for anomaly detection and time-series forecasting. A weighted voting approach enhanced anomaly detection, minimizing false positives and increasing system reliability.

Over three years, the system was deployed and validated in Serra da Nogueira, north of Portugal, indicating its practical applicability and resilience under real-world conditions. The findings highlight the potential of integrating WSNs with ML to provide actionable information for firefighting teams, ultimately improving decision-making and resource allocation. Therefore, this work establishes a foundation for scalable, efficient, and sustainable wildfire monitoring systems, offering a valuable tool for mitigating the growing global threat of wildfires.

In addition, this collaborative system offers a significant step toward more proactive and adaptive wildfire risk management. The proposed system can enhance existing surveillance by integrating low-cost sensor modules that can dynamically update fire risk estimates in real-time. These modules, when deployed across forested regions, can transmit critical environmental data to identify potential fire ignitions. In this sense, they can improve early detection capabilities and complement traditional monitoring approaches.

Keywords: Wildfire Monitoring; Wireless Sensor Networks (WSN), LoRaWAN; Machine Learning and Anomaly Detection.

Resumo

Os incêndios florestais ameaçam significativamente ecossistemas, economias e vidas humanas, particularmente em Portugal, onde as condições climáticas e geográficas agravam os riscos de incêndio. Esta tese apresenta o desenvolvimento de um sistema integrado de monitoração de incêndios florestais baseado em Redes de Sensores Sem Fio (RSSF), ao aplicar hardware escalável e de baixo custo, além de técnicas avançadas de Aprendizado de Máquina (ML). O trabalho aborda a necessidade urgente de soluções eficientes, confiáveis e sustentáveis para a detecção de incêndios florestais, combinando tecnologias inovadoras e implementações práticas.

O sistema proposto integra módulos de baixo custo, infraestruturas de comunicação com LoRaWAN e processamento de dados baseado na nuvem. Os módulos de sensores, projetados com sensores de temperatura, humidade e infravermelho, foram otimizados para garantir eficiência energética e operação de longo prazo por meio do desenvolvimento do algoritmo de filtragem DROWSIER, o que resulta em maior vida útil da bateria. Ao implementar esse filtro no firmware, os módulos alcançaram 324 dias de operação ininterrupta.

Testes com fogos controlados e supervisionados foram conduzidos para validar o sistema. Os resultados indicam que o RSSF detecta de forma confiável eventos de ignição de fogo em diferentes distâncias (aproximadamente de 4 m a 121 m) e vários intervalos de tempo (de 1 min a 31 min). Modelos avançados de ML, incluindo RWR, ARIMA, RNN e TimesFM, foram aplicados para a detecção de anomalias e previsão de séries temporais. Uma abordagem de votação ponderada aprimorou a detecção de anomalias, minimizando falsos positivos e aumentando a confiabilidade do sistema.

Durante três anos, o sistema foi instalado e validado na Serra da Nogueira, demonstrando sua aplicabilidade prática e resiliência em condições reais. Esses resultados evidenciam o potencial de integrar RSSFs com ML para fornecer informações úteis às equipas de combate a incêndios, aprimorando a tomada de decisões e a alocação de recursos. Portanto, este trabalho estabelece um marco para sistemas de monitoração de incêndios florestais escaláveis, eficientes e sustentáveis, oferecendo uma ferramenta valiosa para mitigar a crescente ameaça global de incêndios florestais.

Além disso, este sistema colaborativo oferece um passo significativo em direção a uma gestão de risco de incêndio florestal mais proativa e adaptável. Este sistema pode aperfeiçoar a vigilância existente ao integrar módulos de sensores de baixo custo que podem atualizar dinamicamente estimativas de risco de incêndio em tempo real. Esses módulos, quando distribuídos em regiões florestais, podem transmitir informações para identificar potenciais pontos de ignição. E deste modo, podem melhorar as capacidades de detecção precoce e contribuir com abordagens tradicionais de monitoração.

Palavras-chave: Monitoração de incêndios florestais; Redes de sensores sem fio (RSSF), LoRaWAN; Aprendizado de Máquina e Detecção de Anomalias.

Acknowledgements

This thesis represents the culmination of years of research, hard work, and collaboration, and I am deeply grateful to the many individuals and organizations who supported and guided me throughout this journey.

First and foremost, I wish to express my heartfelt gratitude to my supervisors, Prof. Paulo Costa, Prof. José Lima, and Prof. Ana I. Pereira, for their invaluable guidance, patience, and encouragement. Your expertise and mentorship have been instrumental in shaping this work, and your support extended far beyond academic counsel, for which I am profoundly thankful. In addition to the academic and professional support I have received, I would like to reflect on the invaluable experiences during my Erasmus. My heartfelt thanks go to the University of Parma and the IoT Lab for providing an inspiring academic environment that greatly enriched my learning journey. I am particularly grateful to Prof. Gianluigi Ferrari, Prof. Luca Davoli, and Prof. Laura Belli for their guidance, support, and warm welcome during my time there. The time spent abroad enhanced my academic and professional skills and provided me with cherished memories and friendships that will last a lifetime.

I am deeply indebted to the Bragança district firefighters and civil protection teams, particularly Commander João Noel Afonso and Paulo Ferro, whose cooperation and expertise were crucial for the experimental stages of this work. I also extend my sincere thanks to the *Guarda Nacional Republicana* (GNR), *Instituto da Conservação da Natureza e das Florestas* (ICNF), and the *Comando Sub-Regional de Emergência e Proteção Civil das Terras de Trás-os-Montes - Autoridade Nacional de Emergência e Proteção Civil* (ANEPC) for their collaboration and unwavering support during this project.

I gratefully acknowledge the PROMOVE-Fundação "la Caixa". The Fundação para a Ciência e a Tecnologia (FCT) for financial support through national funds FCT/MCTES (PIDDAC) to CeDRI (UIDB/05757/2020 and UIDP/05757/2020) and SusTEC (LA/P/0007/2021). And also for supporting within the scope of the PhD Grant Reference SFRH/BD/08598/2020. Additionally, my sincere appreciation goes to the institutions that supported this project: the Faculty of Engineering of the University of Porto (FEUP), the Polytechnic Institute of Bragança (IPB), Research Centre in Digitalization and Intelligent Robotics (CeDRI), Mountain Research Center (CIMO), and Instituto de Engenharia de Sistemas e Computadores, Tecnologia e Ciência (INESC TEC).

I sincerely thank my colleagues and collaborators in the SAFE project. Your contributions, ideas, and partnerships enriched this research and made the challenges more rewarding. A special thanks to Prof. José Rufino, João Mendes, Beatriz Flámia, Matheus Zorawski, and David Valente.

A special acknowledgment goes to Adriano Silva, Leonardo Furst, Luis Piardi, and Tiago Franco, whose dedication and assistance were crucial during the maintenance of the modules and the battery replacements. Your help was instrumental in overcoming the challenges of this work, and I am profoundly grateful for your support.

I would like to extend my heartfelt gratitude to the many friends I have been fortunate to gain throughout these years. The list of names is long, and while I cannot mention everyone individually, please know that your presence in my life has been deeply appreciated. Thank you for the countless moments of joy, reflection, shared challenges, and unforgettable experiences. From hours of leisure and study sessions to cherished memories – good and bad – your companionship has been an invaluable part of this journey.

Finally, my deepest gratitude and love go to my girlfriend, Paula Remor. Paula, your unwavering support, patience, and encouragement have been my anchor throughout this journey. You have been my safe harbor in moments of doubt and my source of endless inspiration daily. Thank you for being my companion, confidante, and most of all, for listening to me.

Thadeu Brito

*“Sempre faça tudo com muito amor e com muita fé em Deus, que um dia você chega lá.
De alguma maneira você chega lá.”*

Ayrton Senna.

Contents

1	Introduction	1
1.1	Contextualization and Motivation	2
1.1.1	The SAFe Project	5
1.2	Research Questions and Hypothesis	6
1.3	Research Methodology	10
1.3.1	Definition of Research Themes	12
1.4	Structure of the document	13
2	Overview of WSN Monitoring Systems	16
2.1	General Contextualization of WSN	16
2.2	Related Work on WSN-Based Monitoring Systems	18
2.2.1	WSN for wildfire	19
2.2.2	Summary of WSN-Based Monitoring Systems	21
2.3	WSN System Architecture	21
2.4	Final Considerations	29
3	The sensor nodes	30
3.1	General Contextualization of Fire Sensors	30
3.2	State of the Art: Fire Sensors, Communication and Microcontrollers	33
3.2.1	Fire sensors: Types and operating principles	40
3.2.2	Microcontrollers and Communications	45
3.2.3	Summary Fire Sensors, Microcontrollers and Communication	47
3.3	5FTH's Design and Assembly	48
3.3.1	Integration of low-cost sensors	49
3.3.2	Development of wireless detection modules	57
3.3.3	Assembly of wireless modules	63
3.3.4	Firmware description	68
3.3.5	Message Payload to LoRaWAN Gateway	73
3.4	Module Sensors' Results	74
3.4.1	Testing WSN's battery consumption	76
3.4.2	Testing WSN's communication	81
3.5	Discussion of Nodes' Results	84

3.6	Final Considerations	85
4	An Integrated Approach to Data Acquisition, Storage and Visualization Systems	87
4.1	General Contextualization of Systems	88
4.2	Bibliographic Review and Related Works: IoT for Monitoring Systems	90
4.2.1	Related Work on Key Components of Environmental Monitoring Systems	93
4.2.2	Summary of Key Components of Environmental Monitoring Systems . .	98
4.3	Development of the Monitoring System Architecture	99
4.3.1	Selection of Data Acquisition Tools	101
4.3.2	Development of Process Management Tools	103
4.3.3	Integration of Dashboard Visualization Tools	107
4.3.4	Security and Scalability Considerations	108
4.4	System Architecture and Implementation	109
4.4.1	LoRaWAN Configuration on Modules and TTN.	110
4.4.2	Node-RED Configuration	111
4.4.3	InfluxDB Configuration	117
4.4.4	Grafana Configuration	119
4.4.5	Docker Implementation	122
4.5	Discussion and Final Considerations	124
5	Data Analysis and Machine Learning	127
5.1	General Contextualization of Machine Learning Applications for WSN	128
5.2	Related Works: Machine Learning applied in WSN data	130
5.2.1	Data Structure and Description in WSNs	131
5.2.2	Exploratory Data Analysis in WSN Data	133
5.2.3	Machine Learning Models for WSN Data Analysis	135
5.2.4	Summary of Data Structure and Descriptions, Exploratory Data Analysis and Machine Learning Models	141
5.3	Datasets	142
5.3.1	Testing WSN at Low TCD Region - Dataset 1	143
5.3.2	Testing WSN at High TCD Region - Dataset 2	145
5.3.3	Testing WSN at High TCD Region - Dataset 3	149
5.3.4	Testing WSN at Low TCD Region - Dataset 4	150
5.3.5	Testing WSN at Low TCD Region - Dataset 5	152
5.4	Exploratory Data Analysis	153
5.4.1	Dataset 1 - Description and Analysis	155
5.4.2	Dataset 2 - Description and Analysis	165
5.4.3	Dataset 3 - Description and Analysis	177
5.4.4	Dataset 4 - Description and Analysis	186
5.4.5	Dataset 5 - Description and Analysis	192
5.5	Data Processing	198

5.5.1	Comparative Analysis of Module Versions	199
5.5.2	DROWSIER Filter	204
5.5.3	Temporal Data Analysis	207
5.6	Machine Learning Models	211
5.6.1	Rolling Window Regression	212
5.6.2	ARIMA	219
5.6.3	RNN	226
5.6.4	TimesFM	233
5.7	Combination of Techniques	240
5.7.1	Low TCD Scenario: Combining Techniques	241
5.7.2	High TCD Scenario: Combining Techniques	243
5.8	Testing and Validating ML Strategies for Forest Fire Scenarios	244
5.8.1	Applying the approach in Dataset 3	245
5.8.2	Applying the approach in Dataset 4	250
5.8.3	Applying the approach in Dataset 5	254
5.9	Final considerations	259
6	Conclusions and Future Work	262
6.1	General Overview	262
6.2	Main Contributions and Outcomes of Research Questions and Hypotheses	263
6.3	Future Work	268
	References	270
A	Wireless Module Schematic, PCB Layout and Gerber view	291
A.1	Wireless Module Schematic Circuit	291
A.2	Wireless Module PCB Layout	293
A.3	Wireless Module - Gerber files view	293
B	Wireless Module Enclosure	296
C	Source Code Used in System Development	298
C.1	Code Implementations for TTN	298
C.1.1	Section 4.3.1: LoRaWAN Data Decoding on TTN	298
C.1.2	Section 4.4.1: TTN Parameter Definitions for Firmware	299
C.2	InfluxDB Codes for Read and Write Queries	299
C.2.1	Section 4.3.2.2: InfluxDB Write Query Example	299
C.2.2	Section 4.3.2.2: InfluxDB Read Query Example	300
C.3	Node-RED flows	300
C.3.1	Figure 4.3: Node-RED flow for processing data from the TTN MQTT topic.300	300
C.3.2	Figure 4.4: Node-RED flow for processing and storing signal quality data. 301	301
C.3.3	Figure 4.5: Node-RED flow to get data from TTN storage	302

C.3.4	Figure 4.6: Node-RED flow for periodic InfluxDB backup	304
C.3.5	Figure 4.7: A GUI to assist in the identification of modules' locations . .	304
C.3.6	Figure 4.9: Example of Node-RED flow for real-time data processing and analysis.	306
C.3.7	Accessing Data from InfluxDB - Bucket SAFe_project	307
C.3.8	Writing Data in InfluxDB - Bucket SAFe_project	308
C.3.9	Figure 4.14: System Architecture Containerization with Docker implemen- tation.	308
D	Supplementary Data Analysis	311
D.1	Trends Analysis for Dataset 1	311
D.2	Trends Analysis for Dataset 2	317
D.3	Linear Regression Approach	323
D.3.1	LR models for WSN in Low TCD	324
D.3.2	LR models for WSN in High TCD	327
D.4	RWR Predicted vs Actual Readings (Dataset 1)	330
D.5	RWR Predicted vs Actual Readings (Dataset 2)	332
D.6	ARIMA Predicted vs Actual Readings (Dataset 1)	334
D.7	ARIMA Predicted vs Actual Readings (Dataset 2)	335
D.8	RNN Predicted vs Actual Readings (Dataset 1)	337
D.9	RNN Predicted vs Actual Readings (Dataset 2)	339
D.10	TimesFM Predicted vs Actual Readings (Dataset 1)	341
D.11	TimesFM Predicted vs Actual Readings (Dataset 2)	343

List of Figures

1.1	Burnt areas over time in European countries (1980-2022). Data source: EFFIS Technical Report [10].	3
1.2	Essential elements for a forest fire detection system, emphasizing the integration of wireless sensor networks, scalability, efficiency, prevention, and safety enhancement.	7
2.1	Architecture of the proposed wildfire monitoring system. ① Monitored region, ② Sensor modules, ③ LoRaWAN data transmission, ④ LoRaWAN gateway, ⑤ Mobile internet, ⑥ Cloud server, ⑦ Control center, and ⑧ Fire ignition early attack [30].	24
3.1	List of all terms found for fire detection using sensors in the most cited literature from 2016 to 2020 after removing computer vision terms.	38
3.2	List of all terms found for fire detection using sensors in the most cited literature after classification.	39
3.3	Desing of the prototype module.	51
3.4	Final assembled prototype module.	52
3.5	Data collected of the prototype module after 24 hours at controlled environment.	53
3.6	Preparation of the prototype module, consumption measurement, and fixation on the trunk tree for testing in an outdoor environment.	54
3.7	Data collected of the prototype module after five days at outdoor environment.	55
3.8	Simplified module's layout [135].	57
3.9	Example of a circuit for read data from the BPV10NF sensor.	60
3.10	Transmission process of the RFM95/RFM96 applied in each module. Adapted from [143].	61
3.11	DC-DC converter diagram used in each module.	62
3.12	Developed sensor module 5FTH [29].	64
3.13	Developed sensor module 5FTH in the 3D printed case [135].	64
3.14	The PCB's 5FTH 2.0 in 3D simulated view and final assembled.	66
3.15	The 5FTH 2.0 in 3D preview assembly simulation and final assembled.	67
3.16	The module's fixing simulation to visually analyse the IR sensors' field of view.	68
3.17	An example of flowchart with main tasks that a microcontroller must perform to save battery power while performing data acquisition.	69

3.18	The flowchart of all the tasks representing the microcontroller's operations. . . .	70
3.19	Encoding and decoding procedures. Size of $S[i]$ is 16 bits whereas T (temperature) and H (humidity) are 8 bits. AUX is an auxiliary variable that was created with the 4 bits that remain free [145].	74
3.20	Example of fixation of the module outdoors.	76
3.21	The battery discharge graph over 44 days of a WSN with eight modules, a transmission rate of 60s, and no sleep mode.	77
3.22	The total amount of data sent and online days during the test of a WSN with eight modules, a transmission rate of 60s, and no sleep mode.	77
3.23	The battery discharge graph over 172 days of a WSN with eight modules, a transmission rate of 60 seconds, and sleep mode activated.	78
3.24	The total amount of data sent and online days during the test of a WSN with eight modules, a transmission rate of 60s, and sleep mode activated.	79
3.25	The battery discharge graph over 324 days of a WSN with twenty modules using the DROWSIER filter.	80
3.26	The total amount of data sent and online days during the DROWSIER's test in a WSN with twenty modules.	81
3.27	Fixation of the module outdoors close to the Gateway.	82
3.28	Time interval over 365 days of a module using the DROWSIER filter.	83
3.29	The histograms with the time intervals' distribution from different perspectives. .	84
4.1	Base System Architecture for Environmental Monitoring, derived from literature review.	100
4.2	Environmental Monitoring System Architecture Components and Integration. . .	109
4.3	Node-RED flow for processing data from the TTN MQTT topic. Available in Appendix C - Algorithm C.5.	111
4.4	Node-RED flow for processing and storing signal quality data. Available in Appendix C - Algorithm C.6.	112
4.5	Node-RED flow for GET data from TTN storage. Available in Appendix C - Algorithm C.7.	112
4.6	Node-RED flow for periodic InfluxDB backup. Available in Appendix C - Algorithm C.8.	113
4.7	Node-RED flow for managing and storing module coordinates. Available in Appendix C - Algorithm C.9.	114
4.8	GUI for storing module coordinates.	115
4.9	Example of Node-RED flow for real-time data processing and analysis. Available in Appendix C - Algorithm C.10.	116
4.10	Grafana Panels for Temperature, Humidity, IR Sensors, and Battery Levels. . . .	120
4.11	Grafana Panels for Status Indicators.	121
4.12	Grafana Map Panel with Sensor Module Locations.	121

4.13	Heat Maps for SNR and RSSI Metrics.	122
4.14	System Architecture Containerization with Docker implementation.	123
5.1	Placement of five sensor modules in the first scenario with low TCD.	144
5.2	Five sensor modules acquire data of fire ignition in the first scenario with low TCD.	144
5.3	Data obtained by 09-5FTH during WSE in the first scenario.	145
5.4	Placement of 10 sensor modules in the second scenario (with high TCD).	146
5.5	Night-time test conducted as part of the additional WSEs to evaluate sensor performance under low-light conditions.	146
5.6	The two versions of 5FTH modules being tested in the high TCD scenario.	147
5.7	Data obtained by 09-5FTH during the third WSE in a high TCD scenario.	148
5.8	Data obtained by 09-5FTH during night-time (WSE2) in the second scenario.	149
5.9	The scenario used for cheking the ML models and strategies with new data.	149
5.10	The scenario with fire spread during data collection for Dataset 4.	150
5.11	Spatial arrangement of the WSN modules across the test area in Dataset 4.	151
5.12	The scenario with fire spread during data collection for Dataset 5.	152
5.13	Spatial arrangement of the WSN modules across the test area in Dataset 5.	153
5.14	Temperature (a), Humidity (b), IR sensors 1-5 (c-g), and Battery (h) in Dataset 1 over the observation period.	156
5.15	Total number of entries for Temperature, Humidity, IR sensors (IR1-IR5), and Battery recorded for each module in the Dataset 1.	156
5.16	Histogram of Temperature (a), Humidity (b), and IR sensors 1-5 (c-g) in Dataset 1.	159
5.17	Boxplots illustrating the minimum, mean, maximum, and range for each variable (Temperature (a), Humidity (b), and IR sensors 1-5 (c-g)) in Dataset 1.	160
5.18	Correlation matrix of Dataset 1, illustrating the relationships among each variable.	161
5.19	Temperature trends in Dataset 1 over 6-hour, 12-hour, and 24-hour.	162
5.20	ACF and PACF plots for each variable (Temperature (a), Humidity (b), and IR sensors 1-5 (c-g)) in Dataset 1, with red dashed lines indicating significance thresholds for identifying statistically significant lags.	164
5.21	Temperature (a), Humidity (b), IR sensors 1-5 (c-g), and Battery (h) in Dataset 2 over the observation period.	166
5.22	Total number of entries for Temperature, Humidity, IR sensors (IR1-IR5), and Battery recorded for each module in the Dataset 2.	167
5.23	Daily sensor data entry counts for each day, illustrating the acquisition volume throughout the Dataset 2 collection period.	168
5.24	Histogram of Temperature (a), Humidity (b), and IR sensors 1-5 (c-g) in Dataset 2, which illustrate the data distribution and frequency over the observation period.	170
5.25	Boxplots illustrating the minimum, mean, maximum, and range for each variable (Temperature (a), Humidity (b), and IR sensors 1-5 (c-g)) in Dataset 2	172

5.26	Correlation matrix of Dataset 2, illustrating the relationships among Temperature, Humidity, and IR sensor values (IR1-IR5).	174
5.27	Temperature trends over different time intervals (6-hour, 12-hour, and 24-hour) in Dataset 2.	175
5.28	ACF and PACF plots for each variable (Temperature (a), Humidity (b), and IR sensors 1-5 (c-g)) in Dataset 2.	177
5.29	Temperature (a), Humidity (b), IR sensors 1-5 (c-g), and Battery (h) in Dataset 3 over the observation period.	179
5.30	Total number of entries for each module in the Dataset 3.	179
5.31	Daily sensor data entry counts for each day, illustrating the acquisition volume throughout the Dataset 3 collection period.	180
5.32	Histogram of Temperature (a), Humidity (b), and IR sensors 1-5 (c-g) in Dataset 3, which illustrate the data distribution and frequency over the observation period.	182
5.33	Boxplots illustrating the minimum, mean, maximum, and range for each variable (Temperature (a), Humidity (b), and IR sensors 1-5 (c-g)) in Dataset 3	184
5.34	Correlation matrix of Dataset 3, illustrating the relationships among Temperature, Humidity, and IR sensor values (IR1-IR5).	185
5.35	Temperature (a), Humidity (b), IR sensors 1-5 (c-g), and Battery (h) in Dataset 4 over the observation period.	187
5.36	Total number of entries for each module in the Dataset 4.	188
5.37	Histogram of Temperature (a), Humidity (b), and IR sensors 1-5 (c-g) in Dataset 4.	189
5.38	Boxplots illustrating the minimum, mean, maximum, and range for each variable (Temperature (a), Humidity (b), and IR sensors 1-5 (c-g)) in Dataset 4.	191
5.39	Correlation matrix of Dataset 4, illustrating the relationships among Temperature, Humidity, and IR sensor values (IR1-IR5).	191
5.40	Temperature (a), Humidity (b), IR sensors 1-5 (c-g), and Battery (h) in Dataset 5 over the observation period.	193
5.41	Total number of entries for each module in the Dataset 5.	194
5.42	Histogram of Temperature (a), Humidity (b), and IR sensors 1-5 (c-g) in Dataset 5, which illustrate the data distribution and frequency over the observation period.	195
5.43	Boxplots illustrating the minimum, mean, maximum, and range for each variable (Temperature (a), Humidity (b), and IR sensors 1-5 (c-g)) in Dataset 5	197
5.44	Correlation matrix of Dataset 5, illustrating the relationships among Temperature, Humidity, and IR sensor values (IR1-IR5).	197
5.45	Timeline of overlapping data collection for both module versions in Dataset 2.	199
5.46	Correlation matrix for Group 1 and Group 2, displaying the relationships between Battery, Humidity, Temperature, and the five IR sensor readings.	201
5.47	Correlation matrix for Group 1 and Group 2, displaying the relationships between all variables during WSE.	202

5.48	Graphical representation of the difference between actual and RWR Model's predicted variables' values in Dataset 1. The plots display the actual versus predicted values for Temperature (a), Humidity (b), and IR1 through IR5 (c)-(g).	215
5.49	Graphical representation of the difference between actual and RWR Model's predicted variables' values in Dataset 2 during first WSE. The plots display the actual versus predicted values for Temperature (a), Humidity (b), and IR1 through IR5 (c)-(g).	218
5.50	Graphical representation of the difference between actual and ARIMA predicted variables' values in Dataset 1. The plots display the actual versus predicted values for Temperature (a), Humidity (b), and IR1 through IR5 (c)-(g).	222
5.51	Graphical representation of the difference between actual and ARIMA predicted variables' values during the first WSE in Dataset 2. The plots display the actual versus predicted values for Temperature (a), Humidity (b), and IR1 through IR5 (c)-(g).	225
5.52	Graphical representation of the difference between actual and RNN predicted variables' values in Dataset 1. The plots display the actual versus predicted values for Temperature (a), Humidity (b), and IR1 through IR5 (c)-(g).	229
5.53	Graphical representation of the difference between actual and RNN predicted variables' values in Dataset 2. The plots display the actual versus predicted values for Temperature (a), Humidity (b), and IR1 through IR5 (c)-(g).	232
5.54	Graphical representation of the difference between actual and TimesFM predicted variables' values in Dataset 1. The plots display the actual versus predicted values for Temperature (a), Humidity (b), and IR1 through IR5 (c)-(g).	236
5.55	Graphical representation of the difference between actual and TimesFM predicted variables' values during first WSE in Dataset 2. The plots display the actual versus predicted values for Temperature (a), Humidity (b), and IR1 through IR5 (c)-(g).	239
5.56	Aggregated Anomaly Scores from Dataset 1 (Low TCD Scenario) Using Unweighted and Weighted Strategies.	242
5.57	Aggregated Anomaly Scores from Dataset 2 (High TCD Scenario) Using Unweighted and Weighted Strategies.	243
5.58	Aggregated Anomaly Scores from Dataset 3 (High TCD Scenario in Summer) Using Unweighted and Weighted Strategies.	249
5.59	Aggregated Anomaly Scores from Dataset 4 (Low TCD Scenario in Montesinho) Using Unweighted and Weighted Strategies.	253
5.60	Aggregated Anomaly Scores from Dataset 5 (Low TCD Scenario in Rebordão) Using Unweighted and Weighted Strategies.	258
6.1	Long-term validation of the proposed system in Serra da Nogueira. The plot shows the real-time forecasted and collected data on the Grafana dashboard.	267
A.1	Full Wireless module schematic.	292

A.2	PCB layout of the wireless module.	293
A.3	Gerber files view of the Wireless Module.	294
A.3	Gerber files view of the Wireless Module.	295
B.1	Drawing of the RND 455-01087 [288].	297
D.1	Humidity trends in Dataset 1 over 6-hour, 12-hour, and 24-hour.	312
D.2	IR1 sensor trends in Dataset 1 over 6-hour, 12-hour, and 24-hour.	313
D.3	IR2 sensor trends in Dataset 1 over 6-hour, 12-hour, and 24-hour.	314
D.4	IR3 sensor trends in Dataset 1 over 6-hour, 12-hour, and 24-hour.	315
D.5	IR4 sensor trends in Dataset 1 over 6-hour, 12-hour, and 24-hour.	316
D.6	IR5 trends in Dataset 1 over 6-hour, 12-hour, and 24-hour.	317
D.7	Humidity trends over different time intervals (6-hour, 12-hour, and 24-hour) in Dataset 2.	318
D.8	IR1 sensor trends over different time intervals (6-hour, 12-hour, and 24-hour) in Dataset 2.	319
D.9	IR2 sensor trends over different time intervals (6-hour, 12-hour, and 24-hour) in Dataset 2.	320
D.10	IR3 sensor trends over different time intervals (6-hour, 12-hour, and 24-hour) in Dataset 2.	321
D.11	IR4 sensor trends over different time intervals (6-hour, 12-hour, and 24-hour) in Dataset 2.	322
D.12	IR5 sensor trends over different time intervals (6-hour, 12-hour, and 24-hour) in Dataset 2.	323
D.13	Lagged feature generation using synthetic sinusoidal data, showing how past values are incorporated over time	324
D.14	Graphical representation of LR Model's prediction errors for each variable in Dataset 1. The plots display the error values for Temperature (a), Humidity (b), and IR1 through IR5 (c)-(g).	327
D.15	Graphical Representation of RL Model's prediction errors for each variable in Dataset 2. The plots display the error values for Temperature (a), Humidity (b), and IR1 through IR5 (c)-(g).	329
D.16	Graphical representation of RWR Model's Prediction variables' values in Dataset 1. The plots display the actual versus predicted values for Temperature (a), Humidity (b), and IR1 through IR5 (c)-(g).	331
D.17	Graphical representation of RWR Model's prediction variables' values in Dataset 2 during the WSE1. The plots display the actual versus predicted values for Temperature (a), Humidity (b), and IR1 through IR5 (c)-(g).	333
D.18	Graphical representation of ARIMA prediction variables' values in Dataset 1. The plots display the actual versus predicted values for Temperature (a), Humidity (b), and IR1 through IR5 (c)-(g).	335

D.19	Graphical representation of ARIMA prediction variables' values in Dataset 2. The plots display the actual versus predicted values for Temperature (a), Humidity (b), and IR1 through IR5 (c)-(g).	337
D.20	Graphical representation of RNN prediction variables' values in Dataset 1. The plots display the actual versus predicted values for Temperature (a), Humidity (b), and IR1 through IR5 (c)-(g).	339
D.21	Graphical representation of RNN prediction variables' values in Dataset 2 during first WSE. The plots display the actual versus predicted values for Temperature (a), Humidity (b), and IR1 through IR5 (c)-(g).	341
D.22	Graphical representation of TimesFM prediction variables' values in Dataset 1. The plots display the actual versus predicted values for Temperature (a), Humidity (b), and IR1 through IR5 (c)-(g).	343
D.23	Graphical representation of TimesFM prediction variables' values in Dataset 2. The plots display the actual versus predicted values for Temperature (a), Humidity (b), and IR1 through IR5 (c)-(g).	345

List of Tables

3.1	Count of papers addressing fire sensors in academic databases.	34
3.2	Number of papers found with some relation to fire sensors and systematic review.	35
3.3	Number of papers found with some relation to fire sensors and systematic review from 2016 to 2020.	35
3.4	List of studies in fire detection from 2016 to 2020 in the top five of most cited in academic source platforms per year.	37
3.5	Dataset's structure generated by a given sensor module stored on the server.	75
3.6	Example of the dataset stored with floating millisecond differences in time intervals.	83
4.1	LoRaWAN Configuration Table.	110
4.2	Buckets' structure with their respective measurements, tags, and fields.	118
5.1	Dataset's structure generated by a given module stored on the server associated to a WSEs.	143
5.2	Modules placement details and fire observation timing for Dataset 1.	145
5.3	Modules placement details, WSEs, and initial and ending time for Dataset 2.	147
5.4	Detailed labels, and initial and ending time for WSEs in Dataset 2.	148
5.5	Modules placement details and WSE timing for Dataset 3.	150
5.6	Modules placement details and WSE timing for Dataset 4.	151
5.7	Modules placement details and WSE timing for Dataset 5.	152
5.8	Range Minimum and Maximum of the RWR Models Metrics for WSE period in Dataset 1.	213
5.9	Range Minimum and Maximum of the RWR Models Metrics for the first WSE period in Dataset 2.	216
5.10	Range Minimum and Maximum of the ARIMA Models Metrics for WSE period in Dataset 1.	220
5.11	Range Minimum and Maximum of the ARIMA Models Metrics for the first WSE period in Dataset 2.	223
5.12	Best RNN hyperparameters for variables in Dataset 1.	227
5.13	Range Minimum and Maximum of RNN Models Metrics for WSE period in Dataset 1.	228
5.14	Best RNN hyperparameters for variables in Dataset 2.	230

5.15	Range Minimum and Maximum of the RNN Models Metrics for WSE period in Dataset 2.	231
5.16	Range Minimum and Maximum of the TimesFM Models Metrics for WSE period in Dataset 1.	235
5.17	Range Minimum and Maximum of the TimesFM Models Metrics for first WSE period in Dataset 2.	238
5.18	Range Minimum and Maximum of RWR Models Metrics for WSE in Dataset 3. .	246
5.19	Range Minimum and Maximum of ARIMA Models Metrics for WSE in Dataset 3.	246
5.20	Range Minimum and Maximum of RNN Models Metrics for WSE in Dataset 3. .	247
5.21	Range Minimum and Maximum of TimesFM Models Metrics for WSE in Dataset 3.	248
5.22	Range Minimum and Maximum of RWR Models Metrics for WSE in Dataset 4. .	250
5.23	Range Minimum and Maximum of ARIMA Models Metrics for WSE in Dataset 4.	251
5.24	Range Minimum and Maximum of RNN Models Metrics for WSE in Dataset 4. .	251
5.25	Range Minimum and Maximum of TimesFM Models Metrics for WSE in Dataset 4.	252
5.26	Range Minimum and Maximum of RWR Models Metrics for WSE in Dataset 5. .	255
5.27	Range Minimum and Maximum of ARIMA Models Metrics for WSE in Dataset 5.	256
5.28	Best RNN Hyperparameters for Variables in Dataset 5.	256
5.29	Range Minimum and Maximum of RNN Models Metrics for WSE in Dataset 5. .	257
5.30	Range Minimum and Maximum of TimesFM Models Metrics for WSE in Dataset 5.	257
D.1	Best LR Parameters and Performance Metrics for Each Variable in Dataset 1. . .	325
D.2	LR Model Performance Metrics for Pre-Fire Period in Dataset 1.	325
D.3	Best LR Parameters and Performance Metrics for Each Variable in Dataset 2. . .	328
D.4	LR Model Performance Metrics for Pre-Fire (first WSE) Period in Dataset 2. . .	328

Acronyms

ABP Activation By Personalization. 78, 97, 102, 299

ABS Acrylonitrile Butadiene Styrene. 66

ACF Autocorrelation Function. 154, 163, 175, 176, 212, 216, 219, 223, 323, 324, 327

ADC Analog-Digital Converter. 53, 74, 181, 194, 211, 318, 320, 321

ADF Augmented Dickey-Fuller. 154, 162, 163, 175, 185, 186, 192, 198, 208–210, 219

AES Advanced Encryption Standard. 97, 108

AI Artificial Intelligent. 5, 128, 131, 269

ANN Artificial Neural Networks. 135

API Application Programming Interface. 95, 96, 103–105, 107, 111–113, 119, 124, 307, 308

AppEUI Application Unique Identifier. 102

AppKey Application Key. 102

AppSKey Application Session Key. 102, 110

AR Autoregressive. 138, 140, 163, 176, 219

ARIMA AutoRegressive Integrated Moving Average. 14, 127, 128, 139, 141, 186, 211, 219–225, 227, 230, 234, 240, 241, 243, 244, 246–248, 251, 253–256, 258, 259, 266, 334–337

AUC Area Under the Curve. 136

BPN Back Propagation Network. 135

BW Bandwidth. 46, 118

CNN Convolutional Neural Network. 31, 136, 138, 140

CO Carbon Monoxide. 20, 22, 33, 42, 44, 136, 137

CO₂ Carbon Dioxide. 20, 22, 42, 44, 136

- CPS** Cyber-Physical Systems. 95
- CR** Coding Rate. 46
- CRC** Cyclic Redundancy Check. 46
- CSS** Chirp Spread Spectrum. 46
- DBS** Density-Based Scheme. 136
- DBSCAN** Density-Based Spatial Clustering of Applications with Noise. 138
- DevAddr** Device Address. 102, 110
- DevEUI** Device Unique Identifier. 102
- DLVQ** Dynamic Learning Vector Quantization. 135
- DNN** Deep Neural Network. 136
- DROWSIER** Data Re-send Only Warnings Sensing an Ignition Else Repose. 14, 72, 79–82, 84, 85, 178, 180, 186, 192, 198, 204, 206, 207, 210, 211, 245, 250, 254, 264
- DT** Decision Tree. 135–137
- EDA** Exploratory Data Analysis. 127, 130, 131, 133–135, 141, 153, 154, 175, 199, 211, 216, 265, 323, 324, 327
- EFFIS** European Forest Fire Information System. 3
- EIF** Extended Isolation Forest. 137
- ESTiG** School of Technology and Management of Bragança. 81
- EU** European Union. 3
- FHDR** Frame Header. 46
- FM** Foundation Model. 140
- FPGA** Field Programmable Gate Array. 268
- FRM Payload** Frame Payload. 46
- FS** Fire Status. 142, 158, 179, 188, 193, 212, 214, 242, 243, 330, 332, 334, 336, 337, 339, 342, 344
- GKE** Google Kubernetes Engine. 98
- GPS** Global Positioning System. 19, 31, 94, 105, 107, 115–118, 133

- GRNN** General Regression Neural Network. 135
- GRU** Gated Recurrent Unit. 136, 139
- GSM** Global System for Mobile Communications. 20, 21
- GUI** Graphical User Interface. 92, 103, 109, 111, 114, 115, 117
- HDFS** Hadoop Distributed File System. 95
- HTTP** Hypertext Transfer Protocol. 97, 103, 104, 111–113
- I/O** input/output. 58, 59, 62, 65, 66
- IAQ** Indoor Air Quality. 95
- IB** Information Bottleneck. 138
- ICNF** *Instituto da Conservação da Natureza e das Florestas*. 4
- ID** Identifier Device. 75, 102, 106, 111, 113–115, 117, 118, 142, 179, 188, 193
- IForest** Isolation Forest. 137, 141, 212, 214, 217, 219, 221–225, 227, 228, 231, 235, 238, 266
- IoT** Internet of Things. 5, 16, 19, 33, 46–49, 87, 89–91, 95–100, 102–104, 128, 130, 132
- IP** Internet Protocol. 46, 93, 100
- IPB** Polytechnic Institute of Bragança. 58, 64, 76, 78, 81, 124, 142, 143, 146, 149, 245
- IQR** Interquartile Range. 134, 180, 196, 197
- IR** Infrared. 4, 5, 33, 37, 42, 44, 45, 47–51, 53, 54, 56, 58–60, 63–65, 67–69, 73–75, 100, 106, 108, 110, 118, 119, 129, 142, 143, 145, 146, 148, 155–158, 161–163, 166–169, 171, 173, 175, 176, 180, 181, 184, 185, 188, 191–198, 200, 201, 203, 207–213, 215, 216, 218, 220, 222–224, 226–228, 230, 231, 233–235, 237, 238, 240–244, 246–249, 251–253, 255–260, 264, 266, 268, 313–316, 318–321, 323, 328, 332, 333, 337, 339, 341, 343, 345
- ISM** Industrial, Scientific, and Medical. 46, 47
- LLMs** Large Language Models. 140
- LoRa** Long Range. 5, 21, 26, 46–48, 58, 61, 71, 95
- LoRaWAN** Long Range Wide Area Network. 5, 21, 22, 26, 27, 29, 46, 47, 58, 60, 61, 68, 69, 71–73, 75, 76, 84, 85, 90, 93–98, 100–102, 105, 108–110, 112, 115, 116, 121, 125, 204, 207, 264, 265, 268
- LPWAN** Low Power Wide Area Network. 19, 26, 45, 46

- LR** Linear Regression. 131, 135, 137, 141, 206, 212, 213, 216, 323–325, 327, 328, 330
- LSTM** Long Short-Term Memory. 129, 136, 138, 139
- LSTNet-Attn** Long Short-Term Network with Attention. 138
- LTE** Long-Term Evolution. 27
- MA** Moving Average. 140, 205, 206, 219
- MAC** Medium Access Control. 46
- MAD** Mean Absolute Deviation. 136
- MAE** Mean Absolute Error. 136, 139, 141, 213, 216, 219, 220, 224, 226, 228, 231, 234, 235, 238, 246–248, 251, 252, 255, 325, 327, 328
- MAPE** Mean Absolute Percentage Error. 136, 141, 219, 220, 224, 226, 228, 231, 234, 235, 238, 247, 248, 251, 252, 256–258
- MedAE** Median Absolute Error. 213, 325, 327
- MHDR** Medium Access Control Header. 46
- MIC** Message Integrity Code. 46
- ML** Machine Learning. 9–11, 13, 14, 21, 28, 29, 85, 86, 90, 92, 96, 99, 101, 105, 110, 116–119, 126–131, 135–137, 141, 142, 145, 149, 150, 179, 198, 203, 211, 212, 245, 250, 253, 254, 259, 261, 263, 266–268
- MLP** Multilayer Perceptron. 135–137, 140
- MQTT** Message Queuing Telemetry Transport. 91, 93–95, 98, 100–102, 104, 106, 111, 112, 124, 265
- MSE** Mean Squared Error. 139, 141, 213, 219, 220, 226, 228, 234, 235, 248, 252, 256, 257, 325, 327
- MTS** Multiple Time Series. 138, 140
- MVIForest** Majority Voting Isolation Forest. 137
- NAS** Network-attached storage. 109, 114
- NLP** Natural Language Processing. 139, 140
- NwkSKey** Network Session Key. 102, 110
- OS** Operating Systems. 124

- OTAA** Over-The-Air-Activation. 97, 102
- PACF** Partial Autocorrelation Function. 154, 163, 175, 176, 212, 216, 219, 223, 324, 327
- PCB** Printed Circuit Board. 14, 60, 63–66, 68, 293
- PHDR** Physical Header. 46
- PHY Payload** Physical Payload. 46
- PIR** Passive Infrared. 51
- PISA** Prompt based tIme Series forecAsting. 139
- PNDFCI** *Plano Nacional de Defesa da Floresta Contra Incêndios*. 4
- RBFN** Radial Base Function Network. 135, 136
- ReLU** Rectified Linear Unit. 137
- RF** Random Forest. 137
- RMSE** Root Mean Squared Error. 135, 136, 139, 141, 213, 216, 219, 220, 224, 226, 228, 231, 234, 235, 238, 246–248, 251, 252, 255, 256, 325, 327, 328
- RNN** Recurrent Neural Networks. 14, 127–129, 136, 140, 141, 186, 211, 226–228, 230, 231, 234, 240, 241, 243, 244, 247, 248, 251–254, 256–259, 266, 337, 339, 341
- ROC** Receiver Operating Characteristic. 137, 139
- RSSI** Received Signal Strength Indicator. 21, 118, 121, 122
- RTC** Real-Time Clock. 50
- RWR** Rolling Window Regression. 14, 127, 128, 211–213, 215, 217, 219, 227, 230, 234, 240, 241, 243–246, 248, 250, 253–255, 258, 259, 266, 330, 332
- SAFe** Forest Monitoring and Alert System. 5, 6, 10, 11, 13, 22
- SARIMAX** Seasonal Autoregressive Integrated Moving Average Exogenous. 140
- SD** Secure Digital. 50, 52
- SF** Spreading Factor. 46, 61, 78, 80, 118
- SMS** Short Message Service. 21
- SNR** Signal-to-noise Ratio. 21, 118, 121, 122
- SPI** Serial Peripheral Interface. 58, 61

- SQL** Structured Query Language. 95, 97
- SSE** Sum of Squared Errors. 153
- STL** Seasonal-Trend Decomposition with Loess. 154, 162, 174, 175, 311, 317
- SVM** Support Vector Machine. 135–137
- TCD** Tree Cover Density. 128, 143, 145, 147, 149, 150, 152, 153, 155, 165, 167, 169, 177, 180, 186, 192, 212, 216, 220, 223, 224, 226–228, 230, 231, 234, 237, 238, 240–245, 250, 251, 253, 254, 257, 259–261, 319, 323, 324, 327, 328, 330, 332, 334, 336, 337, 339, 341, 342, 344
- Time-GAN** Time-series Generative Adversarial Networks. 138
- TimesFM** Time Series Foundation Model. 14, 127, 128, 140, 141, 211, 233–235, 237, 238, 240, 241, 243, 244, 248, 252–254, 257–259, 266, 341–344
- TLS** Transport Layer Security. 108
- TOST** Two One-Sided Tests. 202, 203
- TS-ICE** Time Series Individual Conditional Expectation. 138
- TS-LIME** Time Series Local Interpretable Model-agnostic Explanations. 138
- TS-SHAP** Time Series SHapley Additive exPlanations. 138
- TSDB** Time Series Database. 91, 92, 95, 101, 105, 107, 125
- TSMixup** Time Series Mixup. 139
- TTM** Tiny Time Mixer. 140
- TTN** The Things Network. 71, 72, 93, 94, 100–113, 117, 122, 124, 125, 265, 298, 299
- UAV** Unmanned Aerial Vehicle. 31
- UI** User Interface. 92
- USB** Universal Serial Bus. 52, 58
- UV** Ultraviolet. 49–51, 53, 55, 56
- UX** User Experience. 92
- VM** Virtual Machine. 110, 124, 125

WSE Wildfire Supervised Events. 142–151, 153, 155, 156, 167, 173, 179, 188, 191, 193, 197–199, 201–203, 212, 214–220, 222–226, 228–231, 233–235, 237–249, 251–256, 258, 259, 265, 266, 325–327, 330, 332, 334–337, 339, 341–345

WSN Wireless Sensor Networks. 4–10, 12–14, 16–29, 32, 33, 41, 43, 48, 49, 73, 75–80, 84–86, 92, 97, 100, 105–111, 114, 116–119, 121, 125, 127–138, 141–144, 146, 149, 151–153, 155, 158, 167, 177–179, 186, 192, 198, 204, 207, 211, 220, 223, 226, 227, 230, 234, 240, 242, 245, 250, 254, 259, 262–269

Chapter 1

Introduction

Forests are essential for our planet's environmental, social, and economic sustainability. They provide critical ecological services, such as preserving biodiversity, sequestering carbon, and regulating the climate. Additionally, dense forests serve as sources of renewable materials for bioenergy and other applications. However, forest ecosystems are increasingly threatened by wildfires' rising frequency and intensity, driven by climate change, land-use transformations, and human negligence.

Over the years, despite significant efforts by authorities to protect forested areas – through establishing protocols and measures aimed at curbing forest density decline – monitoring these areas presents considerable challenges. Factors such as mountainous terrain and irregular soil relief make comprehensive oversight difficult. In a nutshell, the authorities, unfortunately, cannot keep an eye on every piece of forest. Consequently, authorities face limitations in ensuring continuous surveillance of all forested regions.

In this context, leveraging emerging technological advancements for forest monitoring becomes crucial. Such innovations can enable early detection of wildfires, allowing firefighting teams to respond promptly, mitigate fire impacts, and prepare themselves based on real-time information. From this perspective, the objective of this thesis is to develop a forest monitoring system that enhances environmental surveillance and supports existing monitoring initiatives, contributing to the preservation of natural ecosystems.

This chapter is structured as follows:

- Section 1.1 – provides a global and regional overview of wildfires, their ecological and economic impacts, and the limitations of current detection systems, highlighting the need for innovative solutions; introduces the project, its objectives, and its technological framework, outlining how it forms the foundation for this research.
- Section 1.2 – defines the Research Questions guiding this study and presents the Hypothesis that integrates sensor modules and algorithms for wildfire detection.
- Section 1.3 – describes the modular approach adopted in this dissertation concerning the key components and themes of the research. The details include the core themes of this thesis,

which address the challenges of wildfire monitoring through technological and analytical advancements.

- Section 1.4 – summarizes the content and objectives of each chapter, providing a roadmap for the dissertation.

1.1 Contextualization and Motivation

Wildfires have historically shaped ecosystems and landscapes, playing a vital ecological role. While natural phenomena such as lightning and volcanic eruptions have traditionally driven these events, human activities and climate change have increasingly exacerbated their intensity and frequency. Rising global temperatures and prolonged fire seasons have created conditions more conducive to extreme wildfires, posing challenges even for fire-adapted species [1]. The growing severity of these events underscores the importance of integrating historical fire regimes into current forest management strategies to mitigate risks associated with wildfires, ultimately balancing ecological health with human safety [2]. Wildfires can be triggered by a combination of natural phenomena – lightning, volcanic eruptions, and spontaneous combustion – and human activities, including agricultural practices, unattended campfires, machinery operations, and deliberate acts. These ignition sources interact in complex dynamics, creating a multifaceted interplay of environmental and anthropogenic factors that influence wildfire behavior and outcomes [3].

In forested areas, wildfire intensity changes significantly. Crown fires, which engulf tree canopies, cause extensive damage by generating heavy smoke and affecting larger ecological areas. In contrast, surface fires may rejuvenate ecosystems by clearing dead material and promoting regrowth. Grasslands and shrublands experience faster-spreading fires due to the abundance of dry vegetation. Although generally less severe than crown fires, these events disrupt ecological balance while supporting fire-adapted species and landscape health. Agricultural areas are also prone to wildfires, often sparked by land-clearing practices or escaped controlled burns. While these fires can sometimes benefit soil fertility, they frequently result in uncontrolled destruction, threatening crops and nearby ecosystems [4, 5].

Recent statistics reveal alarming trends in wildfire activity. In the United States alone, approximately 3 Mha of land were burned by wildfires as of October 2020, with an apparent increase in both the number of fires and the area affected [6]. Canada has reported an average of over 8000 fires annually between 2001 and 2021, burning more than 2.1 Mha [7]. In Australia, 2019/20 was record temperatures - with 1.5°C above the historical mean - coinciding with devastating wildfires that consumed approximately 19 Mha [8]. The fires in Brazil represent a significant environmental and public health crisis. In August 2024, the country recorded approximately 11 Mha burned (of which 4 Mha were native forests), an increase of 116% compared to the same period of the previous year, with emphasis on the Amazon and Cerrado (Brazilian Savanna) biomes, which suffered the greatest losses of areas [9].

As wildfires intensify globally, Europe is no exception, according to **European Forest Fire Information System (EFFIS)**, wildfires have similarly become a pressing issue, exacerbated by rising temperatures and prolonged drought periods linked to climate change. While the Mediterranean region has traditionally experienced seasonal wildfires, recent trends show an increase in both frequency and severity across northern and central Europe, areas historically less prone to such events. This is evidenced by the recorded wildfires along the whole mainland, which in 26 of the 27 **European Union (EU)** countries, except Luxembourg, recorded any wildfire [10].

In 2022, the total burnt area across the **EU** reached 837 212 ha, a dramatic increase compared to 2021 when 449 342 ha were affected. This indicates a significant escalation in wildfire activity, driven by two major yearly peaks. The first occurred in March, characterized by a widespread occurrence of fires across multiple countries. The second, more devastating peak occurred in July, marked by fewer but larger fires that contributed substantially to the year's total damage. Notably, 350 437 ha of the total burnt area were located within Natura2000 sites, representing 44% of the burnt area in **EU** countries. This number is three times higher than the damage reported in 2021, underlining the growing threat to Europe's protected natural habitats. The impact was concentrated in three countries: Spain, Romania, and Portugal, which collectively accounted for three-quarters of the total damage to these environmentally sensitive areas [10].

Figure 1.1 illustrates the annual burnt areas from 1980 to 2022, highlighting trends in wildfire impact over time. Notably, Southern European countries such as Portugal, Spain, and Italy have consistently reported higher burnt areas, aligning with their Mediterranean climates and recurring drought conditions.

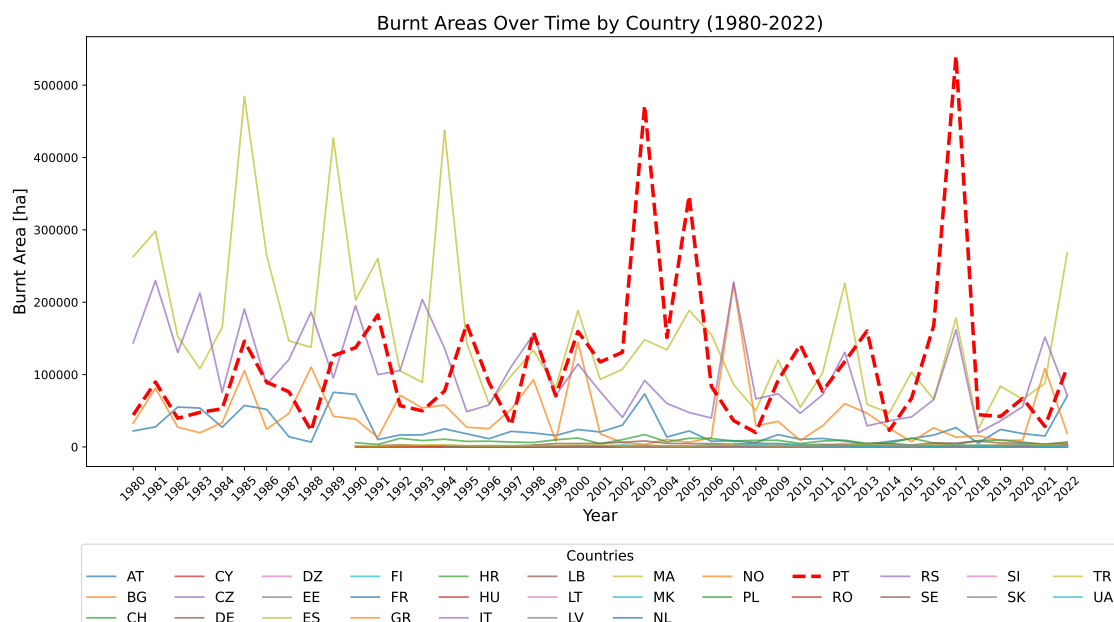


Figure 1.1: Burnt areas over time in European countries (1980-2022). Data source: EFFIS Technical Report [10].

This temporal analysis underscores the persistent vulnerability of certain regions and the growing impact of climate change. Peaks in the burnt area often correspond to extreme weather events, such as heatwaves, which exacerbate fire intensity and spread.

The Southern Member States, composed of Portugal, Spain, France, Italy, and Greece, over the last 43 years have suffered the impact of these events. In 2022 alone, these countries accounted for 51184 wildfires, with Portugal contributing 10390 fires, or 20%. Spain, Italy, France, and Greece accounted for 21%, 13%, 45%, and 2%, respectively. Over the period from 1980 to 2022, Portugal reported an average of 18741 fires annually, while Spain had 14093 and Italy 8641. In terms of burnt area, in 2022, a total of 538846 ha were affected across these nations, with Portugal representing 110097 (20% of the total), Spain 267947 (50%), and the remaining percentages distributed among Italy, France, and Greece. Historically, Portugal has seen a cumulative burnt area of nearly 5 Mha over the last 43 years, underscoring its vulnerability to wildfires and the pressing need for effective mitigation strategies [10].

The North and Center regions were the most affected by the total area burned in Portugal in 2022, accounting for 52499 ha and 52838 ha of burned area, respectively. Together, these two regions represented approximately 96% of the total area burned in the country. The land affected by these fires included 55309 ha of “Forest and Other Wooded Land” (50% of the total), 43761 ha of “Shrublands” (40%), and 11027 ha of “Agricultural Land” (10%). The peak of fire activity was concentrated in the summer, with July and August alone contributing to nearly 88% of the total burnt area. These two months represent extensive damage to forests, shrublands, and agricultural lands, reflecting the heightened risk during Portugal’s dry and hot summer season [10].

As forest fires are considered complex events in terms of causes, intensity, behavior, variability, resistance to control, size and severity, their early detection is critical [11]. In this context, and to prevent small fires from escalating into extreme events, the maximum time interval, from ignition to effective firefighters’ response, should not exceed six minutes [12]. To achieve this efficiency, in fighting forest fires within the estimated time, it is necessary to highlight the importance of identifying the ignition as quickly as possible together with the accuracy of the event location [13, 14]. In addition to early detection capability, estimating fire propagation direction and speed is also crucial in extinguishing fires [15]. For these purposes, there are several methods: human observation towers [16]; systems employing image acquisition [17, 18] and **Infrared (IR)** detectors [19]; satellite systems and images [20]; and **Wireless Sensor Networks (WSN)** [21].

In the *Plano Nacional de Defesa da Floresta Contra Incêndios (PNDFCI)* prepared by the *Instituto da Conservação da Natureza e das Florestas (ICNF)* [22], it establishes numerous strategies for combating forest fires. It is highlighted the existence of observation towers distributed along the forest reserves. However, the lack of reliability in observation towers that are managed by human operators and their performance, in terms of the ability to detect a rising fire, does not go beyond 10% in the most unfavorable areas (this value may rise to close to 60% in more favorable zones) [11, 23]. This reason has led to the testing of different technologies to improve both the detection capability and the use of fixed or mobile surveillance cameras in the spectrum that extends from visible to **IR**. Even so, the accuracy of these systems depends on the type of

topography, time of day, meteorological conditions (such as cloudiness index, light reflection and other environmental conditions) [24, 25].

On the other hand, with the development of new technologies within the scope of Industry 4.0, the forestry sector can be digitized in terms of solving issues such as rural fires [26, 27]. Integrating several systems, making them collaborative is a possible solution since there is still no type of operating system to alert wildfires based on these new approaches in Portuguese forests. For example, the service provided by [28] using **Internet of Things (IoT)** can alert forest ignitions in up to 4 minutes through modules that measure temperature, humidity, and CO_2 levels. The company's project is not to open access license, and hence, the company does not provide much information about the functioning of its system. That means any authority responsible for fight wildfires that intend to protect a given forest will need to sign the company's plan to use the service. In this sense, the literature does not present studies with developed techniques for forest fire alerts through low-cost sensors that transmit information through **Long Range (LoRa)** communication in Portuguese cases.

Given Portugal's high vulnerability to wildfires and the limitations of existing systems, there is an urgent need for a solution that combines cost-effectiveness, scalability, and rapid response capabilities. This thesis addresses this gap by proposing a novel open-source **WSN** solution designed to overcome the unique challenges of Portuguese forests. The system uses low-cost, open-source sensors strategically deployed in high-risk areas, ensuring accessibility and adaptability to diverse forest conditions. By enabling these sensors to work collaboratively, the system gathers critical environmental data, which is processed using algorithms to detect anomalies and predict fire-related events based on historical patterns. This approach integrates Industry 4.0 technologies, bridging the gap between traditional forestry practices and modern digital solutions, with a particular focus on rural regions.

1.1.1 The SAFe Project

The **Forest Monitoring and Alert System (SAFe)** represents a pivotal advancement in mitigating the risk of wildfires in Portugal, particularly in high-risk regions like the Serra da Nogueira in Trás-os-Montes. Launched to address the limitations of traditional wildfire detection systems, **SAFe** integrates cutting-edge technologies to offer a comprehensive approach to forest monitoring. Its architecture combines **WSNs**, **Long Range Wide Area Network (LoRaWAN)** communication, and data analysis driven by **Artificial Intelligent (AI)** to provide real-time information into forest conditions [29].

At its core, the **SAFe** project implements two distinct types of sensor modules: the HTW and 5FTH modules. The HTW module is designed for expansive environmental monitoring and is equipped with advanced sensors to measure temperature, humidity, and other ecological variables. In contrast, the 5FTH module emphasizes cost-effectiveness and localized fire detection, featuring **IR** sensors, temperature, and humidity. Both modules use LoRaWAN technology to transmit data to a centralized control center, ensuring robust and energy-efficient communication [30].

The **SAFe** system operates on a three-stage process: data acquisition in the forest, data processing via cloud-based algorithms, and real-time alert generation for authorities. This structured approach enables early fire detection, efficient data management, and timely dissemination of information to firefighting teams and civil protection agencies. By leveraging optimized algorithms for data filtering and transmission, the project not only reduces energy consumption but also enhances the accuracy of fire detection alerts [31].

SAFe's comprehensive framework underscores its ambition to bridge the gap between traditional fire monitoring methods and modern, technology-driven solutions. The project represents a tangible step towards Forestry 4.0, where digitization and intelligent systems transform forest management. The innovative ideas and technologies developed through **SAFe** establish a foundation for this thesis. It builds upon its achievements to propose an open-source, scalable, low-cost wildfire detection system tailored to Portuguese forests.

1.2 Research Questions and Hypothesis

Digitization of Portuguese forests to identify fire ignition within a six-minute interval requires the collaboration of an entire system, not just modules inserted into a **WSN**. A complete infrastructure is essential, incorporating both hardware and software components and decision-making subsystems to support firefighting teams. This comprehensive approach ensures that the challenges associated with wildfire monitoring – such as scalability, efficiency, and prevention – are effectively addressed.

Given the challenges outlined in the contextualization, this research project aims to address the critical gaps in wildfire monitoring systems based on low costs and open source. To achieve this, the following Research Questions and the Hypothesis guide the development of the proposed solution.

As illustrated in Figure 1.2, the development of an effective forest fire detection system must integrate multiple essential components. Starting with the **WSN**, which is the system's backbone and enables real-time data collection and transmission. Scalability ensures that the system can be adapted to forests of different sizes and densities, while low-cost sensors make the solution accessible to regions with limited resources. Efficiency is critical for minimizing response times, and prevention strategies are central to reducing the likelihood of catastrophic events. Finally, safety enhancement underscores the system's role in protecting both natural ecosystems and human lives.

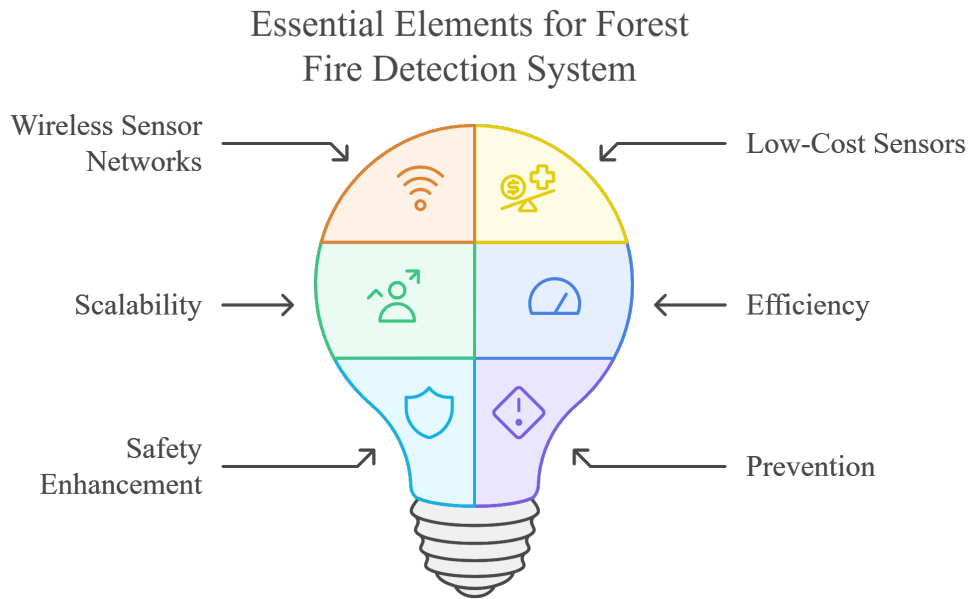


Figure 1.2: Essential elements for a forest fire detection system, emphasizing the integration of wireless sensor networks, scalability, efficiency, prevention, and safety enhancement.

To build a robust and effective forest fire detection system, it is essential to consider a combination of technological, logistical, and strategic factors. The elements highlighted in Figure 1.2 represent the foundational components required for a comprehensive wildfire monitoring and management approach. These elements are described in more detail below.

- **WSN:** The proposed system's backbone is deploying a **WSNs** capable of performing real-time data acquisition by utilizing distributed sensor modules to monitor the critical environmental parameters related to the fire ignitions. This decentralized architecture allows the system to operate effectively even in large, remote forested areas, where traditional surveillance methods often fail. Moreover, integrating **WSN** ensures continuous monitoring and early detection, which is crucial for minimizing fire response times.
- **Low-Cost Sensors:** Affordability is essential in making fire detection systems accessible, particularly in resource-constrained regions. The system democratizes access to advanced technology at a low cost while maintaining reliability and accuracy. This approach contrasts with proprietary systems that often require significant financial investment, making them impractical for widespread use in rural areas.
- **Scalability:** Wildfires occur in diverse terrains and climates, ranging from small woodlands to vast forested landscapes. Scalability is a key requirement for ensuring the system's adaptability to forests of different sizes and densities. The proposed system is designed to handle these variations by incorporating modular and flexible components that can be expanded or reduced based on the region's specific needs. This feature is especially vital for applying the system across Portuguese forests, where terrain and vegetation change widely.

- **Efficiency:** Efficiency is not limited to the operational speed of the system but also encompasses energy consumption, data transmission, and computational processing. The proposed system prioritizes low-power solutions to ensure sustainability in remote implementations, while efficient algorithms process data quickly to detect anomalies indicative of fire ignition. This dual focus on resource optimization and rapid response maximizes the system's effectiveness.
- **Safety Enhancement:** Beyond monitoring and prevention, the system aims to enhance the safety of both natural ecosystems and human populations. By providing precise, real-time data, the system improves the situational awareness of firefighting teams, enabling them to respond more effectively and safely to fire events. Furthermore, it protects biodiversity by minimizing the impact of wildfires on critical habitats.
- **Prevention:** Preventing data patterns is as vital as detecting them early. The proposed system integrates predictive models to identify anomalous data based on historical fire events and environmental conditions. By anticipating potential ignition points, the system can trigger preventative measures, such as sending alerts to authorities.

Integrating these essential elements addresses the critical gaps identified in wildfire detection systems and directly informs the research questions guiding this study. These questions focus on understanding the proposed solution's feasibility, implementation, and effectiveness in addressing the challenges described. The following discusses these questions, providing a roadmap for the research and its expected contributions.

Research Question 1

*Which types of sensors should be used in each module of the **WSN**, and will they be capable of identifying fire ignition within the critical six-minute interval?*

The human ability to detect biomass combustion relies on physical parameters such as light, temperature, smoke, and noise, which are characteristic byproducts of fire. Similarly, sensors designed to mimic these human senses can play a decisive role in detecting fire ignition. Identifying the most suitable types of sensors for this task requires an in-depth review of the literature on fire detection technologies and evaluating their precision, reliability, and energy efficiency.

This study aims to select which parameters and sensors can capture critical fire indicators. Additionally, it is important to check whether these sensors can achieve the level of responsiveness required to detect ignition within the six-minute threshold, ensuring early intervention and minimal fire spread.

Another vital consideration is the data sampling rate of these sensors. The chosen sensors must balance the need for high-frequency data acquisition with energy consumption constraints and data transmission bandwidth. Excessive data sampling can lead to higher power usage and communication delays, while insufficient sampling might compromise detection accuracy. Therefore, the

selected sensors must meet both operational and technical criteria, ensuring their effectiveness within the framework of a **WSN** designed for wildfire monitoring.

Research Question 2

*Are the raw data obtained by the **WSN**'s modules suitable for identify a forest ignition?*

After defining the appropriate set of sensors for each module, evaluating their effectiveness through real-world testing becomes indispensable. Prototype modules will be developed and deployed in controlled environments with supervised fire scenarios to replicate ignition events under several conditions. The analysis of raw data focuses on understanding the sensitivity and accuracy of the sensors in capturing early indicators of fire ignition.

This includes observing whether the data collected provides a clear and reliable signal of combustion events. Data visualization and exploratory analysis will be used to identify patterns and anomalies as indicative of fire ignition, ensuring that the system can distinguish between genuine fire events and false positives caused by environmental noise or non-fire phenomena. Furthermore, the quality and granularity of the raw data will determine its suitability for integration with advanced analytical models. This involves evaluating whether the raw data contains sufficient temporal and spatial resolution to support predictive algorithms and ensuring that the data is robust enough to perform under the constraints of limited bandwidth and power typical in **WSN** deployments.

Research Question 3

*Which approach would appropriate the necessities of analysing the multiple data sent by the **WSN**?*

The distribution of the **WSN** deployed in large forested areas generates vast and continuous data, often from multiple modules operating simultaneously. analysing this volume of information manually is impractical, mainly when the goal is to provide rapid and accurate fire detection. The complexity arises not only from the data volume but also from its heterogeneous nature. Each sensor module may measure different parameters, resulting in multidimensional datasets. Additionally, the data may contain noise, outliers, or overlapping signals from non-fire-related phenomena, which can complicate analysis. Addressing these challenges requires robust methods capable of preprocessing, filtering, and interpreting the data in real time. Therefore, developing automated data interpretation approaches becomes pivotal to ensure the system's effectiveness.

In this perspective, this research explores **Machine Learning (ML)** approaches, focusing on forecast-based algorithms. These algorithms analyse temporal patterns in the data, enabling the system to predict potential fire ignition points and issue alerts before an event escalates. Techniques such as anomaly detection, clustering, and time series analysis will be investigated to identify the most effective ways of simplifying and automating data interpretation.

Furthermore, integrating data visualization tools will enhance the accessibility of the processed information, allowing firefighting teams to understand and respond to alerts quickly. The goal is to

ensure that the proposed solution processes data efficiently and translates the results into actionable information, bridging the gap between raw data acquisition and decision-making.

Hypothesis

The previously proposed Research Questions have culminated into a Hypothesis that will formulate:

*With the integration of a **WSN** installed in Portuguese forests with **ML** algorithms, it will be possible to alert the firefighters the correct positioning for decision-making.*

This hypothesis synthesizes the core objective of this research: to design and validate an innovative wildfire detection system based on modern technological advancements. By combining the real-time data acquisition capabilities of a **WSN** with the predictive power of **ML** algorithms, the system aims to provide accurate and timely alerts to firefighting teams.

The hypothesis is supported by the Research Questions and builds on their outcomes:

- The selection of sensors (**Research Question 1**) ensures that the data necessary for early fire detection can be collected accurately.
- Evaluating raw data suitability (**Research Question 2**) guarantees that the system can effectively process and interpret relevant signals.
- Developing appropriate analytical methods (**Research Question 3**) enables the transformation of data volume into actionable information.

Testing this hypothesis involves the proposed system's iterative design, implementation, and validation in both controlled and real-world environments. Success will demonstrate that integrating these technologies can significantly enhance the decision-making process during wildfire events, reducing response times and minimizing environmental and societal impacts.

1.3 Research Methodology

The dissertation builds upon the continuing research initiated by the **SAFe** project. The **SAFe** project implements a set of innovative operations aimed at identifying forest ignition and monitoring fauna. The application of **SAFe**'s sensor modules for forest data acquisition is implemented in the Bragança region, specifically in the Serra da Nogueira territory. This project complements existing surveillance systems, such as cameras and observation towers, by providing firefighters and civil protection agencies with enhanced details and real-time information. While the **SAFe** project encompasses a broad range of objectives and technologies, this dissertation focuses on specific components of the project that align with its research objectives. These components include:

- **Development of Low-Cost Sensor Modules:** Extending the initial concept of **SAFe** modules, this research designs and validates cost-effective sensor modules (subsequently referred to as 5FTH) tailored for wildfire detection in high-risk forest areas.
- **Development of the SAFe System:** This includes the complete data flow pipeline, from data transmission and storage to visualization, ensuring efficient and reliable communication between the sensor network and guaranteeing system scalability.
- **Analysis of Low-Cost Module Data:** Comprehensive data analysis techniques are employed to evaluate the performance of the low-cost modules, assess their ability to detect anomalies, and integrate **ML** models for predictive data patterns.

To ensure a comprehensive and systematic approach, the dissertation adopts a modular structure, where each central theme is treated as a distinct and independent research task. This method is based on the instructional framework described in *Educational Research: Competencies for Analysis and Applications* by Gay and Mills [32]. According to the authors, splitting a research project into individual themes and manageable tasks allows for deeper focus and mastery of each component while maintaining coherence when integrating the results. Following this approach, each chapter of the dissertation addresses a specific theme – inspired by the topics mentioned above – interconnecting them within the broader goals of the **SAFe** project.

This separation ensures that each research component is developed in depth while facilitating the integration of findings into a unified system supporting wildfire monitoring and prediction. Moreover, the modular approach provides the dual benefits of treating each task with methodological rigor and ensuring the overall system functions cohesively. By structuring the research in this method, the dissertation can demonstrate each component's individual contributions while highlighting their interdependencies within the more extensive system.

The modularity of the research is directly reflected in the structure of the dissertation. Each chapter is dedicated to a specific theme or research task, following a consistent framework to ensure clarity and coherence throughout the document. Each chapter begins with a contextualization, introducing its theme and establishing the necessary background and relevance within the broader context of wildfire monitoring and the overall goals of the research. Following this, the objectives of the chapter are clearly defined, aligning them with the overarching aims of the thesis.

To provide a robust foundation for the research, each chapter includes a related works review, which examines existing literature, technologies, and methodologies relevant to the chapter's theme. This review ensures a comprehensive understanding of the current state of knowledge when the theme was developed and identifies gaps and opportunities addressed within the chapter. After, each chapter contains a methodology section, which details the specific methods and processes employed to achieve the chapter's objectives. This includes descriptions of experimental designs, frameworks, tools, and any theoretical approaches that guided the work. Subsequently, the development and application segment describes the practical implementation of the proposed methods or technologies.

Each chapter ends with a presentation of its results and their corresponding final considerations. In this sense, the chapters' final sections discuss the implications of the findings concerning the broader research goals and provide a summary of how the chapter contributes to the thesis.

Four core themes were dedicated to ensuring the research along the dissertation, following the structured plan mentioned before. Each theme represents a critical component of the proposed wildfire monitoring system, reflecting the logical progression from understanding and designing monitoring systems to their implementation and evaluation. The following subsection provides a detailed overview of these themes, highlighting their objectives and contributions to the thesis.

1.3.1 Definition of Research Themes

These themes represent the logical progression of the research, from understanding and designing monitoring systems to implementing and evaluating the complete solution.

Understanding and Structuring WSN Monitoring Systems

The first theme explores how **WSN**-based monitoring systems operate, analysing their robustness and limitations. This includes studying systems designed for diverse applications, such as wildfire detection, environmental monitoring, or industrial use. Through this analysis, the research identifies the key architectural components required for a functional monitoring system and establishes a framework for its operation. The outcome of this theme is a defined architecture for the wildfire monitoring system, detailing the essential elements to be developed in subsequent themes. This theme is closely related to the scope of Chapter 2, which provides an extensive review of existing **WSN** applications and technologies.

Low-Cost Sensor Modules

The second theme focuses on the design and implementation of low-cost sensor modules devoted to wildfire detection. This includes the selection and justification of sensors based on their ability to measure critical parameters. The development process also involves ensuring data quality during acquisition, designing the physical structure of the module, and evaluating its performance through transmission reliability and battery life tests. The results of this theme contribute to the creation of robust, energy-efficient modules suitable for deployment in real-world scenarios. This theme is comprehensively addressed in Chapter 3, which delves into the development, testing, and validation of the sensor modules.

Development of the Data Management System

The third theme addresses the design and implementation of the software infrastructure responsible for managing the data generated by the **WSN**. This system operates across the entire data flow pipeline, from receiving sensor data to storing, visualizing, and processing it. The goal is to create a reliable and scalable system that supports real-time operations while providing the necessary

infrastructure for advanced data analysis. The outputs include a fully functional system for data transmission, database management, and visualization tools that can facilitate decision-making processes. This theme is extensively explored in Chapter 4, which focuses on designing and implementing the software framework to serve as the technological pillar of the proposed wildfire monitoring solution.

ML for Anomaly Detection and Prediction

The final theme focuses on applying ML techniques to analyse the data collected by the WSN. This involves detecting anomalies in the data patterns and assessing the feasibility of real-time predictive models. The objective is to determine the viability of ML algorithms for deployment in an active WSN, enabling the system to provide timely and accurate forecasts of potential wildfire events. The findings of this theme inform the integration of predictive analytics into the monitoring system, ensuring its capability to operate as a proactive decision-support tool. This theme is the central focus of Chapter 5, which examines the application of ML models for analysing the data enhancing the system's predictive capabilities.

Integration of Themes

While each theme focuses on a specific component of the proposed system, their integration is essential for achieving the overarching goals of this research. The architectural framework defined in Chapter 2 guides the development of sensor modules in Chapter 3, which in turn generate the data managed by the system described in Chapter 4. Finally, the derived findings from ML models in Chapter 5 demonstrate the feasibility of deploying a fully functional wildfire monitoring system. Together, these interconnected themes provide a comprehensive solution for wildfire detection and prevention. The integration of these themes is presented in Chapter 6, which focuses on implementing and validating the complete wildfire monitoring system.

1.4 Structure of the document

This dissertation is organized into seven chapters, each one dedicated to a specific theme that contributes to developing and validating the proposed wildfire monitoring system. The document's structure reflects the modular approach adopted in the research, ensuring clarity and consistency throughout the work.

Chapter 1 introduces the global and regional context of wildfires, emphasizing the challenges they pose to ecological systems, economic stability, and public safety. This chapter presents the SAFe project as a response to these challenges, outlines the research questions and hypothesis guiding this work, and provides an overview of the methodology adopted. It establishes the motivation and framework for the dissertation, supporting the subsequent chapters with a clearly defined problem and set of objectives.

Chapter 2 extensively reviews the related works in WSN-based monitoring systems, particularly on wildfire detection. This chapter explores existing solutions and their limitations, such as energy efficiency, scalability, and data reliability. Furthermore, it proposes a high-level architecture for the wildfire monitoring system, offering a conceptual framework that incorporates related work information and identifies key innovation areas. This chapter lays the foundation for the detailed technical developments presented in subsequent chapters.

Chapter 3 focuses on the design, assembly, and evaluation of the sensor modules, which are the core components of the proposed WSN. It provides an in-depth discussion of the sensors selected for fire detection, the criteria for their choice, and their integration into the modules. The chapter also addresses key technical challenges, such as energy consumption and communication reliability, presenting innovative solutions – for instance, the Data Re-send Only Warnings Sensing an Ignition Else Repose (DROWSIER) filtering algorithm to extend battery life. Extensive performance testing, including real-world deployment scenarios, validates the effectiveness and efficiency of the modules.

Chapter 4 details the software infrastructure developed to support the wildfire monitoring system. It describes the implementation of data acquisition, storage, and visualization tools, all based on open-source technologies to ensure accessibility and replicability. The chapter also explores the integration of predictive analytics capabilities, highlighting how the system facilitates real-time monitoring and supports decision-making processes. The robust architecture presented in this chapter ensures the seamless data flow from sensor nodes to end-users.

Chapter 5 investigates the application of advanced ML techniques to analyse and interpret the data collected by the WSN. It examines four models, including AutoRegressive Integrated Moving Average (ARIMA), Rolling Window Regression (RWR), Recurrent Neural Networks (RNN), and Time Series Foundation Model (TimesFM), for their effectiveness in anomaly detection and time-series forecasting. The chapter emphasizes the development of a voting majority mechanism to consolidate outputs from multiple models, improving the system's accuracy and robustness in detecting fire ignition events. Supervised and controlled experiments with fires indicate the system's capability to process complex datasets and deliver actionable information in real-time.

Chapter 6 integrates the findings and developments from all previous chapters into a unified system. It validates the system through long-term deployment in real-world conditions, demonstrating its scalability, reliability, and practical applicability. This chapter also summarizes the research contributions, discusses the implications of the findings, and outlines future directions for advancing wildfire monitoring systems. It serves as a synthesis of the dissertation, connecting individual contributions to the broader goals of the research.

The document also includes appendices that provide supplementary materials and technical details, enriching the main content with additional insights and resources. Appendix A presents the detailed schematic circuit, Printed Circuit Board (PCB) layout, and Gerber files for the wireless module developed in Chapter 3. Appendix B contains the specifications and measurements of the enclosure used for the wireless modules. The enclosure, produced by RND Components, is essential for protecting the modules in harsh environmental conditions. Appendix C includes the

source code used during the system development. The code contains the technical implementations across the system, including data collection, processing, and integration of several technologies. Appendix D – provides additional plots and visualizations that support the data analysis discussed in the main chapters. These supplementary materials help the reader explore the trends, distributions, and patterns present in the datasets.

Chapter 2

Overview of WSN Monitoring Systems

This chapter establishes the groundwork for the thesis, bridging existing knowledge and the proposed system's innovative contributions. Therefore, it introduces the core concepts, challenges, and advancements in **WSN**-based monitoring systems, focusing on their application to wildfire detection. The structure of this chapter is as follows:

- Section 2.1 – discusses the evolution and benefits of **WSNs**, emphasizing their adaptability, scalability, and efficiency across several domains, particularly environmental monitoring and disaster management.
- Section 2.2 – reviews the related works in **WSN** applications, highlighting technological advancements, energy consumption challenges, and specific use cases in wildfire detection. Also, it identifies gaps and opportunities that guide the development of the proposed system.
- Section 2.3 – presents the architecture of the proposed wildfire monitoring system, detailing its key components, such as sensor modules, communication protocols, cloud infrastructure, and control center functionalities.
- Section 2.4 – reflects on the analysis and proposed architecture, outlining the essential steps that will guide the subsequent chapters on sensor module design, data infrastructure development, and advanced analysis tools.

2.1 General Contextualization of WSN

Advances in microelectronics have driven the development of compact, cost-effective, and energy-efficient electronic devices capable of environmental sensing and data processing. These advancements have been closely linked to the rapid evolution of the **IoT** and the growing demand for connectivity across several domains [33]. The rise of the **IoT** has enabled transformative applications such as autonomous vehicles [34], intelligent traffic systems [35], smart homes [36], and digital twins [37], among others. These applications gather data from devices or the environment, situating **WSNs** as one of the fastest-evolving technological fields due to their unique advantages.

Compared to wired networks, **WSNs** offer significant benefits. For instance, they require less time and cost for installation, can be deployed in dynamic environments or on moving objects to track parameters in real-time and allow for easier reconfiguration of the monitored area when necessary. This is because a **WSN** is generally composed of small, self-contained devices equipped with sensors that monitor and collect data from their environment [21].

These devices communicate wirelessly, enabling data collection, processing, and transmission. **WSNs** find applications in diverse fields, including environmental monitoring, industrial automation, healthcare, and smart infrastructure. Typically, they consist of distributed and dedicated sensors designed to capture and transmit environmental data to a central location for analysis [38].

Through integrating distributed sensors and advanced communication protocols, **WSNs** have become a mainstay of environmental monitoring and decision-support systems. These networks enable real-time data acquisition and processing across vast, often inaccessible areas, making them vital in addressing global challenges such as climate change, resource management, and disaster prevention. By offering innovative solutions to monitor and mitigate environmental risks, **WSNs** have proven their transformative potential [39].

In wildfire detection, **WSNs** signify a shift from traditional methods, such as human observation towers and satellite imagery, to automated systems capable of delivering real-time alerts with high precision. Applying low-cost, energy-efficient sensors and scalable network architectures, **WSNs** provide a solid foundation for early warning systems in fire-prone regions. These networks enable the seamless integration of diverse sensor modules, which monitor parameters essential for identifying early signs of fire ignition [40].

Despite their promise, deploying **WSNs** for wildfire detection presents significant challenges. Key issues include ensuring reliable communication across long distances, optimizing energy consumption for extended operations, and reducing false alarms caused by environmental noise. Furthermore, balancing cost-effectiveness and scalability remains a pressing concern, especially in resource-constrained regions with limited technical infrastructure. Overcoming these challenges is critical to ensuring the reliability and scalability of wildfire detection systems.

Given these complexities and challenges, understanding the state of the art in **WSN** applications and identifying best practices are essential steps in designing a robust monitoring system. Therefore, this chapter addresses these aspects with two objectives. First, it conducts a comprehensive review of related works, focusing on existing **WSN**-based monitoring systems across different application domains, with particular emphasis on wildfire detection. This review will provide an overview of the current technological landscape, highlighting advancements, limitations, and opportunities for improvement.

Second, based on the findings from the literature review, this chapter proposes a high-level architecture for the wildfire monitoring system. This architecture will serve as a foundational framework, defining the essential tools, components, and design considerations necessary to effectively develop and implement the proposed system. This chapter establishes the groundwork for the subsequent development and validation stages presented in later chapters by addressing these objectives.

2.2 Related Work on WSN-Based Monitoring Systems

WSNs have been applied across a wide range of domains due to their adaptability, scalability, and efficiency in collecting and transmitting data. Their ability to operate in remote or extreme environments, coupled with advancements in low-power and low-cost sensing technologies, has made WSN a catalyst for modern monitoring systems. Given the central role of WSNs, it is crucial to analyse existing works to understand their application scope, limitations, and opportunities for improvement. This review focuses on WSN-based monitoring systems, particularly those designed for extreme environments or disaster scenarios.

A WSN-based system for thermal energy harvesting in nuclear power plants is described in [41], highlighting its effectiveness in extreme environments. The designed prototypes exhibited radiation resistance and reliable performance in harvesting waste heat energy, providing a self-sufficient solution for critical monitoring. Despite challenges with the sensitivity of electronic circuits to radiation, the research establishes a significant advance in the use of wireless sensor networks in technology-intensive sectors.

In healthcare applications, [42] addresses the energy challenges of WSNs. Therefore, an innovative architecture is proposed to reduce energy consumption and extend the network lifetime. This proposal is based on replacing the traditional binary transmission with a quaternary scheme, which uses amplitude and phase modulation to transmit more information per symbol, reducing energy consumption by 41%. The simulations performed in MATLAB and SPICE indicated that the system can reduce total energy consumption compared to traditional protocols. The test environment included 200 sensors, simulating scenarios of hospitals and rehabilitation centers where accuracy and energy savings are critical. The main advantages include scalability and energy efficiency, especially in healthcare monitoring applications, such as detecting medical emergencies, tracking vital signs, and managing large volumes of data in real-time. On the other hand, the complexity of the quaternary system and the compatibility with existing binary systems may present some challenges.

Underground coal mine monitoring with WSNs is described in [43], indicating potential safety and operational efficiency improvements. The proposed system uses strategically positioned sensors to monitor gases, temperature, and pressure, with data transmission via efficient routing. The results indicate robustness, scalability, and energy efficiency, making WSNs a viable solution for operations in harsh environments such as coal mines. Although connectivity is limited when applying WSN in underground mines and initial costs due to the infrastructure, the study presents a promising approach for modernizing safety and productivity in underground mining.

With a browser-based interface and affordable hardware, an efficient and low-cost system for environmental monitoring in agriculture is presented by [44]. Implementing WSNs to provide real-time data makes the system ideal for smallholder farmers seeking to integrate precision agriculture practices. The modules send data via Wi-Fi transmissions, which may presents connectivity limitations in remote access environments. Even with this limitation, the system shows significant potential to improve agricultural productivity and promote sustainability.

The impact of **IoT** technologies on the modernization of the agricultural sector is highlighted in [45]. One of the points highlighted as a critical component of this transformation is the use of **WSNs**. This is because **WSNs** can perform real-time data collection, enabling efficient monitoring of essential parameters in agriculture. This information can be transmitted to farm management systems, where it is processed to optimize crop-related decisions. In addition, significant advantages can be gained in agriculture due to some characteristics of **WSN**, such as low energy consumption, connectivity in remote areas, and the ability to operate in mesh topologies. Connectivity technologies such as ZigBee, Bluetooth, and **Low Power Wide Area Network (LPWAN)** are widely used, providing flexibility and scalability for different agricultural scenarios.

A detailed review of the available technologies for wildfire detection, highlighting the importance of methods such as **WSN**, image processing, and deep learning, is performed in [46]. Solutions based on **WSN** and **IoT** offer a scalable and efficient approach for real-time monitoring, while deep learning technologies increase the accuracy of fire detection. However, challenges related to energy, cost, and the need for large datasets still persist, requiring continuous advancements to meet the demands of complex forest scenarios. The following subsection delves into research and developments specifically targeting the use of **WSNs** for wildfire detection.

2.2.1 WSN for wildfire

The application of **WSNs** in wildfire detection has attracted significant attention due to their potential to address limitations in traditional monitoring methods, such as human observation and satellite imaging. By enabling real-time data collection and transmission, **WSNs** offer a scalable and efficient solution for monitoring vast forested areas, even in remote or inaccessible regions. The ability to deploy low-cost, energy-efficient sensor nodes further enhances their suitability for wildfire monitoring, making them a viable choice for areas with limited resources or challenging terrains. Research in this field has explored several configurations, sensor types, and communication protocols to detect fires and monitor forests.

In [47], it is presented the development of a wildfire monitoring system using **WSNs**. The motes used in the system include temperature, relative humidity, and barometric pressure sensors, as well as an integrated **Global Positioning System (GPS)** unit for precise localization. The data collected by the motes are transmitted to a base station via radio frequency communication, which stores them in a MySQL database. During the tests, the motes were placed on elevated supports, and the tests were performed in controlled burns in low-density grasslands in California, with an estimated fuel load of one ton per acre. However, the motes face limitations and inadequate thermal protection for areas of high fuel density. Furthermore, the need for adjustments in the sampling frequency for different environmental conditions and battery life was not discussed throughout the work.

The application of **WSNs** for monitoring and early detection of forest fires in Turkey is explored in [48], highlighting the country's vulnerability to this type of disaster due to climatic and geographic conditions. **WSNs** are presented as an economical and technologically viable solution for remote areas, allowing for greater accuracy and coverage in measurements of critical parameters.

Aerial deployment and autonomous operation are points discussed in the paper, although there is no in-depth discussion of how to perform the procedure. The method of data transmission from the modules is also not discussed, and how challenges of high battery consumption could affect the autonomy of the modules and how to overcome this challenge.

A monitoring system for wildfire detection that combines WSNs and satellite data is presented in [49]. In this context, the paper proposes the combination of local sensor data with satellite monitoring and aerial patrolling to form an integrated detection model. With a focus on efficient communication and predictive models, the system offers higher accuracy and lower cost than traditional methods. The sensors are based on radio frequency communication. Although the study highlights the potential of WSNs as a crucial tool for forest fire prevention and management, it does not delve into the implementation details of the system, such as battery consumption, sensor models, or module development.

Intending to extend the useful life of WSN nodes, [50] proposes a wireless sensor node with solar panels. It is designed to detect forest fires early, using smoke, Carbon Monoxide (CO), methane, humidity, and temperature sensors. Powered by solar energy, the nodes connect to a base station through radio frequency communication and self-organizing networks, offering high sensitivity and accuracy for small fires. The tests carried out with burning paper indicated a detection rate of 80% at short range. On the other hand, the authors do not determine the capacity of the solar panel to charge the module under the shade of trees, which may harm the cost-benefit of manufacturing the modules.

Authors from [51] presents an integrated system of WSN and neural networks for real-time detection of forest fires, focusing on precisely identifying combustion phases. Smoke, Carbon Dioxide (CO₂), and temperature sensors increased accuracy, possibly increasing costs. Tests with woody biomass briquettes exhibited an accuracy of up to 82.5% with multiple inputs, highlighting the system's potential for applications in forest monitoring and fire risk management. Despite limitations during phase transitions, the system offers an economical and viable solution for environmental disaster control. On the other hand, battery consumption analysis is not mentioned, nor is the implementation of WSN in real-world monitoring situations.

Combining multiple sensors, thermal cameras, and wireless communication for forest monitoring, [52] describes a system based on WSN to detect fires, monitor wildlife, and combat human intrusions. Powered by solar energy, the system provides real-time Wi-Fi data and automated alerts, significantly improving efficiency and accuracy in forest management. Despite the potential of the proposed system, some interference and scalability limitations may restrict implementation, such as the use of Wi-Fi to transmit data. Furthermore, the paper does not indicate the system's implementations in real-world situations.

The paper [53] presents a forest fire detection system based on WSNs and Global System for Mobile Communications (GSM) modules, highlighting its potential to warn of fires in their early stages. The system combines smoke and fire sensors for data collection and uses a cluster topology to optimize communication and power consumption. Tests have shown that the system is reliable for application in remote areas, despite challenges related to precise localization and dependence on

GSM infrastructure. This is because the framework used in the work relies on data sent via **Short Message Service (SMS)**.

The **LoRaWAN**-based monitoring system presented in [54] addresses a viable solution for fire detection in rural areas, highlighting its energy efficiency and affordable cost. The development of the approach combines environmental sensors with **LoRa** communication, allowing real-time detection and rapid response to critical events. Despite challenges such as infrastructure dependence and limitations in extreme conditions, the system revealed high potential for applications in environmental monitoring and natural disaster mitigation. On the other hand, the tests described were for **Received Signal Strength Indicator (RSSI)** and **Signal-to-noise Ratio (SNR)** verification purposes with a **WSN** composed of four nodes. Therefore, the developed nodes do not have described behavior during the execution of fire tests.

2.2.2 Summary of WSN-Based Monitoring Systems

This section reviewed a range of applications of **WSNs**, including healthcare, agriculture, and environmental monitoring. The studies reveal how **WSNs** can provide scalable, efficient, and cost-effective solutions for real-time data collection and analysis. These systems leverage advancements in low-cost sensors and wireless communication technologies, enabling real-time data acquisition and analysis in challenging environments.

Particular attention was given to wildfire detection, in which **WSNs** offer promising advantages over traditional monitoring systems, such as real-time alerts, automation, and the ability to operate in remote areas. The reviewed works highlighted the use of diverse technologies, such as **LoRaWAN**, solar-powered nodes, and **ML** models, each addressing specific challenges in deployment, energy efficiency, and communication. While promising, the studies reviewed emphasize the persistent challenges in remote and resource-constrained settings. However, issues such as battery consumption, environmental interference, and validating system performance in real-world fire scenarios indicate areas for further research and development.

Considering these limitations, the proposed system aims to advance the capabilities of **WSNs**, offering a practical and scalable solution tailored to the unique challenges of wildfire-prone in Portugal. In this context, the following section focuses on conceptualizing the proposed wildfire monitoring system. To address these challenges and realize the potential of **WSNs** in wildfire detection and prevention, the next section focuses on conceptualizing the system architecture. This includes defining the core components and their interconnections, which will serve as the foundation for the proposed monitoring system.

2.3 WSN System Architecture

Based on the discussions from Chapter 1, which emphasized the critical need for innovative wildfire monitoring solutions, and the directions from the related works reviewed earlier, this section presents the architecture of the proposed wildfire monitoring system. The challenges highlighted – such as the limitations of traditional methods, energy efficiency concerns, and the need for real-time

data acquisition – are addressed through a meticulously designed system based on the **SAFe** project framework.

The architecture integrates advancements in **WSN** technology, as seen in the related works on environmental and disaster monitoring, the solution must adapt to the unique requirements of wildfire detection in Portugal. Considering the advancements from research on **LoRaWAN**, energy-efficient nodes, and multi-sensor networks, the architecture aims to overcome the limitations of existing systems and deliver a robust, real-time monitoring framework.

As highlighted in Chapter 1, wildfire-prone areas, such as those in Portugal, present significant challenges due to their topography, forest density, and climatic conditions. Therefore, rural and forested areas demand strategic sensor placement to ensure effective monitoring and timely anomaly detection. This process involves careful consideration of terrain features, vegetation type, and accessibility, as these factors influence the effectiveness and sustainability of the deployed modules. This step aligns with practices observed in the related works, where regional risk assessment guided sensor deployment for optimized coverage, ensuring a balance between comprehensive data acquisition and operational feasibility.

Considering that these areas often extend extensive kilometers, a significant number of modules must be deployed to ensure comprehensive coverage. This makes cost a critical factor, as large-scale deployment requires affordable modules to remain economically viable. Moreover, prioritizing low-cost modules enables broader adoption in resource-constrained regions, addressing scalability without compromising system performance. Beyond cost, modules must also be resilient to harsh environmental conditions, easy to maintain, and capable of transmitting data reliably over long distances. Each module must include essential components to guarantee data acquisition, robust communication, and long-lasting energy sources to support autonomous operation over extended periods. These features collectively ensure the system remains efficient, reliable, and sustainable in wildfire-prone rural and forested areas.

The sensor modules form the system's heart, embodying the low-cost, scalable, and energy-efficient design principles. Each module must include:

- **Multi-sensor integration:** Following the examples from the literature, it was seen that several types of sensors can be applied to detect fires, from sensors for temperature, humidity, gas detection (**CO**, **CO₂**, among others) to the flame sensors. Therefore, the monitoring can be deployed to detect fires in different modes, enabling comprehensive data acquisition for fire detection.
- **Energy-efficient operations:** Inspired by strategies in related works, in which the most consumption is due to lack of standby mode, the modules should be able to run in sleep cycles, event-driven data transmission, and pre-filtering algorithms to conserve energy while maintaining accuracy.

Robust communication is pivotal for reliable **WSN** operations. The challenging terrain and dense vegetation can significantly impact signal strength and network connectivity in wildfire-prone

areas. Therefore, it is essential to adopt resilient communication protocols to such environmental factors while ensuring energy efficiency. As seen in the previous section, authors employing similar methodologies have shown the effectiveness of multi-hop communication, mesh networks, and adaptive routing protocols in enhancing data transmission reliability over long distances. These technologies enable the seamless integration of numerous sensor modules, ensuring that critical data from all regions of the monitored area is transmitted to centralized systems for processing and analysis. By addressing the communication challenges inherent to these environments, the system can deliver timely and accurate information to decision-makers, supporting proactive and reactive measures in wildfire management.

Given that these zones are typically remote and far from urban areas or existing infrastructure, it is often unrealistic to expect sensor modules to transmit data directly to a central point in the city. This limitation necessitates alternative strategies to ensure reliable data transmission. One effective solution is strategically placing gateways within or near the monitored region. These gateways can serve as intermediary points, collecting data from the sensor network and forwarding it to central systems.

To bridge the gap to urban infrastructure, the gateways can be connected to the internet through mobile networks (e.g., 4G/5G) or wired connections, depending on the region's availability. This approach extends the network's communication range and ensures that data from even the most remote areas can be transmitted securely and efficiently. Moreover, these devices can be solar-powered to ensure operation in off-grid locations, minimizing maintenance requirements and enhancing sustainability. Integrating such infrastructure with the network aligns with the principles of scalability and modularity, enabling the system to adapt to multiple environmental conditions and deployment scales. By combining these strategies, the WSN can overcome the challenges of remoteness and limited connectivity.

The proposed system aims to provide or complement real-time information to authorities responsible for wildfire management, such as local fire departments, environmental agencies, and emergency response teams. To achieve this, it is essential to understand these authorities' workflows, tools, and preferences for monitoring and responding to fire-related events. This includes understanding how they currently visualize the region being monitored, such as through maps, dashboards, or field reports, and determining the most effective methods for delivering the data collected by the sensor modules.

Beyond data collection, it is equally important to synthesize the information to be displayed in the control center, ensuring that it is concise, actionable, and aligned with the authorities' decision-making needs. One practical approach is implementing processing and predictive analyses that transform raw data into actionable information. These analyses can help authorities identify trends, anticipate potential risks, and prioritize interventions, enhancing their ability to respond effectively to wildfire incidents.

To better understand the proposed system, Figure 2.1 illustrates the architecture of the proposed wildfire monitoring system, encapsulating its key components and their interconnections. The diagram provides a high-level overview, emphasizing the data flow from the sensor modules in

the monitored region to the control center and cloud server. It also highlights the integration of advanced communication protocols, real-time processing, and predictive analytics.

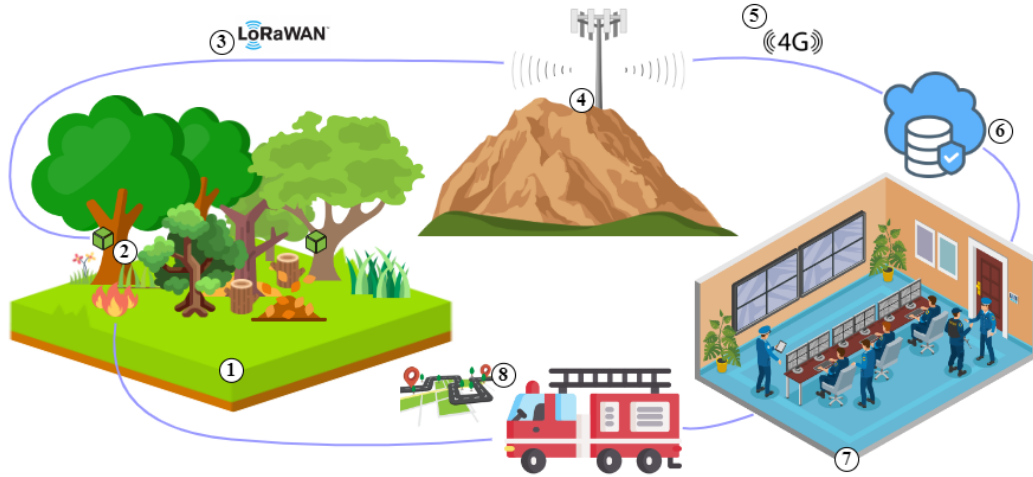


Figure 2.1: Architecture of the proposed wildfire monitoring system. ① Monitored region, ② Sensor modules, ③ LoRaWAN data transmission, ④ LoRaWAN gateway, ⑤ Mobile internet, ⑥ Cloud server, ⑦ Control center, and ⑧ Fire ignition early attack [30].

The proposed wildfire monitoring system has eight principal components that collectively ensure its functionality, efficiency, and adaptability to wildfire detection and management challenges:

- ① **Monitored region:** Represents the area where the **WSN** is deployed to collect data.
- ② **Sensor modules:** Indicates the real-time data acquisition performed by the **WSN**.
- ③ **LoRaWAN data transmission:** Symbolizes the process of transmitting the acquired data.
- ④ **LoRaWAN gateway:** Highlights the best practices for gateway positioning.
- ⑤ **Mobile internet:** Portray the connection between the gateway and the Cloud service.
- ⑥ **Cloud server:** Signifies the storage and processing of data.
- ⑦ **Control center:** Showcases the visualization and interpretation of the information.
- ⑧ **Fire ignition early attack:** Express the actions taken to counter wildfires proactively.

These components encompass all critical aspects of the system, from data acquisition and processing to communication and decision-making support. In the following, each component will be detailed, highlighting its purpose, design considerations, and role within the overall system architecture.

① Monitored region

The selection of the monitored region has a vital function in the effectiveness of wildfire detection systems. While the intuitive approach may be to spread sensor modules across the entire area of interest evenly, this strategy often proves inefficient. Such an approach results in several challenges, including the generation of a high volume of redundant data, increased efforts for module deployment and maintenance, and higher overall costs – even when using low-cost modules. These factors highlight the importance of adopting a more strategic method for sensor placement.

To address these challenges, the use of risk maps and forest density maps becomes essential. Risk maps identify areas prone to wildfire ignition based on historical data, climatic conditions, and topographical features [24, 55]. Similarly, forest density maps provide the distribution of biomass across the region, which directly correlates to a fire's potential intensity and propagation [56]. By combining these two maps, a hybrid approach can be employed to define areas that exhibit a high risk of wildfire and contain significant levels of biomass, thereby optimizing the placement of sensor modules.

However, even with this combined map approach, randomly or uniformly spreading sensor modules across identified high-risk areas remains inefficient and costly, as demonstrated in [57]. Such deployment strategies can result in suboptimal coverage, increased signal interference, and reduced overall system performance. To address this issue, in [31], a systematic placement strategy is proposed that combines both fire risk and forest density maps, ensuring that modules are strategically positioned to optimize data acquisition and sensing coverage while maintaining reliable and efficient operation.

② Sensor modules

The modules represent the senses of the proposed wildfire monitoring system, tasked with collecting environmental data to detect and monitor fire ignition in real time. Existing approaches in the literature have employed a range of microcontrollers and sensors to monitor forests applied in a WSN. However, these works also reveal critical limitations and challenges that the proposed system aims to address.

One major limitation in the related works is the significant energy consumption of sensors, particularly those designed for gas detection. While effective for identifying early signs of combustion, gas sensors are known for their high power consumption and often lack a standby mode. This absence of energy-saving features results in rapid battery discharge, particularly in remote deployments where periodic maintenance is complex. Another challenge is the high cost of gas sensors, which restricts their widespread adoption in large-scale implementations. Despite their potential for enhancing fire detection accuracy, the cost of these sensors limits their viability for resource-constrained regions. Moreover, WSNs based on gas sensors may wrong pinpoint the fire ignition when relying solely on gas sensors. Environmental factors such as wind can carry smoke away from the actual fire source, leading to false-positive detections and misaligned responses.

Solar panels have also been explored in existing systems to extend the operational life of sensor modules. However, deploying solar panels in forest environments presents unique challenges. When modules are installed on tree trunks, the efficiency of solar panels is significantly reduced due to shading from the forest canopy. Additionally, the natural growth of vegetation over the monitoring period can further obstruct sunlight, necessitating frequent maintenance to clear the panels.

In this regard, the proposed system adopts several innovative strategies to enhance the effectiveness and efficiency of the sensor modules. Duty-cycling and event-driven transmission are incorporated to optimize energy consumption while maintaining sensor accuracy, ensuring that modules operate efficiently without unnecessary energy consumption. Additionally, the system prioritizes the use of affordable yet reliable sensors, enabling broader adoption in resource-constrained regions without compromising detection performance.

The proposed system also emphasizes a multi-sensor approach, integrating diverse data sources. The system can more accurately identify ignition points and significantly reduce false alarms caused by environmental interference, such as wind-driven smoke displacement. Furthermore, to mitigate the limitations of solar panels in shaded forest environments, the system relies on high-capacity batteries designed to support extended operation periods without requiring frequent recharging or replacement. This eliminates the need for periodic maintenance associated with vegetation growth or insufficient solar energy, ensuring consistent and reliable module performance.

These advancements of the proposed sensor modules are a disruptive solution compared to existing literature, addressing critical limitations and setting a new standard for wildfire monitoring systems. In this sense, Chapter 3 is dedicated to an in-depth exploration of the sensor modules. This chapter will cover the selection of sensors, design and development of the modules, battery consumption optimization, and overall functionality within the system.

③ LoRaWAN data transmission, ④ LoRaWAN gateway, and ⑤ Mobile internet

Communication is fundamental for **WSNs**, particularly in the context of wildfire monitoring. Related works emphasize the use of radio frequency technologies due to their ability to transmit data over long distances while maintaining low power consumption. These technologies prioritize efficient data handling, as **WSNs** typically operate with constrained power and limited bandwidth. Among the most effective solutions are **LPWAN** technologies, which are specifically designed to address the unique challenges of **WSNs**, such as power efficiency, **LoRa** communication, and reduced data rates. Technologies such as **LoRaWAN** have been extensively explored in the literature and applied in diverse monitoring systems.

LoRaWAN stands out as a particularly suitable choice for wildfire monitoring due to its combination of low-power and **LoRa** communication capabilities. With its ability to cover distances up to several kilometers in rural or forested environments, **LoRaWAN** enables reliable communication in areas where traditional connectivity solutions, such as Wi-Fi or cellular networks, are impractical or economically unfeasible. Moreover, its topology facilitates data aggregation, reducing the risk of packet loss and ensuring reliable transmission.

The proposed system employs a hierarchical communication structure, where **LoRaWAN** gateways serve as central point. These gateways are designed with a focus on low-power, long-range communication to facilitate data transmission from remote sensor nodes, addressing the challenges posed by dense vegetation and diverse topography. Moreover, **LoRaWAN** gateways are equipped with capabilities for easy integration into the Cloud. They can connect to a cloud server via **Long-Term Evolution (LTE)** (4G/5G) and/or Ethernet, this infrastructure effectively addresses common issues such as signal interference, data loss, and limited scalability, frequently observed in traditional **WSN** deployments.

Another critical aspect is energy efficiency. Both **LoRaWAN** modules and gateways are designed to operate under low-power conditions, extending the lifetime of the entire network. For sensor modules, low-power sleep modes are integrated to reduce energy usage during idle periods, while gateways are optimized for minimal energy consumption without reducing performance. This design aligns with the principles of sustainability and long-term operation, which are crucial for deployment in resource-constrained areas.

The proposed communication architecture, anchored by **LoRaWAN** and mobile internet, delivers a scalable, reliable, and energy-efficient solution for wildfire monitoring. It ensures seamless data transmission from sensor nodes to the Cloud, overcoming common challenges such as connectivity in remote areas and data reliability.

⑥ Cloud server

The Cloud server is a component responsible for managing, processing, and storage the data collected by the **WSN**. However, an analysis of related works reveals a significant gap in the literature regarding software and frameworks explicitly developed for this type of monitoring system. While hardware components such as sensors and communication protocols receive considerable attention, the data flow within the system is often underexplored.

Existing papers rarely detail how the data is received, stored, and processed after transmission from the **WSN**. For example, the methods and tools for managing data streams, handling large-scale data storage and ensuring system scalability are rarely discussed. Moreover, the lack of emphasis on open-source solutions represents an opportunity to promote transparent and cost-effective system designs that can be adopted and adapted by a broader audience.

Another overlooked aspect in the literature is the potential for data analysis to enhance the value of the collected information. While many systems are designed for real-time monitoring, few address how advanced data processing, such as aggregation, anomaly detection, and trend analysis, can improve decision-making. Furthermore, related works rarely explore how algorithms can be implemented and executed in the cloud to enable real-time data processing and predictive analytics. This omission limits the ability of existing systems to evolve into proactive decision-support tools, which are critical for wildfire management.

To address these limitations, Chapter 4 proposes a system that emphasizes the development of an open-source cloud-based architecture. This architecture aims to seamlessly integrate all stages of the data flow, from reception to analysis and visualization. The system is focused on open-source

technologies to ensure accessibility and adaptability. The development process details how raw data from the **WSN** is received, pre-processed, and stored in the cloud, along with the methods and technologies employed at each step.

Another key innovation is incorporating advanced data analysis techniques within the cloud server. Algorithms for detecting anomalies, identifying trends, and generating alerts are implemented to enable real-time processing of the incoming data. Additionally, **ML** models can be trained and deployed within the cloud environment, adding predictive capabilities to the monitoring framework. These capabilities enhance the system's ability to provide actionable information, such as predicting data patterns or identifying high-risk zones based on historical patterns. As this theme represents a significant contribution to the proposed system and the broader field of monitoring technologies, Chapter 5 is dedicated to developing the data analysis and **ML** models.

⑦ Control center and ⑧ Fire ignition early attack

The control center receives, visualizes, and interprets data collected and processed by the **WSN** and the cloud server. In this component, the authorities are expected to receive an alert or real-time data. As reviewed in the related works, often lack a detailed discussion of control center functionalities and their integration into the overall monitoring system. Typically, emphasis is placed on data collection and transmission, with limited exploration of how data visualization and decision-support tools can enhance firefighting teams' response capabilities. Moreover, integrating advanced visualization tools, such as real-time dashboards and alert mechanisms, remains underexplored in the literature.

The proposed system addresses these gaps by developing an intuitive and functional control center. This includes a real-time dashboard that provides a comprehensive overview of the monitored region, incorporating key features such as:

- **Geospatial Visualization:** Integration of maps to display sensor locations, detect anomalies, and predict data patterns, enhancing situational awareness.
- **Alert Management:** A tiered alert system to classify the severity of detected events, ensuring that critical situations are prioritized for immediate response.
- **Historical Data Analysis:** Tools for visualizing historical data trends, allowing decision-makers to understand patterns and plan for long-term risk mitigation.

Beyond visualization, the control center is a command for coordinating firefighting efforts. This integration ensures that field teams are equipped with accurate, timely information, enabling them to make informed decisions and optimize resource allocation. The proposed system facilitates an early attack on fire ignition by delivering precise location data and contextual information. This early intervention can prevent minor ignition points from escalating into large-scale wildfires, minimizing environmental and economic damage.

2.4 Final Considerations

This chapter provided a comprehensive overview of **WSN**-based monitoring systems, emphasizing their applications in environmental contexts and, specifically, in early wildfire detection. Initially, the evolution and benefits of **WSNs** were contextualized, highlighting their adaptability, scalability, and efficiency in extreme and remote scenarios. Subsequently, a review of related works identified advancements, limitations, and opportunities for improvement, emphasizing how existing technologies face challenges such as energy consumption, high costs, and the lack of integration with open-source solutions.

The proposed architecture outlined in this chapter builds upon these findings, addressing identified gaps. The system integrates key components, such as low-cost, energy-efficient sensor modules, robust communication through **LoRaWAN**, and a cloud-based processing infrastructure for real-time analysis and visualization. These elements were detailed to highlight their individual functions and interconnections within the system, culminating in a scalable and accessible solution for wildfire-prone regions in Portugal.

Additionally, this chapter introduces innovations that distinguish the proposed system from related works. These include advanced energy-saving strategies and the implementation of predictive analyses in the cloud to support decision-making. The content presented establishes a robust foundation for the subsequent chapters. Chapter 3 will explore the development of sensor modules in detail, while Chapter 4 will address the data flow and software infrastructure. Finally, Chapter 5 will focus on developing and application of **ML** models for predictive analyses, completing the system's development cycle. Thus, this chapter achieves its goal of consolidating the state of the art and proposing innovative architecture as the cornerstone for the remainder of the work.

Chapter 3

The sensor nodes

This chapter is dedicated to designing and implementing advanced sensor modules intended to identify forest fires. The main objective is to provide an effective solution for the early detection of forest fires, aiming to preserve natural ecosystems and the population's safety. The presented research addresses the development of innovative monitoring systems, integrating state-of-the-art sensing technologies and advanced analytical methods. The structure of this chapter is divided into sections that address crucial aspects of developing and evaluating sensor modules. The chapter is structured as follows:

- Section 3.1 – provides an overview of the general context of fire sensors, highlighting their importance in the current scenario of wildfires and environmental disasters.
- Section 3.2 – conducts a comprehensive and detailed literature review on fire sensors, microcontrollers, and communications are presented in that provides a solid foundation for subsequent development.
- Section 3.3 – describes the module's design and assembly, including the sensor selection and the communication system.
- Section 3.4 – presents the outcomes achieved by the sensor modules concerning the aspects of communication and battery autonomy.
- Section 3.5 – discusses the findings from the sensor module tests, evaluating their performance and identifying areas for improvement.
- Section 3.6 – summarizes the findings, highlights the contributions of this stage of the work to the field of wildfire monitoring, and outlines future research directions.

3.1 General Contextualization of Fire Sensors

As the 21st century progresses, humanity faces unprecedented environmental challenges, including the growing threat of fires. The incidence of forest, urban, and industrial fires has increased, driven

by factors such as climate change, urban expansion, and agricultural practices. In this context, early and accurate fire detection using sensors becomes a fundamental pillar for risk management and emergency response strategies. Therefore, developing systems that can identify fires in their early stages is essential, and existing sensing technologies need to be thoroughly explored.

Some authors use image processing methods for fire detection through satellite images or digital camera technology [58], but they present some limitations. In [59], a system based on image processing to detect fire ignition is proposed. This system includes movements containing region detection based on background subtraction and color segmentation. Other interesting approaches involving image processing algorithms are [60, 61], whereas in [61], a supervised fine segmentation image classification algorithm for the improvement of image-based forest fire detection is presented. In [60], **Convolutional Neural Network (CNN)** based on the adaptive pooling method and conventional image processing techniques are combined to forest fire image recognition. According to [62, 63], the systems based on computer vision for fire detection, depends on three stages: flame pixel classification, segmentation of moving object, and analysis of a target region. In general, these are precise methods to detect fire in small areas, such as commercial areas, power plants, or forests that are not dense. But, in huge regions or dense forests, if a fire ignition occurs, the fire identification using image processing will alert too late, or it will detect with high imprecision.

This imprecision is primarily due to the extended distances over which the detection systems must operate, and the terrain's topography may further intensify the challenge. In such large areas, the time taken to transmit and process the data can be substantial, slowing timely responses. Moreover, fires emerging behind a mountain may remain undetected if cameras are not strategically positioned to capture such covered areas. Additionally, wind action may significantly compromise the effectiveness of smoke detection through image processing. Wind may disperse smoke, making it less dense and more complex to detect, or it could mislead the identification of the fire's origin point due to the optical illusion created by smoke being blown across the landscape. Furthermore, if all of these factors come together, the impacts could be immeasurable.

Moreover, the image-based system implies a higher cost with sophisticated equipment such as **GPS**, satellites, **Unmanned Aerial Vehicle (UAV)**, and increased image processing infrastructure. The high cost of image-based technologies could result in unscalability across large geographic areas [64]. It is also worth mentioning that the accuracy of this system is affected by weather conditions such as cloudiness and light incidence. For satellite-based detection, cloud cover can significantly obstruct the view, preventing the satellite's sensors from accurately detecting signs of fire beneath the clouds. This limitation can lead to detection delays or even failure to recognize an active fire in its early stages. Similarly, for **UAV**, besides cloudiness, factors such as fog, heavy rain, or even smoke from the fire can disturb the visibility and interfere with the effectiveness of image processing. **UAV** also face challenges in low light conditions, such as during twilight or at night, which can reduce the images' clarity and affect the accuracy of fire detection.

Considering the challenges mentioned above associated with image processing, satellite, and **UAV** technologies, some researchers have focused on studying patterns in the region with a high incidence of wildfires. According to [65], the highest incidence of fire events occurs in summer

during the morning and afternoon, and in [66], it was observed high temperatures and soil dryness contribute to the propagation of fire in areas with vast vegetation in the city of Quito. The work [65] reported the values of temperature and humidity always behave in the same way during a forest fire.

Meteorological factors have a substantial influence on forest fire ignition behavior. Moreover, they are beneficial to define a fire danger rating used by firefighters to determine the risk level of fire occurrence and thus provide an appropriate scale for managing such a crisis [67]. According to [67], temperature, soil temperature, and dew point have a strong correlation with fire occurrence, while humidity, precipitation, and wind speed have less influence on fire ignition. Another unique approach involving meteorological factors is presented in [68], in which drought factor, temperature, relative humidity, and wind are used to create a system to predict forest fire spread during a wildfire. Some types of sensors can detect physical parameters such as humidity, temperature, and combustion chemical parameters [69].

Another fire detection approaches have been addressed for a long time. In forests, the use of WSN has been studied by several authors, for example [70, 71, 72]. Some authors use the temperature, relative humidity and barometric pressure values to detect flames before the area is scorched, since temperature increase, and barometric pressure and humidity decrease as the fire evolves [47]. WSN is a current technology used in several fields [73]. Their capabilities for monitoring large areas, accessing remote places, and reacting in real-time allow their use in forest fire system detection [74].

WSN and information fusion methods are applied in [65] to detect forest fires. A base model was developed using experimental data, considering some time of the year and a specific period of the day. Then, the approach defined the temperature and humidity values that characterize a fire condition. The data collected by the WSN are compared with base model values to determine the presence of a fire. Moreover, sensor fusion techniques to enhance the performance of forest fire detection and location estimation are presented in [75]. Fusion on light and temperature measurements can also be done to perform the fire detection [76]. Since direct sunlight on the light sensor can be mistaken for fire, [76] proposes and compares two methods to outwit this problem, the methods are Threshold and the Dempster-Shafer Methods.

Despite continuous efforts over the years by the academic community to solve the detection of forest fires in their initial stage, in-depth research on fire sensors applied on a large scale is still necessary. Detecting large-scale fires, especially in areas that are difficult to access and with limited resources, requires an innovative approach as it is a complex subject and oriented towards low-cost solutions. In the following, a description of the research objectives for developing effective and affordable large-scale fire detection solutions using fire sensors is presented. These sub-objectives will be used to guide the research process.

Identification of Effective Methods

A crucial step in developing fire detection systems is identifying those methods that have proven effective and those that have not met expectations. Constant literature review and critical analysis of available technologies are fundamental to direct the development of more efficient solutions.

Therefore, the first objective is to identify fire detection methods that have proven effective in different environmental and geographic conditions and that have overcome the limitations of image-based methods (cost and detection time). The state of the art should evaluate the accuracy and reliability of different sensing approaches, such as temperature, humidity, smoke, **CO**, and **IR** sensors, in order to determine which methods are best suited for different scenarios.

Find a Low-Cost Solution for Large-Scale Deployment

Detecting large-scale fires through sensors is a challenge that requires low-cost solutions which can be widely implemented. To achieve this, affordable sensors that can cover large areas need to be created, along with cost-effective materials and components and efficient deployment strategies. By using intelligent techniques, alternative energy sources such as solar panels can be overcome, resulting in long-term energy savings. The second objective is to find low-cost solutions that can be distributed on a large scale and remain autonomous for months. The solution, once developed, would be able to detect forest fires using affordable sensors that can be deployed in large areas such as forests and remote regions.

Use the New Trends in Technologies (Hardware) to Detect Fire

The field of fire detection is not immune to technological innovations. It is critical to incorporate the latest trends in hardware and software to enhance detection capabilities. For example, using advanced microcontrollers and low-power communication systems can increase the lifetime of the modules. Furthermore, integrating **IoT** sensors could enable real-time data collection, improving responsiveness and detection accuracy. In this sense, the third research objective is to use the latest trends in hardware technologies to improve fire detection. It includes the exploration of advanced microcontrollers, high-precision and low-cost sensors, and low-power communication systems.

Develop a Wireless Sensor Network able to detect fires

Creating a **WSN** dedicated to fire detection is a significant step in fighting wildfires. A **WSN** allows for the strategic distribution of sensors across a wide geographic area, forming a robust network to monitor environmental conditions and detect fires. The use of efficient and low-power communication protocols is essential to ensure that the **WSN** operates reliably. The fourth objective is to develop a **WSN** dedicated to fire detection. Therefore, research should explore efficient and low-power communication protocols as well as sensor information fusion to improve detection accuracy.

3.2 State of the Art: Fire Sensors, Communication and Microcontrollers

Detecting forest fires early is critical to successful management, as rapid response times are the key to controlling the spread of the fire [11]. As such, it is of the utmost importance to detect the

first signs of ignition and alert the appropriate authorities immediately [77]. The consequences of failing to alert a forest fire early may be catastrophic [78]. In this sense, the early detection of forest fires is crucial to minimizing their impact on the environment and human life. To mitigate the risks associated with delayed forest fire detections, it is essential to implement effective detection systems that can identify fires in their early stages. With the implementation of advanced detection systems and rapid response times through fire sensors, it is possible to control forest fires and protect the natural resources. For this reason, this section is dedicated to in-depth research on the approaches developed and their usage of the different types of sensors.

To theoretically support and ensure an updated and comprehensive approach to sensor technologies for fire detection, extensive research was carried out in academic databases. This bibliographic search mainly aimed to identify recent studies and advances in the area, with particular emphasis on published research. This search was conducted employing a keyword in the search field, aiming to capture a wide range of relevant publications. The term selected for the search was *fire sensor*, an umbrella term expected to encompass a wide range of technologies and approaches in fire detection research.

The search process was carried out in several recognized academic databases: Scopus, Web of Science, IEEE Xplore, Science Direct, and Google Scholar. These platforms were chosen for their extensive scientific literature collection and relevance in engineering, technology, and environmental sciences. The search on each platform was refined to include papers that use/mention the term *fire sensor* at any moment (Title, Keyword, Abstract, and so on). Therefore, it excluded publications that did not strictly fit within this scope.

Through this process, it was possible to compile a significant collection of papers and publications related to methods of identifying fires using sensors. However, the results of each search on these platforms returned a large volume of papers, as shown in Table 3.1.

Table 3.1: Count of papers addressing fire sensors in academic databases.

Keyword	Scopus	Web of Science	IEEE Explore	Science Direct	Google Scholar
Fire Sensor	51345	9540	60194	69185	181000

After this initial iteration of the bibliographical research, using the term *fire sensor*, it became evident that there is an extensive volume of publications in this domain. Although this broad spectrum of literature offers a comprehensive overview of the field, the substantial amount of papers available has posed significant challenges regarding efficient review and analysis. To effectively manage this wealth of information and focus on the most relevant aspects of this study, it became necessary to refine the search strategy, incorporating a set of more specific keywords.

Search refinement was performed by selecting specific fire detection methods using sensors, covering all possible approaches in this field. In this context, it was decided to incorporate another step in the research process: the analysis of existing systematic reviews in the area, using a combination of the keywords *fire sensor* and *systematic review*.

The aim of this approach was twofold. Firstly, systematic reviews, known for their rigorous and comprehensive methodology, offer a synthesis of existing research, presenting an overview of current trends, research gaps, and, crucially, search strategies employed by other researchers. Second, by examining systematic reviews, it is possible to identify which keywords have been frequently used by other papers in sensor fire detection. With this information, it is expected to define a search strategy with practices established in the scientific community. Therefore, the results obtained are presented in Table 3.2.

Table 3.2: Number of papers found with some relation to fire sensors and systematic review.

Keyword	Scopus	Web of Science	IEEE Explore	Science Direct	Google Scholar
Fire Sensor	51345	9540	60194	69185	181000
Fire Sensor + Systematic Review	4195	9	65	8910	23300

In an effort to ensure that the scope of this search encompassed the latest innovations and discoveries in the field of sensor fire detection, a further refinement of the search strategy was implemented, limiting the publication period of papers to between the years 2016 and 2020 (the last five years from the beginning of this search). This decision was made to focus the analysis on the most current and relevant contributions in the field, reflecting the most recent technological and theoretical advances. The number of papers can be seen in Table 3.3.

Table 3.3: Number of papers found with some relation to fire sensors and systematic review from 2016 to 2020.

Keyword	Scopus	Web of Science	IEEE Explore	Science Direct	Google Scholar
Fire Sensor	51345	9540	60194	69185	181000
Fire Sensor + Systematic Review	4195	9	65	8910	23300
Fire Sensor + Systematic Review (2016 – 2020)	2635	8	48	3319	902

Given the high volume of papers initially identified on several academic research platforms, it became necessary to implement a more rigorous filtering strategy to refine and focus the search. This methodological improvement aimed to ensure the inclusion of only studies directly aligned with fire detection using sensors and relevant systematic reviews in the area. An additional filter was established to achieve this objective, requiring that the expressions `fire sensor` and `systematic review` be present only in the papers' keywords, titles, and abstracts.

In the Scopus database, the initial search resulted in 51345 papers. After refinement using more specific fields, this number was reduced to 92 works, representing an approximate 92.82% decrease. Similarly, in Web of Science, the number of papers found initially was 9540, which decreased to just 8 after refinement, an impressive reduction of about 99.92%. In IEEE Xplore, the initial search identified 60194 papers, which were reduced to 48, indicating a decrease of approximately 99.92%. At Science Direct, the initial search found 69185 papers, and this number was reduced to 48 after refinement, a drop of roughly 99.93%. Finally, on Google Scholar, the initial search resulted in 181000 papers, which were refined to 478, equivalent to a reduction of approximately 99.74%.

Despite these significant reductions, which oscillated between 99.74% and 99.93%, there are still 674 works to be analysed. An additional filter was adopted based on the number of paper

citations to make this analysis more manageable and focused. The five most cited papers from each year (2016 – 2020) of each database were carefully read and examined. This criterion was chosen as an indicator of the relevance and impact of papers in the field of sensors used for fire detection.

In this sense, focusing on each academic platform's five most cited papers of each year, it was expected that this methodology would result in a total of 103 works covering a diverse spectrum of studies and approaches. However, an occurrence was observed: some papers appeared as the most cited on more than one platform. This overlap reflects the relevance and impact of these studies in the field, but it also means that the total number of unique works is lower than initially anticipated. As a result, the adequate number of distinct papers collected for analysis was reduced to 95.

After compiling this list of 95 papers, the abstracts of each of them were carefully read. This step was essential to ensure that each paper was rigorously evaluated for its direct relevance to the central topic of the research - the identification of fires through electronic devices or other sensing methods. During this evaluation, it was found that 55 of the 95 papers, despite being highly cited and relevant in their respective areas, did not mention or explore methods related to fire detection. These papers were, therefore, discarded from subsequent analysis as they did not strictly align with the scope of this study.

After the careful selection and filtering process, a set of 40 papers was identified as central to the study of fire detection methods using sensors. The next step is reading each paper, focusing on identifying and noting the terms and methodologies mentioned by the authors for fire identification. This analysis allowed the collection of terms and expressions used in the field, reflecting a rich diversity of approaches and technologies. To organize and synthesize this information, a comprehensive table was created. Table 3.4 catalogs each of the 40 papers, associating them with the specific terms they used to describe fire identification. This data structure offered a clear and orderly view of the terminologies and methodologies used in the studies.

Table 3.4: List of studies in fire detection from 2016 to 2020 in the top five of most cited in academic source platforms per year.

Authors	Mentioned Fire Detection
Lawley et al. (2016) [79]	Biosensors: biomass humidity, leaf chlorophyll, tree canopy temperature; Thermal and multispectral imaging (Satellites)
Udayraj et al. (2016) [80], Sankar et al. (2016) [81]	Surface heat sensors; Heat flux sensors; Surface temperature sensors; Temperature sensors
Tawalbeh and Bakhader (2016) [82]	Smoke sensors; Temperature sensors
Mobin et al. (2016) [83]	Smoke sensors; Temperature sensors; Flame sensors (IR)
Sulla-Menashe et al. (2016) [84], Atwood et al. (2016) [85], Adolf et al. (2017) [86], Forkel et al. (2017) [87], Fornacca et al. (2017) [88], Pereira et al. (2017) [89], Zhu et al. (2017) [90], Whitman et al. (2018) [91], Chuvieco et al. (2019) [92], Roy et al. (2019) [93], Wang et al. (2019) [94], Beamish et al. (2020) [95], Chuvieco et al. (2020) [96]	Optical sensors; Multispectral imaging (Satellite)
Dugué (2017) [97]	Flame Scanner (IR); Gas Sensors; Oxygen sensors; Laser diode (carbon dioxide)
Singh and Sharma (2017) [98]	Optical sensors; Temperature sensors; Humidity sensors; Smoke sensors; IR sensors; Atmospheric pressure sensors; Gas sensors
Veraverbeke et al. (2018) [99], Naderpour et al. (2019) [100]	Optical sensors; Multispectral imaging (Satellite); IR sensors
Jiang et al. (2018) [101]	Temperature sensors
Muduli et al. (2018) [43]	Smoke sensor; Temperature sensors; Humidity sensor; Gas sensor; Atmospheric pressure sensor
Park et al. (2018) [102]	Temperature sensors; Humidity sensor; Image processing (cameras); Gas sensor; Flame sensor (IR)
Verma et al. (2018) [103]	Temperature sensors; Humidity sensor; Image processing (cameras); Multispectral imaging (Satellite)
Bu et al. (2019) [104]	Image processing (cameras); Gas sensors; Temperature sensors; Radiation sensors (IR); Biological sensor
Wang et al. (2019) [105]	Heat sensors; Smoke sensors; IR sensors
McNay et al. (2019) [106]	Smoke sensors; Temperature sensors; Atmospheric pressure sensors; IR sensors
Ejaz et al. (2019) [107]	Smoke sensors; IR sensors; Image processing (cameras)
Gaur et al. (2019) [108]	Heat sensors; Temperature sensors; Smoke sensors; IR sensor; Gas sensor; Nanotubes; Microwave radiometer sensor; Acoustic sensors; Biosensors (bugs)
Bao et al. (2019) [109]	Radiation sensor (IR); Smoke sensors; Gas sensors
Stauffer (2020) [110]	Ignitable liquid residues; Gas sensors
Ma and Wu (2020) [111]	Smoke sensors; Temperature sensors; Gas sensors
Xu et al. (2020) [112]	Soil moisture sensors; Temperature sensors
Stavroulas et al. (2020) [113]	Temperature sensors; Humidity sensors; Atmospheric pressure sensors; Particle sensors; Particulate matter
Eugenio et al. (2020) [114]	Image processing (cameras); Laser sensors; Multispectral (UAVs); IR sensors; Temperature sensors
Roque and Padilla (2020) [115]	Gas sensors; Temperature sensors; Smoke sensors; Image processing (camera)
Mandal et al. (2020) [116]	Heat flux sensors; Surface temperature sensors
Harikumar et al. (2020) [117]	Image processing (cameras); Temperature sensors

Through an in-depth analysis of the papers selected for this study on fire detection methods, a thoughtful decision was made to exclude specific terms that align more closely with the field of computer vision. This exclusion was based on previously established justifications, which focus on direct relevance and practical applicability in the context of fire detection by sensors. Specific terms eliminated from the final analysis are:

- Thermal and multispectral imaging (Satellites)
- Multispectral imaging (Satellite)
- Image processing (cameras)
- Multispectral imaging (UAVs)

Although these technologies and methodologies are undoubtedly advanced and have their application in fire detection, they fall more clearly into the domain of computer vision and image processing, which, although related, diverge from the central focus of this study. Then, Figure 3.1 defines the list of terms related to fire detection using sensors.

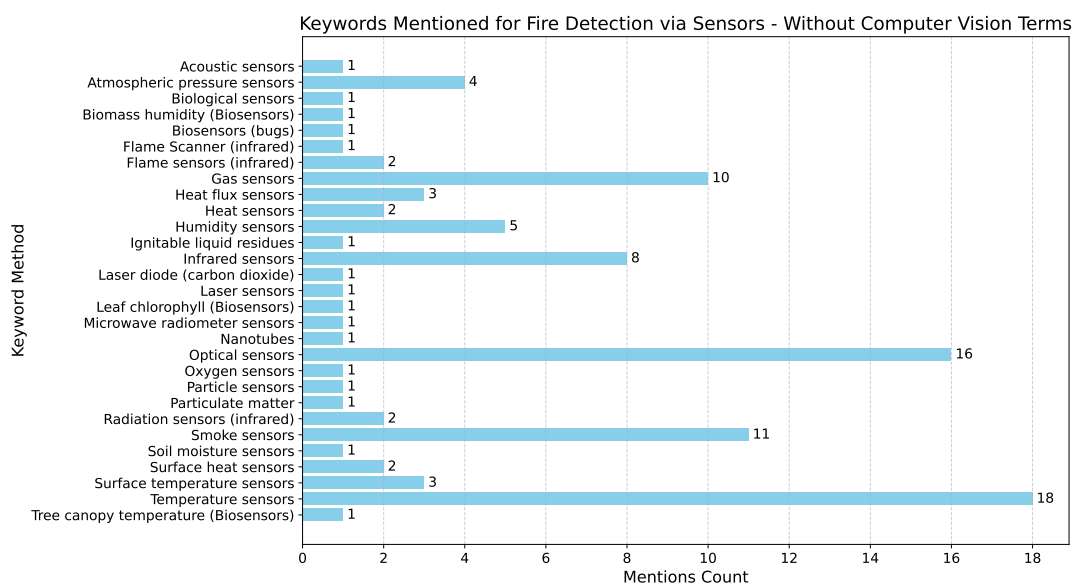


Figure 3.1: List of all terms found for fire detection using sensors in the most cited literature from 2016 to 2020 after removing computer vision terms.

Performing a literature search on sensor fire detection methods with all of the remaining 29 terms can present a significant challenge due to the wide variety of terms and technologies identified. It is expected that few publications will mention all 29 of these terms simultaneously. On the other hand, exploring each term individually could quickly become a chaotic and impractical effort.

To overcome this obstacle, a strategy of grouping terms with common characteristics and functionalities was adopted. For example, terms that contain the words *Heat* and/or *Temperature* could be applied in separate searches from those that do not include these two words. This

method not only simplifies the research process but also facilitates a more precise and organized understanding of different technologies and their applications in fire detection.

Therefore, understanding the functionalities of each of the 29 methods, the terms were organized into seven main groups, each representing a distinct category of sensing technologies. Figure 3.2 presents a comprehensive list of terms that have been separated based on their corresponding functionalities. The list provides a clear and organized view of the different terms and their respective functions.

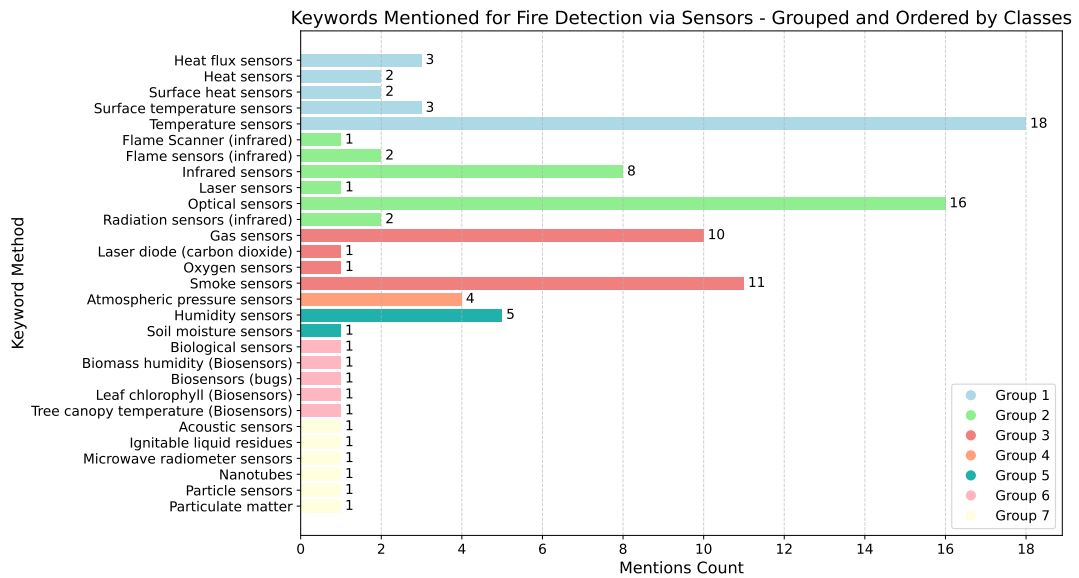


Figure 3.2: List of all terms found for fire detection using sensors in the most cited literature after classification.

- **Group 1 - Temperature Sensors:** This category includes terms related to devices that detect variations in heat and are crucial in the early identification of fires. Temperature sensors are critical because of their ability to warn of abnormal increases in heat, a primary indicator of fires.
- **Group 2 - Optical Sensors:** Groups terms associated with sensors that use optical properties, such as light (through sensors that measure radiation) and visibility (through laser sensors), to detect fires. These sensors are particularly useful in scenarios where flames or smoke can be visually seen.
- **Group 3 - Gas Sensors:** Includes sensors to identify specific combustion gases. These devices are essential for detecting fires in their early stages, especially indoors or where the presence of smoke is limited.
- **Group 4 - Atmospheric Pressure Sensors:** This category encompasses sensors that detect changes in atmospheric pressure. This phenomenon may indicate the occurrence of a fire.

- **Group 5 - Humidity Sensors:** Includes technologies that measure air or environmental humidity, as humidity variations can indicate fires, especially in forest or natural contexts.
- **Group 6 - Biosensors:** Groups of sensors that use biological components to detect the presence of fires or even sensors that are applied to vegetation to collect essential parameters for identifying fires. These are innovative in fire detection and can offer unique approaches to early identification.
- **Group 7 - Miscellaneous Sensors:** A category encompassing a range of sensors with diverse and less common functions, which are still relevant in fire detection.

With the definition of these seven groups, where it was possible to establish an organized structure to address the wide range of fire detection technologies through terms, the next phase of this study is carry out an in-depth study of each group. Therefore, in subsequent subsections, each of the seven identified group - Temperature Sensors, Optical Sensors, Gas Sensors, Atmospheric Pressure Sensors, Humidity Sensors, Biosensors, and Miscellaneous Sensors - will be explored in detail. This in-depth analysis aims to deepen the understanding of each sensor category and examine how these technologies have been applied in practice for fire detection. It will allow for a more complete knowledge of the capabilities and limitations of each technology, providing valuable insights into best practices and innovations in the field.

3.2.1 Fire sensors: Types and operating principles

This section is dedicated to the literature review focused on the groups (defined previously) of fire detection technologies. Here, the main objective is to delve into the academic literature to identify studies and works that used terms related to each of the seven groups and analyse the types of sensors applied, along with their advantages and limitations.

The scope of this analysis is twofold: first, it seeks a detailed understanding of the different types of sensors used in fire detection, highlighting how each technology works and what operating principles are behind their effectiveness. This understanding is crucial for evaluating the efficacy of sensors in detecting fires in different contexts and the nuances that influence their choice in practical applications. Second, the analysis focuses on highlight the pros and cons of each sensor type, as reported in academic studies. This part of the analysis is vital to identify existing limitations in current technologies and to understand areas that require development or innovation.

3.2.1.1 Temperature and Heat Sensors

Through 20 papers, techniques involving fire sensors, sensor models, and their energy consumption were identified. The selection of these 20 papers was made from a total of 29, as analysed in the previous section. These 20 works were chosen based on the appearance of terms present in **Group 1**.

According to [105], heat sensors are devices that can measure heat in environments as part of fire detection systems, such as buildings or batteries. In [108], the authors categorized these

sensors into three main types: fixed temperature, rate of rise, and rate of compensation. The authors also indicate several heat sensor models, with specific mechanisms for detecting thermal changes. Among them, the most common are Thermocouples, Bimetallic heat sensors, and Thermostors.

In this sense, in [80], two approaches for measuring heat flux during fire tests on fabrics are presented through the application of chromel-constantan (80 sensors are used) and chromel-alumel (with 44 sensors) thermocouples on their textile surfaces. In [116], five models of heat flux sensors are applied in the test of finding fabric insulation. The sensors used were the aluminum sensor, skin simulant sensor, epoxy sensor, copper sensor, and PyroCal copper sensor (the last three are manufactured by the company Precision Products). To measure the surface heat flux, in [81], the possibility of using Synergistics sensors to measure the thermal effect in furnaces is highlighted, and the possibility of using Chordal Thermocouples in the thermal evaluation of cylindrical walls is also highlighted. In all these last three works there is no specific information about the energy consumption of the sensors.

Another term that is common for detecting fires by electronic devices is the term: temperature sensors. In [101], temperature sensors are used as part of experiments to measure air temperature in different scenarios, such as fires. In [43], the authors propose the use of temperature sensors in WSNs in mines to increase the early identification of fires. However, none of the last-mentioned works highlight sensor models to be applied. Real-time monitoring through a WSN using temperature sensors is described as essential in [103] and [111]. Both indicate that the energy consumption of the modules is significant to improve the WSN's performance. However, the work in [103] elaborates on a WSN's architecture and does not discuss which sensors could be installed in the nodes. In [111], the authors point out that excellent cost-benefit can be obtained through multimodal sensors because these types of devices can provide several critical parameters in fire identification.

Temperature sensors are incorporated into a system to monitor air quality, as presented in [113]. In this work, the BME280 sensor from Bosch Sensortec is used to measure temperature, relative humidity, and barometric pressure. The sensor was chosen to combine low cost with quality of acquisition. In the same sense as multimodal sensors, in [83], it is proposed to use the SEN11301P sensor in nodes whose objective is to identify fires. This sensor is composed of the DHT11 component, which is capable of providing quality measurements of humidity and temperature while keeping a low cost. Although the authors can identify fire from a lighter, this work does not carry out battery studies and forest applications. In [115], a temperature sensor, model LM35, is used for fire monitoring. This sensor was installed in a prototype that sends alerts when fires are identified. However, the authors do not discuss the module's energy consumption and communication distance efficiency.

3.2.1.2 Optical Sensors

The selection of 22 articles was made based on terms relevant to optical sensors, classified as **Group 2**. From the analysis of these articles, techniques involving optical sensors, sensor models, and their energy consumption were identified.

Optical sensors used in satellites operate in several spectral bands, including near-infrared, mid-infrared, and thermal **IR**. Notable examples include Landsat Thematic Mapper and Enhanced Thematic Mapper Plus [84, 91, 94], Moderate Resolution Imaging Spectroradiometer [86, 88, 90], and Visible Infrared Imaging Radiometer Suite [92, 89, 96]. Sensors onboard satellites provide wide and frequent coverage, enabling continuous monitoring of large areas [85].

These technologies can be adapted to sensors installed directly on trees [98, 87]. Low-cost **IR** sensors can be implemented in modules that continuously monitor the presence of fire and ambient temperature [98]. [87] reports that these sensors can operate similarly to satellite systems but with the advantage of a local and immediate response. Furthermore, [83] addresses that flame sensors operating in the **IR** spectrum are commonly used to detect the presence of fire through the identification of **IR** radiation from flames. These sensors are highly sensitive to the flame spectrum and can detect light within the wavelength of 760 nm-1100 nm.

Flame Scanners are devices used mainly in industrial environments to monitor the presence and quality of flame in combustion systems. These devices are applied to industrial boilers and furnaces [97]. Laser sensors can detect the presence of specific particles in the atmosphere, such as smoke and ash, offering early detection of fires [114]. Radiation sensors are devices that measure thermal radiation emitted by hot objects. These sensors can be used to detect the presence of fire [107, 108]. In [114], **IR** sensors are applied to Remotely Piloted Aircraft Systems platforms to monitor fires and water stress in crops.

None of the 22 papers discuss about the energy consumption of their mentioned sensors.

3.2.1.3 Gas sensors

Forest fire detection can be carried out using gas sensors that monitor the presence of specific gases and particles associated with combustion. Therefore, **Group 3** has 15 selected articles that can guide the development of the module proposed in this chapter through the contributions of operation, model, and energy consumption.

Gas sensors are devices designed to detect and measure the concentration of specific gases in a given environment, such as monitoring in buildings [111] and underground mines [43]. They are often used in fire detection systems to identify gases released during combustion, such as **CO₂** and **CO**. Different types of gas sensors are mentioned in [108], such as metal oxide semiconductor sensors and carbon nanotube-based sensors, both of which have high sensitivity and the ability to detect multiple toxic and explosive gases. Laser diodes can measure **CO₂** because they can perform line-of-sight measurements with fast response time [97]. In [115], the MQ-9 gas sensor measures **CO**, coal gas, and liquefied petroleum gas. In [83], the MQ-2 gas sensor can detect gases such as **CO**, hydrogen, methane, smoke, propane, and alcohol. This sensor is known for its accuracy, fast response time, sensitivity, high reliability, longevity, and cost-effectiveness. Sensor associations can be used to expand the range of gases detected, as is the case with the Multi Gas Detectors presented in [102], with a maximum consumption of 12 W.

Smoke sensors are mentioned in several works as part of fire detection systems. By identifying the presence of particles in the air due to smoke, these sensors can indicate combustion in its

different stages [98, 104, 107]. Using oxygen sensors, it is also possible to find applications that identify fires through the absence or decline of oxygen in the environment. These sensors are based on zirconium oxide, and their applications in heaters and ovens, highlighting the challenges and best practices for reliable measurements, can be found at [97].

3.2.1.4 Atmospheric Pressure Sensors

Group 4 includes four selected articles that discuss the ability of atmospheric pressure sensors to identify fires, providing information on operation, models, and energy consumption.

In [43], which chose an underground mine as a study area, atmospheric pressure sensors are applied to measure air pressure, aiming to monitor and detect fires. Whereas in [113], BME280 barometric pressure sensors were used for environmental monitoring. This sensor is used in the PA-II device for air quality monitoring and measuring temperature, humidity, and barometric pressure. However, the specific energy consumption of this sensor is not discussed.

3.2.1.5 Humidity Sensors

Monitoring air and soil humidity variables can help identify forest fires. Therefore, **Group 5** covers six articles that explore how humidity sensors can be used for this purpose, providing information on operation, models, and energy consumption.

In [98, 43], moisture sensors are discussed in the context of collecting environmental data to detect forest and underground mine fires, respectively. In [103], it is highlighted that humidity sensors need prudence about energy consumption when inserted into WSN for environmental monitoring.

Soil moisture sensors can help maintain proper moisture levels and prevent fires caused by extreme drought. In this sense, [112] addresses soil moisture sensors in the context of intelligent farm irrigation using a soil moisture sensor provided by Youchuang Co., Ltd. The sensor is powered by a 1.5 mF capacitor that can be charged to 3.74 V in 462 seconds from a self-sufficient power source, suggesting efficient energy consumption for periodic monitoring.

3.2.1.6 Biosensors

Using biosensors to detect forest fires involves monitoring biological indicators that can signal the presence of fire. **Group 6** has three articles exploring this approach, providing information on biosensors' operation.

Biosensors are devices that use living organisms or biological components to detect environmental changes; one of these possible changes could be fire. Although the articles reviewed do not provide specific definitions, they discuss several biosensor applications. In [79], tree canopy temperature and chlorophyll fluorescence are mentioned as indicators of vegetation stress. In [104], the possibility of mobile biological sensors in the context of a forest fire detection system is described. These sensors transmit temperature or radiation level changes and send the animals' geographic location. In [108], there is an example of biomimetic sensors based on photomechanical

IR receptors found in “fire-loving” insects. These sensors mimic insects’ ability to detect fires, but do not provide specific details about models or energy consumption.

3.2.1.7 Miscellaneous Sensors

Some types of sensors can identify forest fires through alternative methods, which can be directly or indirectly related to a fire’s characteristics. Therefore, **Group 7** includes three articles that discuss six different types of sensors: acoustic sensors, flammable waste liquids, microwave radiometer sensors, nanotubes, particle sensors, and particulate matter sensors.

Acoustic sensors detect changes in the speed of acoustic waves due to changes in the sensor material parameters. These sensors are used to detect fires in open areas, such as forests, and can operate in adverse conditions, including the presence of smoke, dust, and vapors [108].

Detecting flammable liquid residues can be an indirect method of identifying fires. Several techniques, including portable GC-MS and active air sampling methods, are used to analyse the residues of these chemical materials and determine when they are being used deliberately to start fires [110].

Microwave radiometer sensors are effective in detecting fires under severe conditions, such as the presence of smoke and dust. They can detect fires through obstacles such as walls, making them useful for detection in industrial and forestry areas. And sensors based on carbon nanotubes can identify gases, they are highlighted by their high sensitivity and low cost [108].

Particle and particulate matter sensors measure concentrations of PM2.5 and PM10 in urban and suburban environments. The accuracy of these sensors can be affected by factors such as relative humidity and high concentrations of coarse particles [113].

3.2.1.8 Advantages and disadvantages of each type of sensor

Temperature and humidity sensors are widely used due to their high accuracy in detecting temperature changes. The DHT11 sensor is particularly advantageous because it is multimodal, providing temperature and humidity measurements with good accuracy and low cost. It also supports power-saving modes and can be turned on and off whenever needed. Although the BME280 sensor offers high accuracy and the ability to measure temperature, humidity, and atmospheric pressure, it usually has a higher cost, which can be a limiting factor when scalability is required. The LM35 could be an option for temperature measurements but provides just one parameter.

Optical sensors, especially those operating in the 760nm-1100nm IR spectrum, are highly sensitive to flame radiation. They can detect fires quickly and accurately, even in adverse conditions - such as high smoke or dust. However, these sensors require a direct line of sight for effective detection, which can be challenging in dense forest areas. On the other hand, to increase the efficiency of these sensors, lenses can be used to better capture or direct the light from the flames, expanding the detection area and improving accuracy.

Gas sensors are effective devices for detecting specific gases, such as CO₂ and CO, which are indicative of combustion. Models such as laser diode sensors, the MQ-9, and the MQ-2, as well

as sensors based on carbon nanotubes, are highly sensitive and have low costs. However, these sensors must be constantly turned on for accurate acquisition, resulting in high energy consumption. Furthermore, gas detection can be inaccurate due to the effects of wind, which can disperse gases and make it difficult to pinpoint the exact location of the fire, making its application less effective in open environments.

Atmospheric pressure sensors can detect changes in pressure associated with large fires. However, these sensors may be ineffective during the ignition stage of a fire, as at this stage, there is a small volume of air displacement. Furthermore, the variability of atmospheric pressure in different installation zones is minimal, making real-time monitoring at different points unnecessary by WSN.

Humidity sensors can also indicate susceptibility and presence of fire, whether on the ground or in the air. Therefore, using these sensors can be valuable in a WSN. Once again, the DHT11 and BME280 offer multimodal temperature and humidity measurements. However, the BME280, despite its accuracy, usually has a higher cost, which may limit its applicability in dense WSNs.

Finally, miscellaneous sensors and biosensors, such as acoustic sensors, microwave radiometers, and those that measure chlorophyll fluorescence, offer promising alternative methods for detecting fires. However, many of these technologies are still in the research and development stage, with few commercial options. Additionally, particle and particulate matter sensors can be affected by the seasonality of pollens in forested areas and may require periodic maintenance to ensure their proper functioning.

Therefore, through the 40 papers selected for this study, combining the DHT11 sensor with an IR sensor based on a 760 nm-1100 nm spectrum emerges as the most favorable option for detecting forest fires in large scale by WSNs.

3.2.2 Microcontrollers and Communications

The evolution of wireless communication technologies has been crucial in the development of monitoring and control systems, especially in long-range applications that require low cost and power consumption.

Third (3G) and fourth-generation (4G) mobile networks have significantly improved speed and data capacity. LTE, an evolution of 3G, offers higher data speeds and better spectral efficiency, facilitating high-quality data transmission over mobile networks [118]. With the arrival of 5G, these capabilities have been further expanded, providing even higher data transmission speeds and extremely low latency, essential for critical applications such as real-time monitoring [119].

Despite these advantages, it is essential to note that mobile technologies such as LTE, 4G, and 5G consume significant power, mainly when used in modules that only need to monitor simple numerical data. This can be a considerable limitation for applications that require low power consumption, such as environmental sensors and fire detection systems, where battery life is crucial. In such cases, it is necessary to consider alternative or complementary solutions that can reduce energy consumption without compromising communication efficiency [120].

One promising solution is the use of LPWAN technologies, which fill the gap between mobile (3G, 4G, LTE) and short-range wireless (e.g., Bluetooth, Wi-Fi, and ZigBee) networks. LPWAN

technologies, such as **LoRa**, offer long-range transmission with significantly lower power consumption, making them ideal for **IoT** applications that require periodic transmission of small amounts of data over long distances while maintaining long battery life [121, 122]. In this sense, **LoRa** Alliance proposed the **LoRaWAN**, which is a network that maintains low power consumption and long-range capabilities, defining the network protocol in the **Medium Access Control (MAC)** and network layers [123].

3.2.2.1 Use Cases of Long Range Communication

This type of communication is increasingly used, and it is estimated that by 2024, the **IoT** industry will generate revenue of 4.3 trillion dollars [124]. **LoRaWAN** networks are composed of end devices, gateway and network server in the midst of a rising state. The data is acquired from the modules (end devices) that send the data to the gateway, which in turn relays the messages to the servers through non-**LoRaWAN** networks, such as Ethernet or **Internet Protocol (IP)** over cellular [125].

The **LoRaWAN** communication uses the **Chirp Spread Spectrum (CSS)** modulation, where chirp pulses modulate the signal. It uses an unlicensed sub-gigahertz **Industrial, Scientific, and Medical (ISM)** band that changes in width depending on the region of application, and for Europe, it is 863 MHz to 870 MHz [126]. Within the communication settings, three parameters allow a network adjustment between bit rate and robustness of the transmission: **Bandwidth (BW)**, **Spreading Factor (SF)**, and **Coding Rate (CR)** [127]. The format of the **LoRa** message is subdivided into several layers; the physical layer, which is composed of five fields, the preamble, the **Physical Header (PHDR)**, the **PHDR0_Cyclic Redundancy Check (CRC)**, the **Physical Payload (PHY Payload)** and an error detection tail **CRC**. The **MAC** layer, on the other hand, is a subdivision of the **PHY Payload** field and consists of three fields, the **Medium Access Control Header (MHDR)**, **MAC Payload** and the **Message Integrity Code (MIC)**. Within the **MAC Payload** is the message itself, which consists of three fields, the **Frame Header (FHDR)**, the port and the **Frame Payload (FRM Payload)** [128].

As the adoption of **LoRaWAN** technology continues to grow, its practical applications in **IoT** solutions are becoming increasingly evident. The main application in [129] is the development of connectivity solutions for **IoT** devices using **LPWAN**, focusing on wireless communication technologies such as SigFox and **LoRa**. The modules use **LoRaWAN** with transceivers such as the Semtech SX1272 and the AXEM AX-Sigfox SoC. The work in [128] analyses the energy performance of **LoRaWAN** for several **IoT** applications. Communication is via **LoRaWAN** using the microcontroller MSP430 from Texas Instruments. Transceivers include Semtech SX1272, Semtech SX1276, HopeRF HM-TRLR-LF/HFS, and Microchip RN2483. Whereas in [121], the authors provide an overview of **LoRaWAN** for long-distance communication in **IoT**. It uses **LoRaWAN** with SX1272 and SX1276 transceivers but does not mention specific microcontrollers.

The study in [124] evaluates the performance of **LoRa** in Dakar, Senegal, for multiple **IoT** applications, including environmental monitoring. It uses **LoRa** communication with hardware based on Arduino and Dragino **LoRa** Shield transceivers. In [130], the work models the power

consumption of **LoRaWAN** for monitoring high-voltage power grids. It uses **LoRaWAN** with STM32L073 microcontroller and Semtech SX1272 transceiver.

A paper describes the design and implementation of a **LoRa** network for the **IoT**, intending to create a flexible and low-cost solution for private **LoRa** networks. Applications include smart cities, environmental monitoring, and smart healthcare. Communication is carried out via **LoRaWAN**, and it uses several microcontrollers, including the quad-core Cortex A9 processing unit with a dynamic frequency of up to 1.4 GHz for the **LoRa** Gateway. Additionally, the **LoRa** module consists of two SX1255 transceivers and an SX1301 digital baseband processor [123].

The iLoLeak-Detect is proposed in [131], an **IoT**-based oil leak detection system using **LoRaWAN** technology. The main application is to monitor leaks in oil pipelines to improve operational efficiency and reduce response time to problems in oil and gas companies. The developed system includes sensor modules that measure flow and temperature and is integrated with ARM Cortex M4 microcontrollers. Transceivers used in the system include the Semtech SX1255 for **LoRa** communication. Additionally, the article mentions using the SX1301 digital baseband processor, which assists in data transmission and reception. The work highlights the energy efficiency of the **LoRaWAN** protocol but does not provide specific details about the modules' energy consumption or battery life.

The main aim is to investigate the transmission properties of **LoRa** over seawater, presented in [127]. The paper focuses on a clear line of sight and obstructed path scenarios in the 868 MHz and 434 MHz **ISM** radio bands of **LoRa** for long-distance monitoring and communication in marine environments. It uses several transceivers, including the Semtech SX1272, SX1257, and the HopeRF RFM95. The study mentions that the RFM95 offers a competitive alternative to Semtech transceivers, providing good sensitivity and range. Although the SX1272 is highly sensitive (-140 dBm), the RFM95 stands out for its efficiency and lower cost, making it a popular choice in commercial implementations. Both transceivers proved viable for **LoRa** in line-of-sight and obstructed conditions, with the RFM95 having a slight cost-benefit advantage for specific applications.

3.2.3 Summary Fire Sensors, Microcontrollers and Communication

Early detection of wildfires is crucial for effective management and rapid response, which are essential to controlling the spread of fire. Several sensor technologies have been developed to identify the first signs of fires. This section reviews literature on fire sensors, their technologies, and their applications based on an extensive literature search in academic databases such as Scopus, Web of Science, IEEE Xplore, Science Direct, and Google Scholar. The initial search found many publications, which were refined using specific keywords to focus on fire detection methods using sensors, resulting in a detailed analysis of 40 relevant studies.

Temperature and heat sensors are often used due to their accuracy in detecting temperature variations, with the DHT11 and BME280 sensors highlighted for their multimodal measurements and cost-effectiveness. Optical sensors operating in the **IR** spectrum are sensitive to flame radiation but require a direct line of sight, which can be challenging in densely forested areas. Gas sensors,

such as the MQ-9 and MQ-2 sensors, effectively detect specific combustion gases, but they consume a lot of power and may struggle outdoors. Atmospheric pressure and humidity sensors are also mentioned, each with advantages and limitations. Biosensors and Miscellaneous sensors, such as acoustic and particulate matter, offer promising alternative methods, but many are still in the research phase. Combining multimodal sensors such as the DHT11 with an IR sensor emerges as a practical option for large-scale WSNs for forest fire detection.

Another factor that needs to be considered when designing a forest alert system is communication. Some regions are highly remote, and getting data from there using conventional internet technology (wired or wireless) may not be the best choice. Moreover, in the literature found, none of them shows the collaboration of multiple approaches to increase the reliability of an alert. For example, none of the systems and methods cited produce alerts through the data analysed by WSN based on LoRa and with modules provided with IR, gas, relative air temperature, and humidity sensors.

3.3 5FTH's Design and Assembly

A new technological paradigm has emerged, where digitalization is revolutionizing the industrial, medical, and even educational sectors. Therefore, it is possible to take advantage of and use the same technology to fight fires. This section will focus on the application of this technological revolution in the environmental context, specifically on the development of sensor modules for detecting forest fires. The IoT, which represents the digital interconnection of everyday objects with the internet, emerges as a promising tool in this scenario, offering innovative solutions for environmental monitoring and the prevention of natural disasters.

The relevance of IoT in fighting forest fires is undeniable. According to [98], the integrating of low-cost sensors in forest environments could signify a radical change in the approach to early fire detection, allowing for faster and more effective responses. Therefore, IoT arises in a friendly way as the “thing” could be installed in forests to detect fire ignitions and alert firefighting teams. In this sense, it is proposed in this section to explore the development of specific sensory modules for this purpose, made based on the principles of IoT and the findings of the previous section, which detailed the functioning and effectiveness of different sensors in fire detection and long-range communications.

Based on understanding the comprehensive science of fire sensors, including their detection mechanisms, sensitivity, and limitations (after an in-depth analysis in the previous section). The development of modules must consider the practical application of these concepts through designing and implementing a specific module, named as 5FTH further, within a WSN. The WSN is conceived as an integrated solution for environmental monitoring, where each module sensor has a crucial purpose in the early detection of fires, potentially saving thousands of hectares of forest and preventing significant economic and ecological losses.

Studies highlight the importance of self-sufficient, low-cost sensors for effectively and sustainably monitoring large forest areas [83, 103, 132, 115]. These studies emphasize the need for

devices that not only detect fires efficiently, but are also economically viable for large-scale deployment. Therefore, in a new IoT paradigm, each “thing” will have its own unique characteristics, such as being small, self-sufficient in terms of energy, low cost (a considerable density of sensors will always be necessary), and essentially, having only one sensor (or set of sensors) for a given measurement. In some cases, different types of modules could be complementary in a WSN to achieve common objectives.

Subsequent subsections will detail each stage of the 5FTH development process, from initial conception to final implementation, including challenges encountered and solutions adopted.

3.3.1 Integration of low-cost sensors

Firstly, this approach significantly increases the risks of sensor integration and data interpretation errors. Without fully understanding how each sensor operates separately and together, it may face incompatibility issues and misinterpretations of collected data. Furthermore, the complexity and cost of the project can increase exponentially, because some corrections and adjustments become more complicated when made to an already fully assembled system. Therefore, it was decided to design the modules incrementally in order to understand and adjust, when necessary, the integration of sensors, the behavior of the collected data, and the specific characteristics of time series data.

In this regard, the design and assembly of the initial prototype of the 5FTH module was based on testing the integration of the selected sensors, ensuring that they work harmoniously to capture relevant environmental data. Thus, a simplified approach was chosen, where data would be recorded locally, without the additional complexity of wireless communication. This decision allowed for a deeper focus on the accuracy and reliability of sensor data collection. Therefore, the central element of the prototype is an Arduino Uno, chosen for its ease of use, flexibility, and broad support community. The following sensors were connected to it:

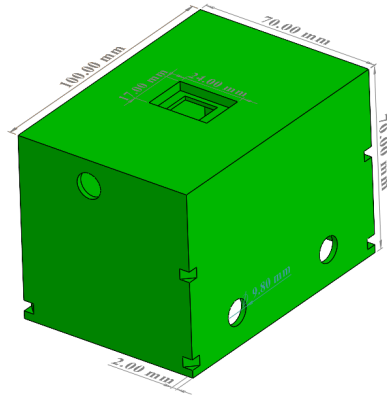
- **Ultraviolet (UV) sensor:** Integrated into the Arduino Uno, it is crucial in detecting ultraviolet radiation in the monitored area with a high potential for forest fires. The UV parameter has a significant correlation between increased UV radiation levels and the likelihood of fire occurrences. The simplicity of installing this sensor on the Arduino Uno, combined with the precision in data capture, makes it an efficient and economical choice.
- **IR Sensors (3 units):** The three IR sensors, coupled to the Arduino Uno, are essential for identifying variations in light, common characteristics in the early stages of fires. Integrating these sensors into the Arduino Uno keeps the overall modules' cost affordable. They enable effective monitoring of sudden light changes in the surrounding environment, providing accurate and timely data critical for rapid detection and response to wildfires.
- **Soil Moisture Sensor:** Connected to the Arduino Uno-based system, it is a key component for monitoring soil conditions. Reduced humidity can be a strong indicator of increased fire

risk. The compatibility of this sensor with the Arduino Uno offers a low-cost and easy-to-implement solution, allowing accurate and regular measurements of soil moisture, essential for assessing fire risks in different environmental conditions.

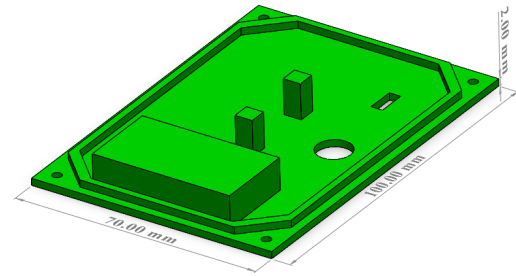
- **DHT11 sensor:** This sensor measures relative air humidity and temperature, two critical factors in fire prevention. Integrating this sensor into the Arduino Uno combines the simplicity of operation with the precision required to monitor these environmental parameters. The low cost and efficiency of the DHT11 ensure that the 5FTH module can provide consistent and reliable data essential for analysing fire-prone conditions.
- **Secure Digital (SD) Card Shield:** Attached to the Arduino Uno, it performs a vital role in recording the data collected by the sensors. Data is saved to the **SD** card every minute, ensuring that the information captured by the sensors is stored securely and accessible. This integration with Arduino Uno offers an efficient and low-cost solution for storing data, which is important for later analysis of conditions that can lead to wildfires.
- **Real-Time Clock (RTC) Shield:** It is responsible for the precise temporal recording of the collected data. This integration allows each sensor data to be accompanied by a timestamp, which is indispensable for time series analysis and understanding trends and patterns in collected data.

The assembly of this prototype was meticulously carried out, strictly following standard electronic design guidelines. In other words, during this process, it was ensured that all installation and configuration recommendations provided by the sensor manufacturers were respected. It is included using official libraries supplied by manufacturers, accessed through official links. This approach not only guaranteed the integrity and optimized functioning of each sensor according to indications already provided by the manufacturers, but also ensured the compatibility and efficiency of the system as a whole.

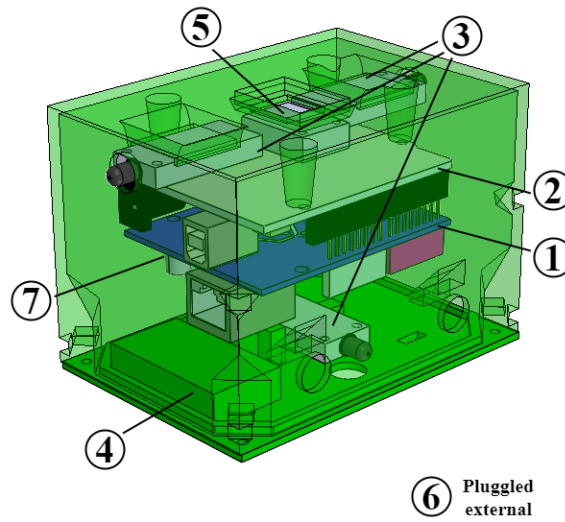
Furthermore, to accommodate all components safely and functionally, the prototype was enclosed in a specially designed box produced by 3D printing. The design of this box was preceded by careful planning, in which all components were meticulously drawn and inserted into the box design. The final assembly of the project can be seen in Figure 3.3. The dimensions of base, available in Figure 3.3a, are 100mm × 70mm × 70mm. The lid has the following dimensions: 100mm × 70mm × 2mm, as Figure 3.3b illustrates. Because it has a concave shape, the thickness of the base walls of the box is 2mm. Additionally, Figure 3.3c describes the components. The central element is the Arduino Uno (1) which, through **SD** card (2), reads and saves data from three **IR** radiation sensors (3), temperature and humidity sensors (4), **UV** index (5), and soil moisture (6) at the time informed by the **RTC** (7).



(a) Base of the prototype module.



(b) Lid of the prototype module.



(c) Arrangement of Arduino Uno (1), SD card Shield (2), IR Sensors (3), DHT11 sensor (4), UV sensor (5), soil moisture sensor (6) and RTC (7).

Figure 3.3: Desing of the prototype module.

After verifying in the simulation of the designed project that all sensors are well accommodated, the enclosure was printed. Before inserting each of the previously described components, an acrylic piece measuring $17\text{ mm} \times 24\text{ mm} \times 2\text{ mm}$ (from Figure 3.3a) was glued to the top of the module. In this way, the UV sensor can perform measurements without any damage that may occur with climate changes when the module is fixed outdoors. The same consideration was taken for the holes where the IR sensors are determined to be located. However, for this case, a spherical Fresnel lenses for Passive Infrared (PIR) with 10mm diameter were installed. Then, all devices were placed inside the box, the final assembly can be seen in Figure 3.4.

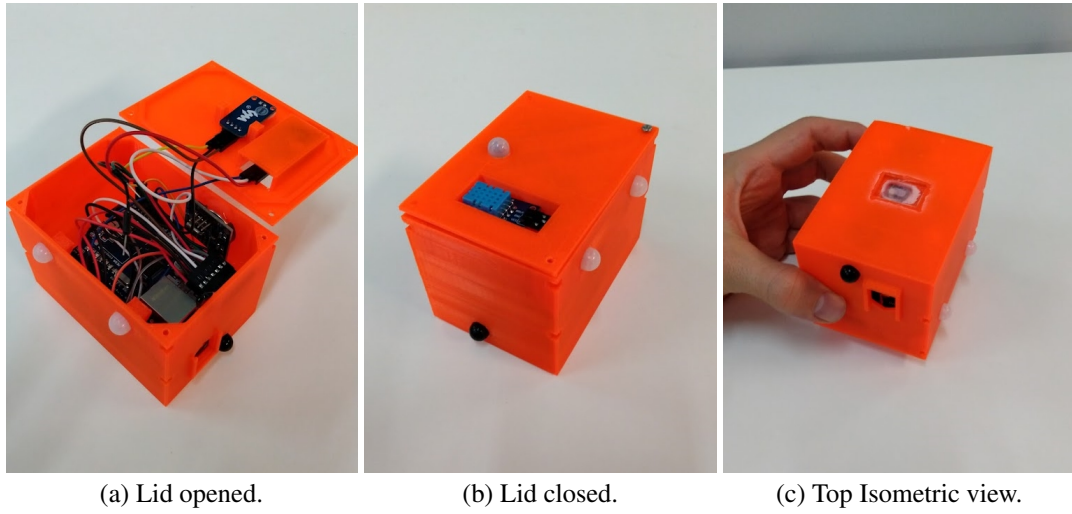
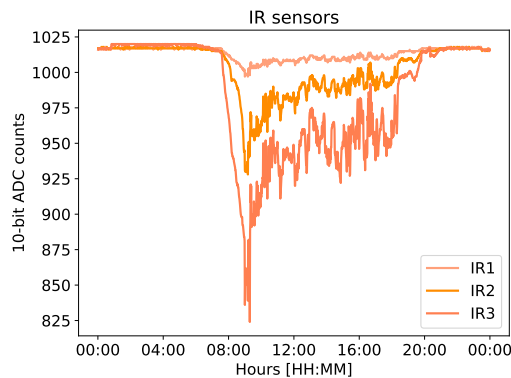


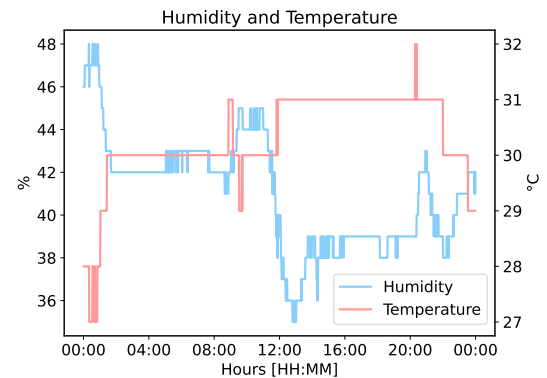
Figure 3.4: Final assembled prototype module.

Once assembled, the prototype underwent data acquisition testing in a controlled laboratory environment. These tests were essential to validate the functioning of the set of sensors before being installed in an external environment. As shown in Figure 3.4c, the module was positioned on a small table and supplied by a power bank via **Universal Serial Bus (USB)**. This configuration allowed extensive testing to be carried out, including collecting environmental data and monitoring the performance of the soil moisture sensor, which was installed in a pot to simulate real-life conditions.

The module remained in the laboratory table for approximately three days, continuously collecting data every 60s. The data collected by this prototype was stored on the **SD** card, allowing for detailed subsequent analysis. After the testing period, the data was extracted from the **SD** card and used to generate graphs, shown in Figure 3.5. The figure is divided into four parts, each corresponding to a specific sensor type. To ensure a clearer analysis, only the 24h of the second day are displayed, as the first and third days contain incomplete data due to the system's activation and shutdown.



(a) IR data.



(b) Humidity and Temperature.

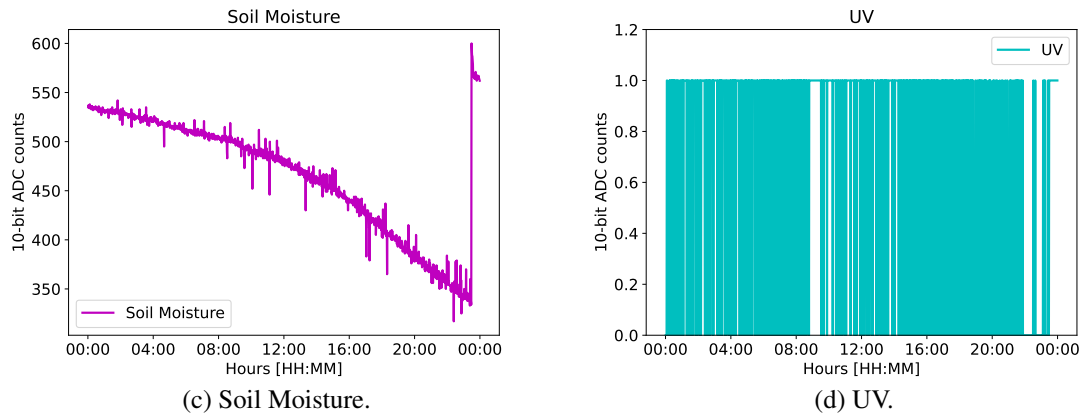


Figure 3.5: Data collected of the prototype module after 24 hours at controlled environment.

Figure 3.5a - IR Sensors: This part of the graph shows the values obtained by the IR sensors. It is essential to highlight that these values are dimensionless, as they are direct readings from the Arduino Uno's 10-bit Analog-Digital Converter (ADC). This data provides critical discernment into light variations in the test environment. This is because the sensor was fixed next to a window, and therefore, the disturbances shown in the graph are made by sunlight. These disturbances begin shortly before 8h00 (early morning) and last until approximately 20h00 (early evening). It is possible to establish the minimum and maximum value of each IR sensor, where values have ranged from 0 to 1023 (values from Arduino Uno analog input). In this regard, when the sensor receives a high light intensity, the values are close to 0. On the other hand, the values tend to be 1023 when the sensor is exposed to a low light intensity. The sensor's sensitivity varies with distance, intensity, and radiation volume.

Figure 3.5b - Humidity and Temperature: This graph shows the relative humidity and temperature values collected by the DHT11 sensor. When humidity data has a downward behavior, temperature data has an upward behavior. This behavior is already expected due to the related works found in the literature [133]. Furthermore, the data has low noise, indicating the stability of the DHT11 sensor.

Figure 3.5c - Soil Moisture: This plot illustrates the soil moisture data from the spot. Similar to the values from the IR sensors, these readings are dimensionless, coming from the Arduino Uno's 10-bit ADC.

In this case, when the soil becomes dry, the acquired data are close to 0, and when the soil turns wet, the values are close to 1023. A declining variation was noted throughout the day. However, there is an abrupt variation shortly before the end of the 24h. This abrupt variation may be due to the laboratory's cleaning service, where employees, in addition to cleaning at night, also water the plant pots of the laboratory.

Figure 3.5d - UV index: The fourth figure shows a graph with the ultraviolet radiation values captured by the UV sensor. This data, also dimensionless and read through the 10-bit ADC, indicates UV radiation exposure during the test period. As the data never had a variation more significant than 1 and less than 0, this data indicates that the sunlight penetrated by the translucency

of the window was not enough to be detected by this sensor (even causing a certain disturbance in the IR data).

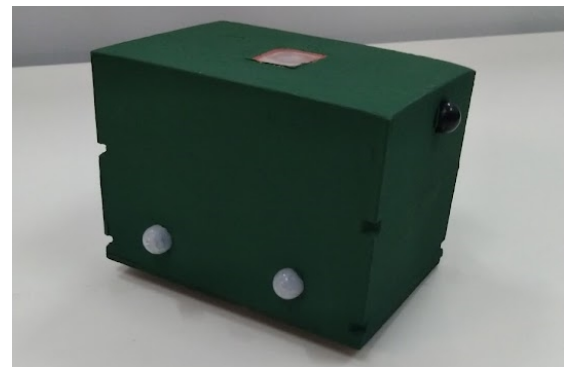
Next, the module was fixed to a trunk tree in a garden, as seen in Figure 3.6a, an ideal place to test its effectiveness and resistance in natural conditions of use. For this test, the module was programmed to collect data every two seconds, a significantly shorter interval than the previous test. The objective of this configuration was to stress the data acquisition system, testing its capacity and reliability under conditions of intense and continuous use. Note that the orientation of the sensors is free, that means there are no barriers to the capture of IR spectra.



(a) Prototype module fixation.



(b) Consumption of the prototype module.



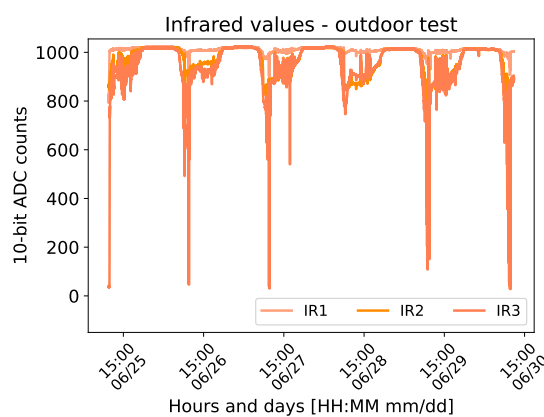
(c) Prototype module after painting process.

Figure 3.6: Preparation of the prototype module, consumption measurement, and fixation on the trunk tree for testing in an outdoor environment.

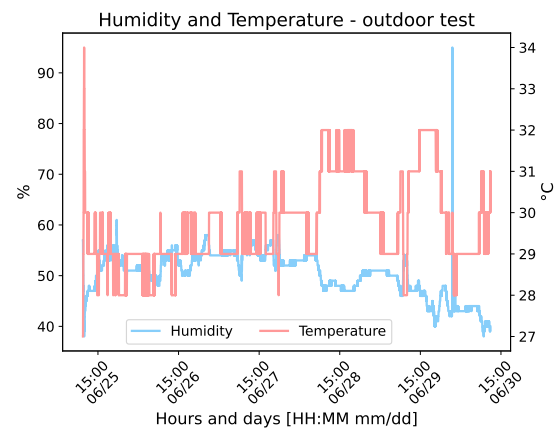
A detailed analysis of the graphs in Figure 3.7 provides a comprehensive understanding of sensor performance in outdoor conditions. The graph in Figure 3.7a corresponds to the recordings made by IR sensors, which revealed intense noise during daytime hours – a notable contrast to the more stable data collected in a laboratory environment. This variation can be attributed to increased exposure to external light sources and natural environmental fluctuations. Despite the noise, the sensitivity of IR sensors can be considered an advantage, especially for quickly detecting nearby fires. The data consistently shows that all significant changes in ambient light were captured by the sensors, as evidenced during daytime hours in the graph. On the fourth day, although a less pronounced drop in the data was observed, it still indicates sensitivity to light, which could be explained by the possible presence of clouds on that day.

The graph combining temperature and humidity data, Figure 3.7b, showed a stable and predictable pattern, as was expected. During the day, an increase in temperature and a drop in relative humidity were observed, while during the night, the temperature decreased and humidity increased. A sudden peak in the humidity line can be observed on the last day, which can be explained by the irrigation system. This cyclical and predictable behavior together with the quick response when the irrigation system is activated, confirms the reliability of the DHT11 sensor in external conditions, reinforcing its usefulness in environmental monitoring.

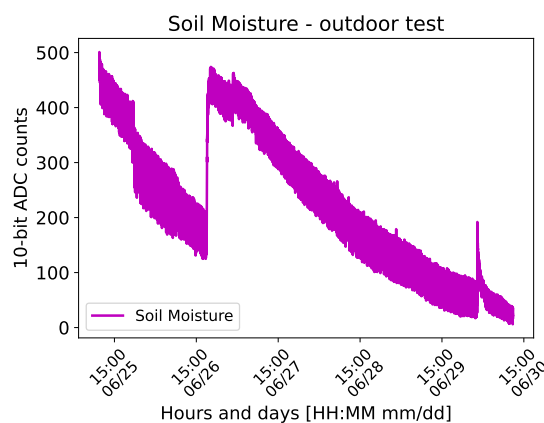
As for soil moisture, the graph in Figure 3.7c showed a significant stability. However, the data revealed a slower sensor response to soil conditions, characterized by a gradual and prolonged decrease in moisture. This behavior can be attributed to the intrinsic characteristics of the soil, which tend to retain or lose moisture more gradually. Finally, in Figure 3.7d, there are some inconsistencies in the data that generate the UV graph. The discrepancies are from records that detect some level of UV during the dawn, precisely at 3h00 in the third day. This behavior is atypical and unexpected, given the absence of sunlight during this period.



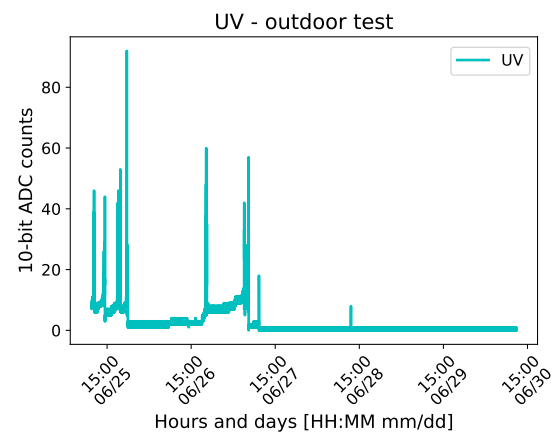
(a) IR sensors - outdoor test.



(b) Humidity and Temperature - outdoor test.



(c) Soil Moisture - outdoor test.



(d) UV - outdoor test.

Figure 3.7: Data collected of the prototype module after five days at outdoor environment.

After carefully analysing the collected data and the overall performance of the sensors integrated into the prototype module, some essential guidance for the subsequent phases of project development and improvement can be obtained. One of the critical aspects identified is the module's energy consumption. It was observed that continuous consumption of 200mA is unacceptable for efficient and sustainable operation. This fact becomes evident when estimating the module's autonomy using an 18650 lithium-ion battery with a nominal capacity of 3350mAh. The theoretical autonomy estimate, based on the battery capacity and the module's consumption, results in an estimated 16h and 45min.

However, this estimate does not consider the voltage decrease during discharge. According to [134], the minimum operating voltage for 18650 batteries is approximately 2.8V to ensure system safety and efficiency. Additionally, extreme ambient temperatures can significantly impact battery performance by increasing internal resistance and reducing available capacity. Therefore, the module's online duration could be even shorter. While adding more batteries could extend the operational duration, it would also increase the cost and size of the device, which is not desirable for this project.

Therefore, it is essential to explore intelligent approaches to reducing this consumption. It may include implementing power management algorithms that activate sensors only under specific conditions or at scheduled intervals, significantly reducing power consumption.

While assembling the sensors and fixing the module to the tree trunk, specific challenges were faced with the UV and soil moisture sensors. The first challenge was installing the UV sensor in the module box, which required cutting and gluing the acrylic and, therefore, it proved to be complex and laborious for scalable applications. The second challenge was faced during the soil moisture sensor installation in the ground, in which its connection to the Arduino Uno was via a long cable, representing considerable logistical challenges.

Considering the need to replicate this process in hundreds of modules, it is concluded that installing the UV and soil moisture sensors is impractical for large-scale production. Furthermore, the instability of the data collected by the UV sensor in external conditions reinforces the decision not to include it in future module improvements. Similarly, despite the stability and adequate behavior of the soil moisture sensor data, it was decided to exclude it from the project. Moreover, the intrinsic characteristics of the soils, which respond slowly to sudden changes in the environment, limit the usefulness of this sensor in an application focused on rapid-fire detection.

However, information about UV and soil moisture is extremely important when the theme is forest fires. Therefore, this data can be acquired through projects that make data available via satellite, offering an alternative and efficient means of obtaining this information without any dedicated sensors.

Therefore, the module's future configuration will focus on IR sensors and the DHT11 sensor (humidity and temperature). These sensors' rapid sensitivity and reliability were determining factors in the decision. They offer an ideal balance between efficiency, cost, and accuracy for project requirements. Furthermore, the module must have a dedicated, internally secured battery to ensure long-lasting, uninterrupted operation. Moreover, data transmission must be carried out

efficiently, with low cost and energy consumption, without compromising reliability. This set of decisions and strategies aims to optimize the module's performance, making it a viable and effective solution for detecting and monitoring forest fires. The following subsection will be dedicated to designing and implementing these improvement topics.

3.3.2 Development of wireless detection modules

The development process must emphasize the importance of meeting multiple operational and technical constraints. Compact size is a primary consideration to ensure the module can be installed discreetly without significantly impacting the natural environment. Battery autonomy also deserves special attention to ensure the prolonged and reliable operation of the device, which is essential for long-term monitoring in forest areas.

Low cost, both in implementation and communication, is the most significant value to be achieved at all times. Through this approach, it is possible to ensure that the implementation is economically viable for large-scale deployment, enabling comprehensive coverage across extensive forest areas. Furthermore, flexibility for adding new sensors was incorporated into the design, allowing the module to adapt and evolve according to emerging technological needs and advances.

Figure 3.8 illustrates in detail the fundamental components considered during the module development. This figure not only highlights the technical aspects of the device but also underlines the harmonious integration of diverse functionalities, reflecting the design's commitment to efficiency, adaptability, and sustainability in environmental monitoring applications.

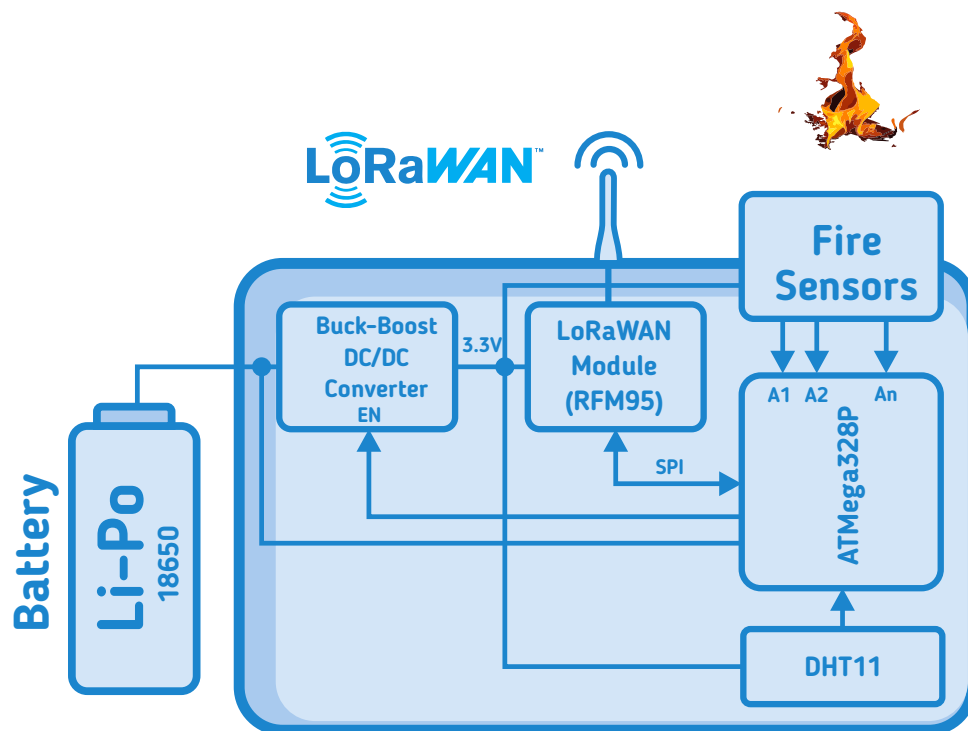


Figure 3.8: Simplified module's layout [135].

A set of components was selected to ensure the device's efficiency, durability, and functionality in forestry environments. The core elements were:

- ATmega328P microcontroller
- IR sensors
- DHT11 sensor
- LoRaWAN RFM95 transceiver
- Buck-boost converter
- Li-Po 18650 battery.

Each component was selected after a detailed analysis of its characteristics and benefits, as described next.

3.3.2.1 The module's core

Although scientific research has identified several suitable microcontrollers for the project, it was not possible to compare them due to the development of the module being inevitably impacted by the global electronic components crisis that occurred between 2021 and 2022 [136]. In this sense, during this period of shortage of electronic components, reported by several media, many models of microcontrollers were absent on the market, making it impossible to compare possible candidates to be installed as the central core of the modules.

At the same time, the Polytechnic Institute of Bragança (IPB) already had a stock of the ATmega328P, acquired due to the excellent applicability of this microcontroller in educational activities. The decision to use the ATmega328P was, therefore, strategic, considering the immediate availability of the component and the pandemic restrictions that affected the availability of components worldwide. Furthermore, it is essential to highlight that the choice of the ATmega328P does not harm the final performance of the module, as it effectively meets the project requirements in terms of processing, power management, and connectivity.

The central element of the module is the ATmega328P [137] microcontroller, a hardware widely used by developers looking for excellent cost-benefit in their projects. As mentioned in [138], the popularity of the ATmega328P is partly due to its compatibility with the Arduino platform, which significantly facilitates programming and development. The microcontroller configuration can be performed via RX and TX pins using a bootloader (and USB converter).

The choice of the ATmega328P to integrate the project modules was based on several of its notable features. One is the availability of sufficient analog and digital input/output (I/O) ports to support the desired sensors and interfaces. It includes compatibility with the Serial Peripheral Interface (SPI) protocol, which is essential for communicating with the LoRa module. Furthermore, the ATmega328P's power supply range, between 2.7 V and 5.5 V, and it is perfectly compatible

with Li-Po battery voltages, which is vital for the efficient and safe integration of the module's power system.

A noteworthy aspect of designing energy-efficient systems is the power consumption of the microcontroller. It is widely recognized that microcontrollers consume less power when operating at lower frequencies. Therefore, selecting an oscillation crystal for the microcontroller is a crucial decision. According to the ATmega328P [137] datasheet (on page 312), the microcontroller can be configured with a 4.0MHz crystal. This configuration allows the ATmega328P to operate with a power supply range of 1.8 V to 4.0 V. Choosing a 4.0MHz crystal is strategic as, in addition to meeting power requirements, it is also suitable for maintaining reliable and efficient microcontroller operation under different power conditions.

Additionally, it is important to consider the specifications of the circuits connected to the microcontroller's I/O pins. According to the ATmega328P [137] datasheet (on page 317), for a voltage to be recognized as a high level on the I/O pins, it must be at least 70% of the microcontroller's supply voltage, while the maximum allowed voltage can be up to 0.5 V above the supply voltage. This characteristic is crucial and must be considered when choosing the battery and designing the sensor circuits connected to the microcontroller, ensuring that all digital inputs function correctly within the voltage specifications.

By selecting the ATmega328P as the core module, it can avail of its flexibility, energy efficiency, and the wide range of resources available for programming and development. As a result, this microcontroller can be used to ensure that the module is operational, reliable, cost-effective, and adaptable to the ever-changing needs of environmental monitoring.

3.3.2.2 Module's sensors

The flexibility of integrating sensors with analog and digital interfaces, respecting the limits of the ATmega328P, allows the system to be adaptable and expandable. In this sense, a particular emphasis was placed on selecting sensors that maximize the system's efficiency, accuracy, and adaptability in forest environments. For this project, the BPV10NF IR sensors [139] and the DHT11 sensor [140, 141] were chosen as elements to perform forest ignition detection.

IR sensors are essential for the early detection of forest fires, as they are capable of identifying radiation variations typical of an ongoing fire. The choice of these sensors is based on their quick response and high sensitivity to radiation changes, making them essential components in the module. According to the sensor's datasheet, it is possible to detect the light spectrum in the range from 780nm to 1050nm at a angle of $\pm 20^\circ$. Such detection can be done by reading the voltage variation in the circuit, which can be obtained through a voltage divider with a 10k Ω resistor, displayed in Figure 3.9.

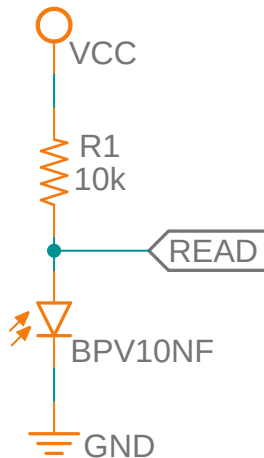


Figure 3.9: Example of a circuit for read data from the BPV10NF sensor.

The choice of the $10\text{k}\Omega$ was considered based on recommendations from the sensor datasheet, which indicates that the sensor should be used in series with resistors within the range of $5\text{k}\Omega$ to $15\text{k}\Omega$ [139]. According to the datasheet, the resistor value directly influences the quality of **IR** measurements. This is because resistors with lower values (closer to $5\text{k}\Omega$) tend to consume more current, which can cause significant oscillations when reading **IR** intensity values. On the other hand, resistors with higher values (close to $15\text{k}\Omega$) result in a lower current, consequently making measurements more stable. However, high values of resistors may affect the sensor sensitivity. In this regard, to achieve a balance between circuit energy efficiency and signal attenuation, the $10\text{k}\Omega$ resistor was selected. Thus, it is expected to be a data read with regular oscillations while maintaining energy efficiency in the circuit.

The DHT11 sensor was selected to measure relative humidity and temperature, both critical factors in monitoring conditions prone to forest fires. This sensor stands out for its low cost, digital output, low installation complexity, and relatively small measurement tolerance ($\pm 5\%$ humidity and $\pm 2^\circ\text{C}$ temperature), as highlighted in [142]. These characteristics make the DHT11 an ideal option for the module. In addition to balancing effectiveness and accessibility, it can also be considered a multimodal sensor (a device capable of capturing information from several sensory modalities). In this module version, a model of DHT11 for **PCB** was chosen, that is, the DHT11 is directly soldered in the **PCB** without any boards or shields that it is typically known.

Fusing these sensors into the module allows collecting environmental data fundamental for quickly identifying a fire starting around the module's attachment. After data acquisition, processed by the ATmega328P, this data is transmitted via **LoRaWAN** by the RFM95. This approach ensures that the module not only effectively detects early wildfire conditions, but also continuously monitors the environment, providing valuable data for preventing future wildfires.

3.3.2.3 LoRaWAN RFM95 transceiver

Among the available transceiver devices on the market, RFM95 was selected due to its compatibility with ATmega328P. The RFM95 module, developed by HOPERF, is a low-power wireless transceiver for LoRa transmission [143]. In this context, choosing the RFM95 transceiver as the module's communication component is a strategic decision aligned with the objectives of energy efficiency, transmission range, and reliability in challenging environments such as forests. This transceiver has the capability to work in stand-by mode, as presented in Figure 3.10. It is beneficial for optimizing power consumption when the modules do not require transmission. Moreover, this transceiver stands out for its ability to transmit data over long distances with low power consumption. It is an essential feature in areas where traditional connectivity is limited or non-existent.

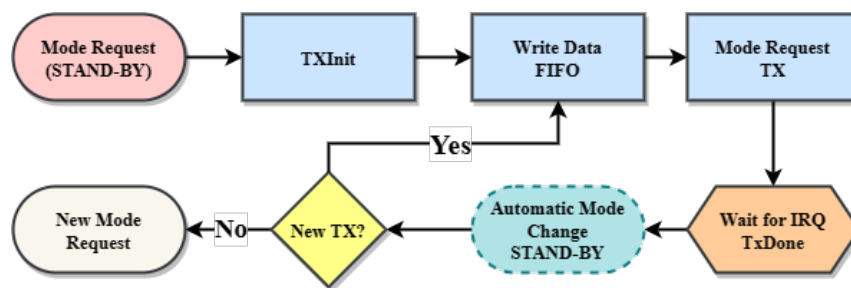


Figure 3.10: Transmission process of the RFM95/RFM96 applied in each module. Adapted from [143].

According to manufacturer [143], the RFM95 has a LoRa-based modem with immunity to interference, which is crucial for minimizing current consumption during transmissions. This feature is particularly significant in forest environments, where the presence of natural obstacles such as trees and topographic variations can interfere with the signal. The sensitivity over -140dBm of the RFM95, as mentioned in [143], makes it suitable for use in several industrial and agricultural sectors, where robustness and transmission reliability are a priority. In this sense, the RFM95 can work with SF ranging from 6 to 12, an essential feature when the module is placed in a coverage area by a given LoRaWAN Gateway. In other words, with this transceiver, it is possible to change the SF according to the distance that there will be communication with the Gateway.

This device is connected to the ATmega328P microcontroller via an SPI port and powered by 3.3 V supplied by the DC-DC converter (this is activated after the ATmega328P microcontroller exits power saving mode, as described in [29]). This power management approach ensures that module power consumption is kept to a minimum, i.e., the RFM95 only operates when necessary, further optimizing system power consumption and extending the device's battery life.

After being activated, the RFM95 receives the data collected by the sensors from the ATmega328P and transmits it to the LoRaWAN Gateway. The transmission is carried out at 868 MHz, which is commonly used in wireless communication applications in Europe. Therefore, using this transceiver not only maximizes energy efficiency, but also ensures that sensor data is sent reliably and securely, even in challenging conditions found in forest environments.

3.3.2.4 Power supply

This module is powered by an 18650 lithium-ion cell battery that will power the entire system. The battery can supply a voltage of up to 4.1 V, and it can affect the operation of other components that operate in the 3.3 V region. A buck-boost converter is applied to maintain the operating range of all sensors and transceivers used in 3.3 V. Therefore, this converter ensures that no sensor changes its acquisition values during battery discharge, and it can protect the RFM95 transceiver over high levels of voltage. In this way, the 18650 battery is directly connected to the converter and the ATmega328P, and the output of the DC-DC converter powers everything else. Figure 3.11 presents the block diagram of the power supply system.

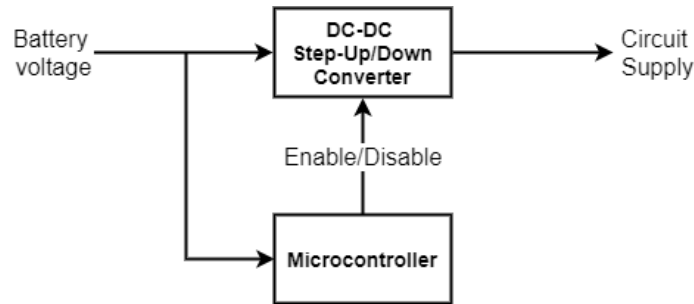


Figure 3.11: DC-DC converter diagram used in each module.

This converter has an Enable pin that is used to shut down all the board components (except the microcontroller). The sleep function of the microcontroller is activated to save energy consumption since this module will be placed without grid connection [29]. Thus, before entering the energy-saving mode, it is necessary to turn off the converter through this digital pin. When it is needed to collect and send data, it is required to activate this pin.

Another surplus value of this DC-DC is obtained when the battery voltage reaches levels below 3.0 V, and the converter stops working (it was checked in the lab by decreasing the voltage step by step). In this case, the sensors and the transceiver will not work either, but the ATmega328P will continue because it is still in the operating voltage range. To avoid this case, one of the ATmega328P's analog ports is used to measure battery levels before performing the data collection and sending process. If the battery voltage is below 3.15 V, the sensor module sends a signal to the control center that the battery level is critical, then the microcontroller turns off the converter and the `Power Down` mode is triggered. In this case, the microcontroller will remain in `Power Down` mode and will no longer measure the sensors' values.

When the battery voltage reaches values around 3.0 V and 3.3 V, a particular situation occurs: the battery voltage is above the microcontroller's `Power Down` mode, but below the minimum operating voltage of the DC-DC converter. In this condition, the ATmega328P microcontroller continues to be powered directly by the battery, while the circuits connected to the *I/O* pins are powered by the converter's fixed 3.3 V output. This voltage difference between the microcontroller's power supply and the *I/O* circuits must be considered (as specified in the previous subsection). In

this case, the maximum difference will not exceed the recommended value, that is, up to 0.5 V above the microcontroller's power supply voltage.

In the opposite scenario, when the battery voltage approaches 4.1 V, the DC-DC converter maintains a fixed output of 3.3 V to the sensors and other circuitry, respecting the recommendation of 70% difference. This ensures that whenever there are variations in the battery voltage, the sensors and microcontroller operate within specifications, avoiding unwanted behavior or operational failures.

Schematic diagrams

The complete circuit schematic, including Gerber files and the detailed project with all connections for each component, can be viewed in Appendix A. However, for the purposes of clarity and convenience, this subsection is dedicated to presenting the pin connections of the ATmega328P to the sensors, as well as providing a list of components used in the assembly of each PCB.

3.3.3 Assembly of wireless modules

This subsection is dedicated to detailing the assembly and electronic configurations of two different versions of the wireless module. Therefore, here, the evolution of the wireless module is covered in detail, focusing on the fundamental differences between its two versions – in terms of assembly and electronic configurations.

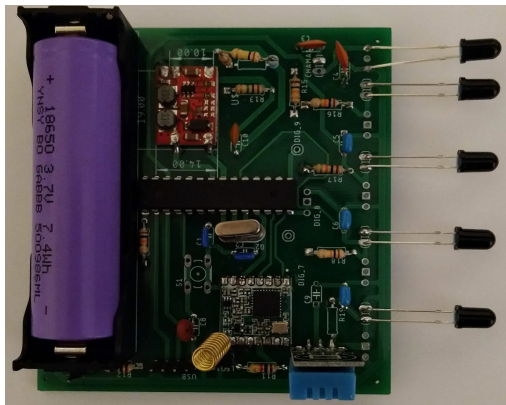
The first version of the module (5FTH 1.0) was an important step marked by experimentation and discovery. This phase was characterized by implementing available ideas and technologies, focused on functionality and proof of concept. The second version (5FTH 2.0), however, results from substantial improvements, where every aspect of the module has been re-evaluated and refined based on practical experience and technical feedback.

The development of these versions highlights the importance of adaptability and continuous evolution in technology projects, especially those applied in dynamic and challenging environments such as forests. Below, each version is detailed, highlighting the motivations behind its characteristics and the improvements implemented.

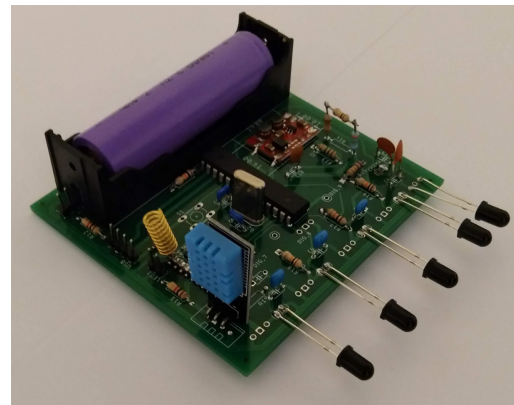
5FTH 1.0

The wireless module's first PCB version was assembled. It can be viewed through different angles in Figure 3.12, with ATmega328P as the core. Figure 3.12a indicates the sectors distributed within the PCB, where the DC-DC converter is placed as close as possible to the battery and the microcontroller.

Figure 3.12b, there is the peripherals that need to be arranged on the edge: the DHT11 needs to have an airflow to perform the measurement, the RFM95 communication antenna to not suffer signal blockages, and the IR sensors not be covered by other components. This step is decisive to check for possible failures between the listed elements. Therefore, the tests performed during this stage are preented in [29].



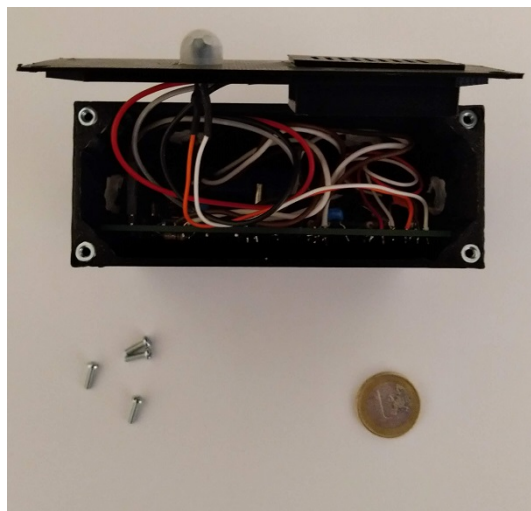
(a) Developed PCB with the components (Top view).



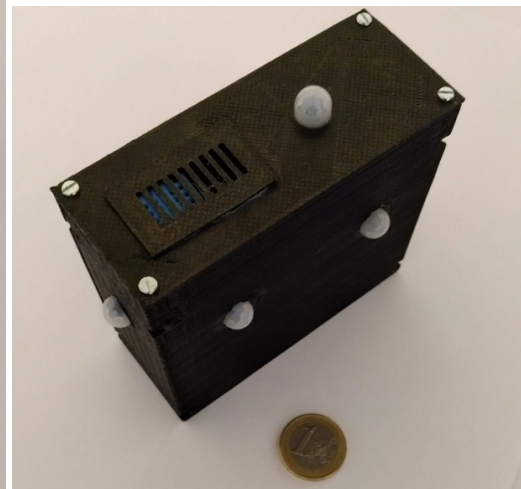
(b) Developed PCB with the components (Side view).

Figure 3.12: Developed sensor module 5FTH [29].

In this version, the **IR** sensors were arranged in a line, which, although functional, reduces the ability to detect the environment omnidirectionally. It was proved after the test with some candles in the laboratory, fully described in [29]. It was observed that using **IR** sensors in a line generates data with similar values because they are all pointed in the same direction. Then, regarding this acknowledgment, another assembly was performed with the **IR** sensors placed in four faces of the box: one sensor on the bottom, left and right sides, and two sensors on the front side. The **IR** sensors were fixed on these four sides of the box, except the top and back faces, because the idea is to attach these 5FTH modules to the tree trunks (the rear face is blocked by the trunks) and also because it is not necessary to point any sensor to the sky (upper face). To achieve that, before rebuilding the **PCB**, the five **IR** sensors were connected via cables, as shown in Figure 3.13.



(a) Top view with wired IR sensors.



(b) Isometric view from the bottom.

Figure 3.13: Developed sensor module 5FTH in the 3D printed case [135].

To evaluate this assembly, nine modules were fixed to tree trunks near the **IPB** campus. Eight of them were installed in pairs and in different locations, such as close to a building, on an orchard,

close to a river, and in an area with high forest density. The ninth module was installed in an open field. The full description of this test can be found in [135].

Using wires to fix the sensors around the box's body added complexity to the design and potential vulnerability to the elements. The same happened with the DHT11 sensor, which is crucial for measuring humidity and temperature; it had to be placed externally because the speed of data collection was compromised due to limited air circulation. Moreover, during the fixation of the modules, it was noticed that it was difficult to tigh the nodes on the tree trunks due to the limitation of the enclosure. For example, a metal wire can not hold the box well in certain positions due to a lack of holes (or other features).

The 5FTH 1.0 also featured a fixed crystal for the ATmega328P, which imposed restrictions on firmware tuning. In this version, during a firmware flash, two options could be used: the first is flashing the microcontroller by removing it from the PCB, and the second is changing the crystal from 4.0MHz to 16.0MHz. Both options could be laborious in case of future firmware improvements when the module is installed in a tree trunk.

Also, status LEDs were included to facilitate monitoring during the testing phase, but this increased energy consumption. Even using pinheads it is not necessary to maintain the LEDs, then it was removed due to avoid the increase components and complexity during assembly and soldering. Furthermore, the distribution of digital I/O across the board proved impractical for other applications, limiting the module's versatility.

Concerning all observations described, the PCB was rebuilt and focused on attending to those requirements. The following subsection will describe each design improvement.

5FTH 2.0

After a detailed analysis of the results obtained with the 5FTH 1.0 and considering the challenges identified during field tests, a bunch of improvements were incorporated into the development of the 5FTH 2.0. This new version aims to overcome previous limitations, improving the module's functionality, efficiency, and practicality, which makes it more adapted to the demands of forest environments.

The layout of the five IR sensors has been optimized to acquire the environmental data collection regarding their vision efficiency. In the 5FTH 2.0, the sensors are distributed to efficiently cover all relevant angles without needing external wires, minimizing vulnerability to the elements and simplifying PCB assembly. Therefore, the PCB design was changed to allow IR sensors to be soldered directly to the board. Then, the IR sensors are positioned on the PCB to cover the four box interest sides for better identification of the appearance of fire ignitions. Furthermore, the DHT11 sensor remains integrated internally and close to airflow.

The electronic design also received significant improvements in the clock frequency source. A switching circuit between two crystal models was introduced, which allows easy exchange between crystals of different frequencies. The additional circuit makes it easier to update the firmware without requiring a direct physical intervention on the module components. Additionally, the option

to install indicator LEDs, although useful in the testing phase, was removed to reduce the final cost of the **PCB**.

The organization of electronic components and unused digital **I/O** has been revised to maximize efficiency and usability. The digital **I/O** ports are grouped closer in this new version, with 3.3 V and GND pads. This grouping facilitates the 5FTH module's flexibility in other applications and simplifies assembly and maintenance. The final assembly of the 5FTH 2.0 design is shown in Figure 3.14. This **PCB** has 86 mm × 76 mm × 22 mm after being fully assembled.

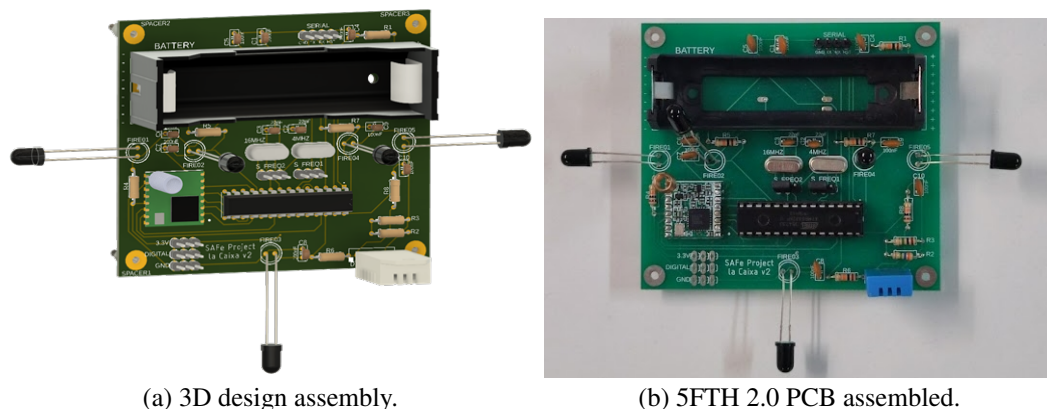


Figure 3.14: The PCB's 5FTH 2.0 in 3D simulated view and final assembled.

In designing the 5FTH 2.0, the enclosure selection was also essential, especially given the environmental conditions in forested areas. To ensure maximum protection and durability of the **PCB**, the RND 455-01087, a high-quality model produced by RND Components, was chosen. This robust plastic case has dimensions of 90 mm × 115 mm × 55 mm, offering adequate space to accommodate the necessary electronic components without compromising portability or design efficiency. It is manufactured with dark gray **Acrylonitrile Butadiene Styrene (ABS)**, which guarantees the module's integrity against physical shocks and abrasions and provides a professional and discreet appearance. More information about the enclosure can be obtained in Appendix B.

The RND 455-01087 enclosure has an IP67 protection rating. This standard indicates that the case's material is entirely dustproof (even in joint base-lid) and capable of being immersed in water at a depth of up to one meter for up to 30 min. Both features benefit the applications in forestry environments, where exposure to dust, humidity, and heavy rain is a constant concern. In this application, the IP67 protection ensures that the electronic components inside the enclosure remain dry and free from environmental disturbances, keeping the module operational even in the harshest conditions. Moreover, the trustable support files provided by the company allow the correct fitting of the developed **PCB** through the preview by assembly simulation. In addition to improving physical resistance, the box has features that allow for safer and more versatile attachment to tree trunks, eliminating the previously observed issues.

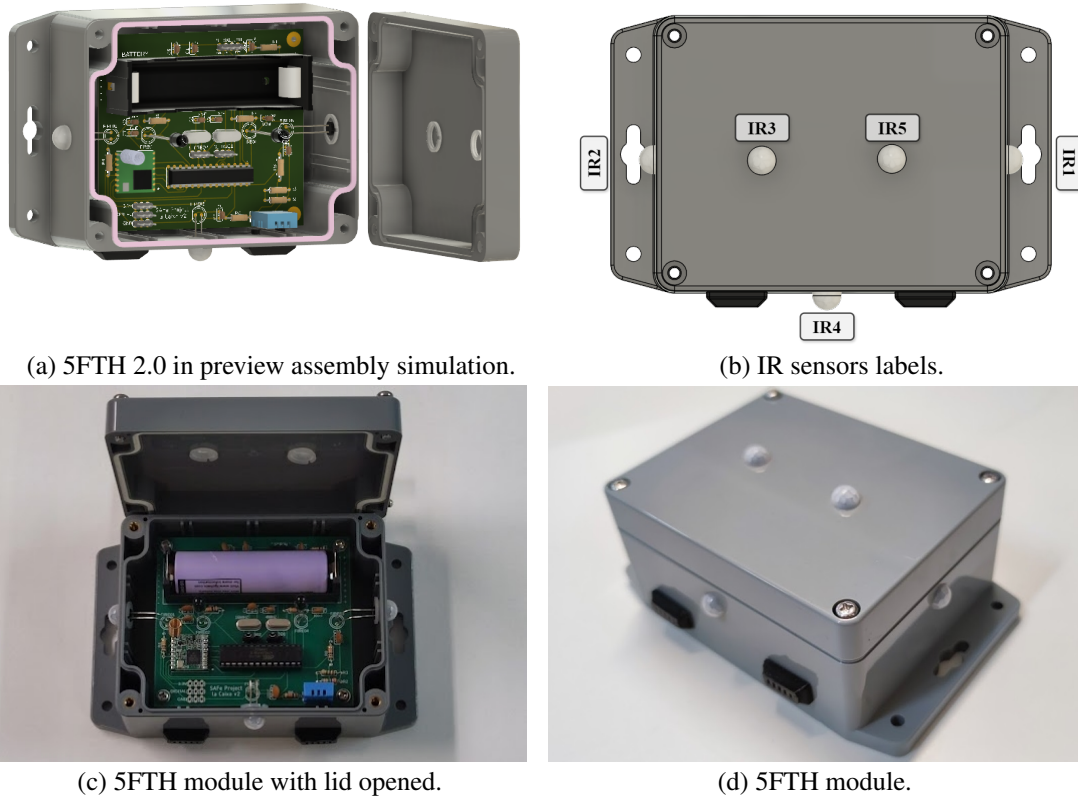


Figure 3.15: The 5FTH 2.0 in 3D preview assembly simulation and final assembled.

The module's layout and detection field, fixed to a tree trunk, were estimated by simulation. The simulated tree has a radius of 0.25 m and a height of 5 m. The module was fixed at a height of 1.8 m from the ground. Additionally, the surrounding grass was designed as a square area of 10 m², providing a more realistic environmental context for the simulation.

Each IR sensor's sensitivity was modeled with a cone that was 5 m meters long (used specifically for visualization in the simulation). As the angular sensitivity of each sensor is defined in the datasheet as $\pm 20^\circ$, the opening angle in the simulation is 40° . Then, the cones were drawn for each sensor, illustrating the field of view and the area covered by each one. The entire simulated scenario can be viewed isometrically in Figure 3.16. It is worth noting that the effects of the Fresnel lens are not considered in this simulation.

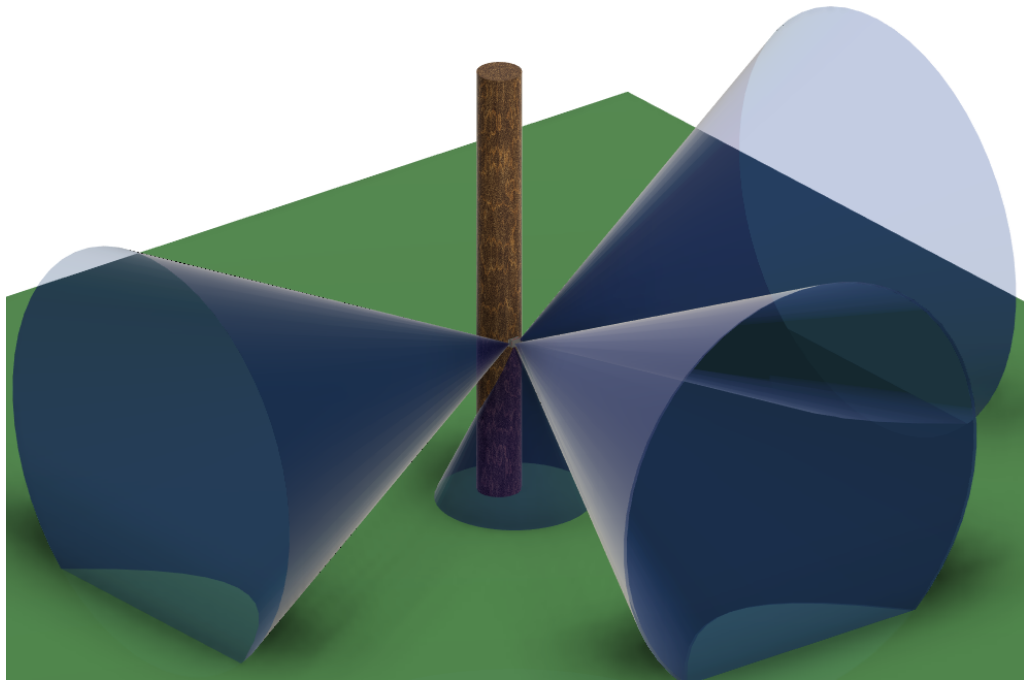


Figure 3.16: The module's fixing simulation to visually analyse the IR sensors' field of view.

With 5FTH 2.0, which protects the **PCB** and its electronic components from environmental conditions, the wireless module is ready for the following phases: firmware configuration and extensive data acquisition testing. In this sense, the following sections are designated.

3.3.4 Firmware description

Properly configuring the firmware, understanding the operational range of the sensors, and determining the data parameters to be sent are essential tasks that ensure the accuracy and reliability of the system as a whole. Therefore, this section describes the entire technical process of develop the module's firmware, ensuring that it not only functions as expected but also optimizes data collection and information transmission.

Firmware configuration involves more than simply programming the microcontroller to work as a data collector. The microcontroller's firmware must have a strategic task flow, the concept is illustrated in Figure 3.17, in order to extend its battery without compromising the reliability of the data. These strategic tasks must consider the functioning of sensors, energy management, data communication, and, above all, the limitations and strengths of the microcontroller. Therefore, it is necessary to understand the range of values captured by each sensor, as it allows precise adjustments to be made to optimize sensitivity and specificity in detecting critical environmental events, such as variations in temperature, humidity, and **IR** spectra as indicators of fire. It is also necessary to respect transmission cycles, through "Best Practices" of using the **LoRaWAN** network [144], therefore, the module can maintain reliability while saving battery (the reasons for using **LoRaWAN** network service are detailed in the next chapter).

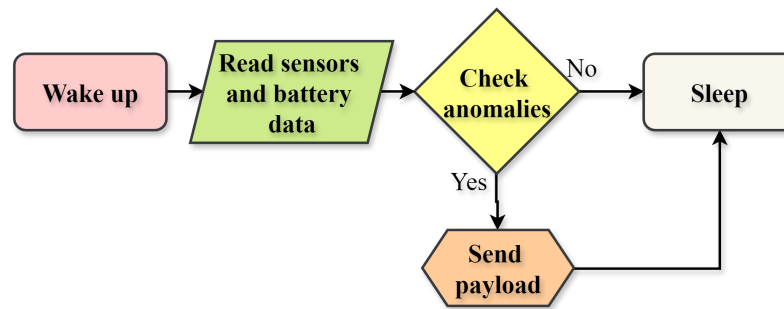


Figure 3.17: An example of flowchart with main tasks that a microcontroller must perform to save battery power while performing data acquisition.

The firmware must be prepared, through filtering or aggregation logic in the module itself, to identify normal and abnormal conditions in the data when collecting and then choose whether or not to send the data. The precision of this process is vital to ensure that the module can perform its functions autonomously and extend its own battery, a valuable ability for applications in remote environments such as forests. Therefore, in Figure 3.18, an image contains the flowchart representing all firmware tasks. This visual representation summarizes the microcontroller's operations from the moment of switching on to operation and auto-shutdown. The description of these tasks is indicated from task ① to ⑦.

The firmware is structured around the fundamental `setup()` and `loop()` functions, respectively ① and ②, typical of programming on ATmega328P microcontrollers.

In the `setup()` function ①, the firmware initially configures the analog A0 and digital D4 pins. Pin A0 is dedicated to measuring the battery level, and therefore, it is necessary to configure an external voltage reference via software using the command `analogReference(EXTERNAL)`. This configuration is crucial to ensure that analog measurements (IR sensors and battery level) remain reliable even as battery voltage levels decrease. With the D4 port, it is possible to control the on-off of the DC-DC converter, which ensures the power supply to all sensors and the transceiver only when required. The process continues with the initialization of the RFM95 transceiver ③, preparing the device with the necessary information for data transmission to the LoRaWAN network. Then, the `setup()` function will be closed after sending the first message ④ with all values as 0. This way, upon receiving the message, it is possible to identify that the module has undergone some reboot (could be due to bugs or battery change). With a message of all-zeros, it is possible to elaborate some procedures during data analysis regarding historical data; for example, a given data analysis might start and stop after a zero message.

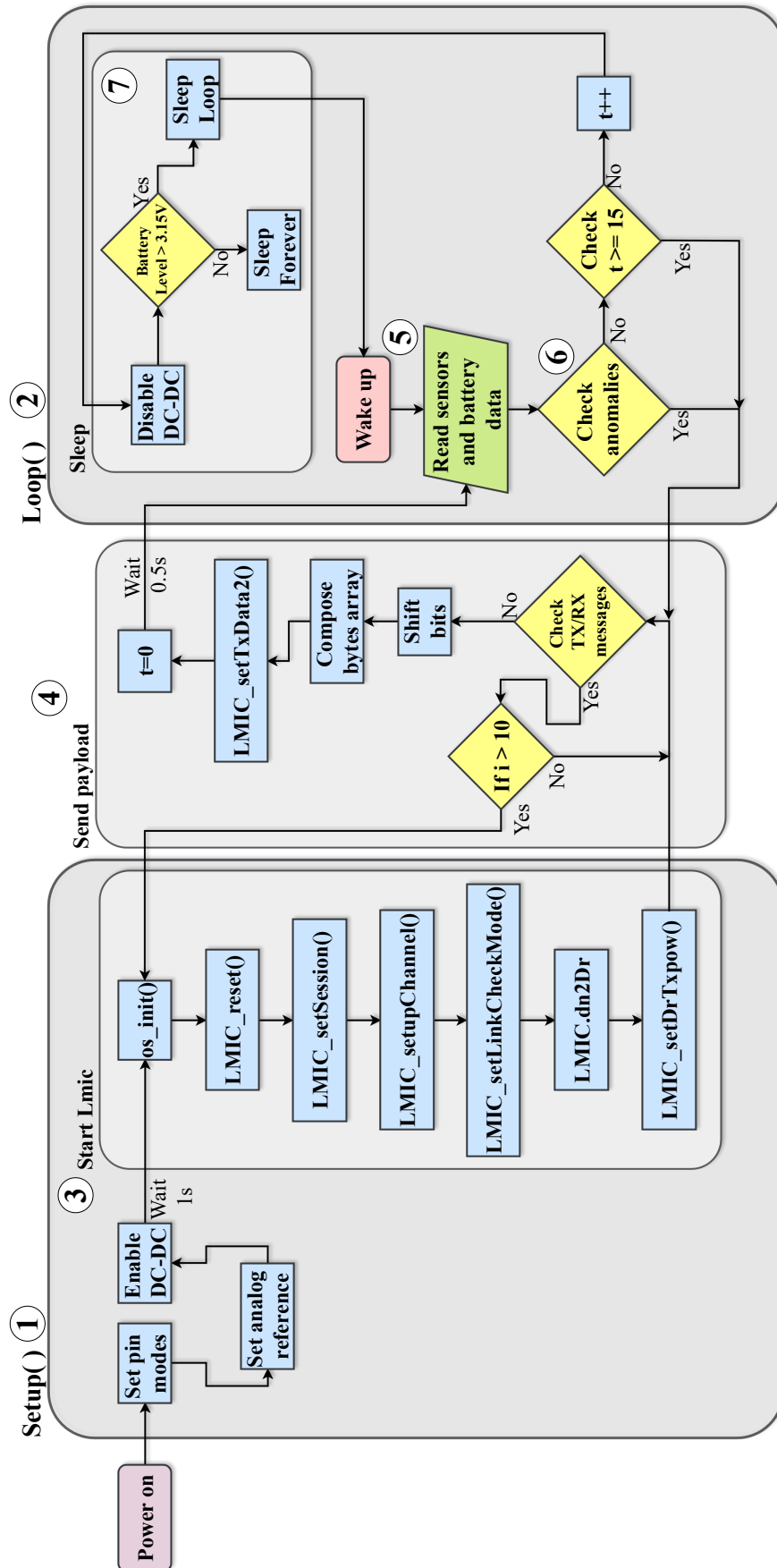


Figure 3.18: The flowchart of all the tasks representing the microcontroller's operations.

Before moving on to the description of the `loop()` function, two processes must be better described as they will be repeated within the `loop()` function. They are: the RFM95 initialization process ③ and the message-sending process ④. LoRa transceiver initialization is performed through a sequence of function calls within the LMIC library. In this sense, each of these processes are listed and detailed below:

- `os_init()`: This function initializes the LMIC operating environment, preparing the software to manage LoRa communication.
- `LMIC_reset()`: This resets all LMIC states to factory defaults, clearing previous settings and any residual data that may affect the new communication session. This reset ensures that earlier operations or residual configurations are not interfered with.
- `LMIC_setSession()`: This function sets static session parameters such as device address (`DevAddr`) and session keys (`NwkSKey`, `AppSKey`), which are crucial for authenticating and encrypting LoRa communication. The correct configuration of these parameters is necessary for secure integration with the LoRaWAN network.
- `LMIC_setupChannel()`: Defines the communication channels that the module will use to transmit data. Correctly configuring channels is important to comply with local regulations and optimize communication with the network. In this work, due to the characteristics of the Gateway used, two groups with eight channels each are defined: the first group contains frequencies of 868.1 MHz, 868.3 MHz, 868.5 MHz, 867.1 MHz, 867.3 MHz, 867.5 MHz, 867.7 MHz, 867.9 MHz, 868.8 MHz; and the second group of channels 865.3 MHz, 865.5 MHz, 865.7 MHz, 865.9 MHz, 866.1 MHz, 866.3 MHz, 866.5 MHz, 866.7 MHz, 865.8 MHz.
- `LMIC_setLinkCheckMode()`: Determines whether the device will actively check network connectivity. Disabling this function can save energy as it reduces the number of control messages sent. Then, it is set to 0.
- `LMIC.dn2Dr`: This parameter defines the data rate for downlinks, adjusting how the device should handle messages received from gateways or network servers. By default, the The Things Network (TTN) service uses `DR_SF9` in its RX2 window, which is why this value is chosen.
- `LMIC_setDrTxpow()`: This function sets the uplinks' data rate and transmission power, balancing range, and power consumption. Proper selection of these parameters is crucial to maintaining transmission effectiveness within desired power consumption limits. The choice of values inserted in this function depends on the distance at which the modules will be fixed. However, the parameters `DR_SF7` and 14 were chosen as data rate and power transmission values, respectively.

The message-sending process ④ begins with checking for any pending messages and ensuring that the transmission channel is free before querying a new message. If the channel is not free, the system will attempt to resolve the congestion through multiple checks and, if necessary, restart the transceiver. In this regard, it will ensure that the actual message will always be sent independently of anything. Once the channel transmission is free of pending issues, the microcontroller processes the sensor data, optimizing the payload through data conditioning techniques before transmission via **LoRaWAN**, using the Shift Bits and Compose Array functions (the complete description of these techniques are published in [145]). Finally, `LMIC_setTXData2()` is called to send the data, and next, a variable t is dedicated to count the loop iterations. The `setup()` function ends with a brief waiting period to ensure synchronization before calling `loop()`.

At this stage, specifically in the transition between the end of the `setup()` function and the beginning of the `loop()` function, it is expected that each module sends an initial message with zero values for all variables. This practice has a dual purpose: first, it serves as a clear indication that the module has been correctly initialized and is ready to begin regular operation; second, these zero values, being only possible to obtain during the first data sending, can facilitate the identification of the exact moment of initialization in the data records. Then, several considerations can be made during data analysis; for example, this could be used as a trigger.

The `loop()` function ② performs the continuous operation cycle. It is executed every 60s due to the maximum duty cycle allowed by the **TTN** with the volume of data that each module generates. The first action of the `loop()` function is reading the sensors and battery data ⑤, which provides the basis for all subsequent decisions in the module.

Immediately after reading, the data is analysed by **DROWSIER**'s digital filter ⑥ to identify any anomalies. This analysis uses a specific method, as described in [30], which, through mathematical methods, can identify abnormal deviations in sensors' data. If an anomaly is detected, the module is programmed to immediately transmit the latest data obtained through the RFM95 transceiver (Send payload process ④). Otherwise, the microcontroller will check the loop iterations.

When the current iteration corresponds to 15, it indicates that the module has not sent data for 15 min. Therefore, the module must send the message of the last values read. This task guarantees constant data transmission even in the absence of anomalies. This iteration value is chosen based on considerations of energy efficiency and communication frequency, as explored in [30]. If the iteration is not 15, the microcontroller continues to increase the number of iterations t , preparing for the next loop.

After incrementing t or after data transmission, the module goes into standby mode ⑦. This standby mode, or sleep mode, is essential during each module's operation cycle. Sleep mode allows the module to save energy by significantly reducing its activity and, consequently, its power consumption until the next cycle of operation.

Sleep mode was implemented using the Low Power library [146] for two distinct cases: `sleep loop` and `sleep forever`. Before carrying out these two cases, the sleep mode starts by turning off the DC-DC converter. It disables all sensors and the RFM95 transmitter, as they are all connected

to the converter output circuit. After the converter is turned off, the microcontroller checks the value of the last battery level reading.

If the battery voltage level is below 3.15 V, it does not have enough power to maintain module operation. This value was determined based on the converter datasheet and experimentally verified in the laboratory. The verification was carried out using an adjustable power supply, which was used to simulate the battery discharge from 4.5 V to 3.0 V with decrements of 0.01 V. This validation evidenced that the converter did not work correctly with voltages lower than 3.1 V. Therefore, to ensure that the module can notify its inability to continue operating, a voltage of 3.15 V was chosen as the threshold. Hence, the microcontroller activates the sleep forever function after sending the hibernation message.

In sleep forever mode, the module sends a message using a control variable named *AUX*, which, in this case, receives the value 15, indicating a critical battery level. Then, the module goes into hibernation, or sleep forever, using the `powerDown(SLEEP_FOREVER, ADC_OFF, BOD_OFF)` command from the Low Power library. On the other hand, if the battery level is above 3.15 V, the module still has enough capacity to continue operating, and the microcontroller starts the sleep loop process. This task uses the `powerDown(SLEEP_8S, ADC_OFF, BOD_OFF)` command six consecutive times. This repetition was determined in the laboratory and ensures that the `loop()` function is executed approximately every 60s. After six repetitions of this command, sleep mode is completed, and therefore, the next step is to execute the wake up function.

As the name suggests, the wake up function reactivates the DC-DC converter, energizing all sensors connected to its output, and reactivating the RFM95 transmitter. After the wake up function is completed, the microcontroller returns to read the sensors and performing all subsequent tasks.

3.3.5 Message Payload to LoRaWAN Gateway

Considering the possibility of a massive number of the modules used in the **WSN**, it is necessary to optimize the size of the data package sent, thus guaranteeing the minimum necessary occupation of bytes to reduce the chances of possible transmission collisions and consequent data losses. Looking for the data produced by each module, some sensors have a 8 or 10-bit resolution while **LoRaWAN** owns 16 bits, which means, in some cases, multiple slots of transmission remaining unused bits. In this way, a paper was published that addresses a communication optimization for **WSN** resorting to encoding and decoding procedures [145]. In this sense, for convenience, only the main information about the data flowing from the nodes to the **LoRaWAN** network is detailed.

This approach intends to sort the bits using as little value as possible in the bytes protocol used to transmit the data by **LoRaWAN**. The main idea is to distribute each bit from each sensor during the encoding process using the minimum number of Bytes as possible. The corresponding number of bits for each variable sent is listed below.

- **IR sensors:** as mentioned before, each node has five **IR** sensors. These sensors make 10 bits reading, which means they generate values between 0 to 1023. To send just one value

from one of them is demanded to use 2-Bytes (16 bits), consequently, to send all of them is required a total of 10-Bytes.

- **Relative Humidity sensor:** this sensor produces a reading between 0 to 99, but 2-Bytes are required to transmit the message.
- **Temperature sensor:** gives a reading between 0°C to 50°C, which results in the use of 2-Bytes to transmit.
- **Battery level:** the battery level is converted from a 10 bits **ADC** and a total of 2-Bytes is required to transmit.

As it is possible to notice, if all the data is sent without any manipulation, some bits may be sent free of information. In order to optimize the *Not Used* bits for each variable, a shift bit operation can be helpful. By maintaining the **IR** sensors as a base, it is possible to insert the other values (humidity, temperature, and battery level) by shifting each bit into the non-used bit's **IR** sensor variable. After this process, there are 4 bits that remain unused. They can be used as auxiliary bits during the transmission for future applications (named as AUX variable). Figure 3.19 presents the encoding, transmission and decoding procedures. As it can be seen, the packet to be sent keeps the size of produced by the **IR** sensors $S[i]$ (where $i \in \mathbf{R} | 0 \leq i \leq 4$):

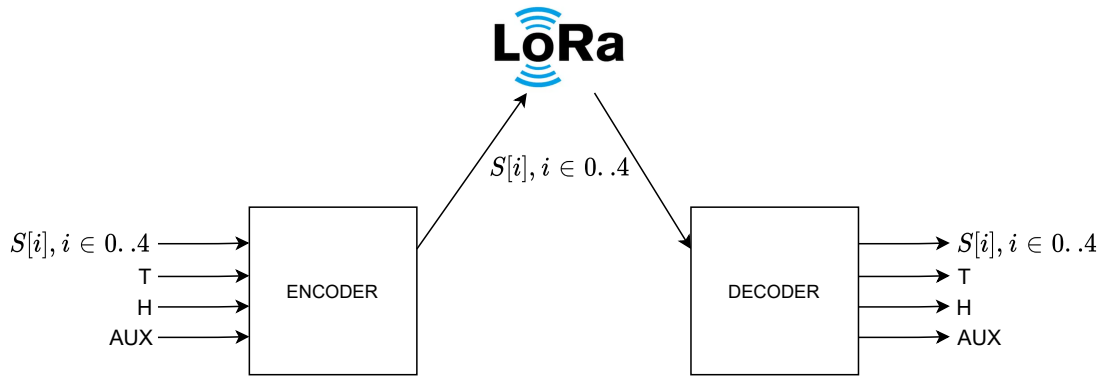


Figure 3.19: Encoding and decoding procedures. Size of $S[i]$ is 16 bits whereas T (temperature) and H (humidity) are 8 bits. AUX is an auxiliary variable that was created with the 4 bits that remain free [145].

By this way, a reduction of transmission from 8 words (16-Bytes) to 5 words (10-Bytes) is obtained. More information about this strategy can be found in [145].

3.4 Module Sensors' Results

After all the development of the wireless module, whose main objective is to monitor forests to prevent wildfires, some tests are relevant to verify its performance in environments without fire and with fire. By performing these tests, it is possible to verify and validate the modules' robustness,

effectiveness, and reliability as they form a **WSN** in multiple operational scenarios. For this purpose, the tests in this section were performed in scenarios without fires, because the objective is to evaluate the 5FTH's capability during its implementation in outdoor environments.

Non-fire testing aims to examine battery durability and effective implementation of power-saving modes. Using a **WSN** with eight modules configured to send payloads every minute, different sleep mode configurations are explored to identify which ones maximize energy efficiency without compromising operational performance. In addition to battery durability, an approach also includes testing data transmission reliability. As each transmitted message is stored in time-series format, as shown in Table 3.5, verifying whether the 60s intervals between messages are maintained consistently is possible. Through this analysis, it is possible to check the occupancy of the **LoRaWAN** network while they work in 60s cycles and, consequently, develop strategies for better data traffic management and communication stability.

Table 3.5: Dataset's structure generated by a given sensor module stored on the server.

Time [HH:mm]	Battery [V]	Humidity [%]	Temperature [°C]	IR1	IR2	IR3	IR4	IR5
Day 1 00:01	3.82	75	10	42	45	42	55	40
Day 1 00:02	3.82	75	10	44	41	44	56	42
...	...	...	...	...	...	...	...	...
Day 5 12:02	3.63	56	19	46	45	46	64	46
Day 5 12:03	3.63	51	19	40	38	39	47	39
...	...	...	...	...	...	...	...	...
Day 20 23:58	3.48	70	17	1023	1023	1023	1023	1023
Day 20 23:59	3.48	71	17	1023	1023	1023	1022	1023

All tests described in this section will be conducted exclusively with the 5FTH 2.0 model. This version was chosen due to its significant improvements over previous versions, especially about sensor integration and reliability of the collected data. During testing, the modules of this version will be named as *n-5FTH*, where *n* corresponds to the specific **Identifier Device (ID)** of each module.

Figure 3.20 displays two examples of the module's installation in an outdoor environment. It is possible to observe that during the fixing of the modules, there was always a concern to place them in strategic positions according to the terrain. In other words, they were fixed to obtain direct exposure to the surrounding environment. As Figure 3.20a and Figure 3.20b show, both modules are not covered by any vegetation, and the view is not impaired by any obstacles close to their estimated sensitivity radius.



(a) Module fixed at IPB campus.

(b) Module fixed at Serra da Nogueira.

Figure 3.20: Example of fixation of the module outdoors.

The details and analysis of the tests are presented in the following subsections. Each subsection explores different aspects and results of the tests to evaluate the 5FTH.

3.4.1 Testing WSN's battery consumption

This test aims to verify the energy consumption of a **WSN** consisting of eight modules, set up to transmit data every 60s without applying adjustments to sleep mode or changing the interval between transmissions. The eight modules were fixed outdoors on the **IPB** campus. The batteries were inserted sequentially, following the module numbering, to ensure an organized startup. All modules were configured to operate on the same **LoRaWAN** gateway, using SF7 and the frequencies corresponding to the first group of channels, as specified in Subsection 3.3.4.

The objective is to verify the maximum period that each module can send data in external environments, and then, a daily average filter is applied to the collected data. The daily averages, represented in the graph shown in Figure 3.21, facilitate the visualization and analysis of the battery discharge, as the filter allows a more explicit and less noisy representation of battery charge variations over time.

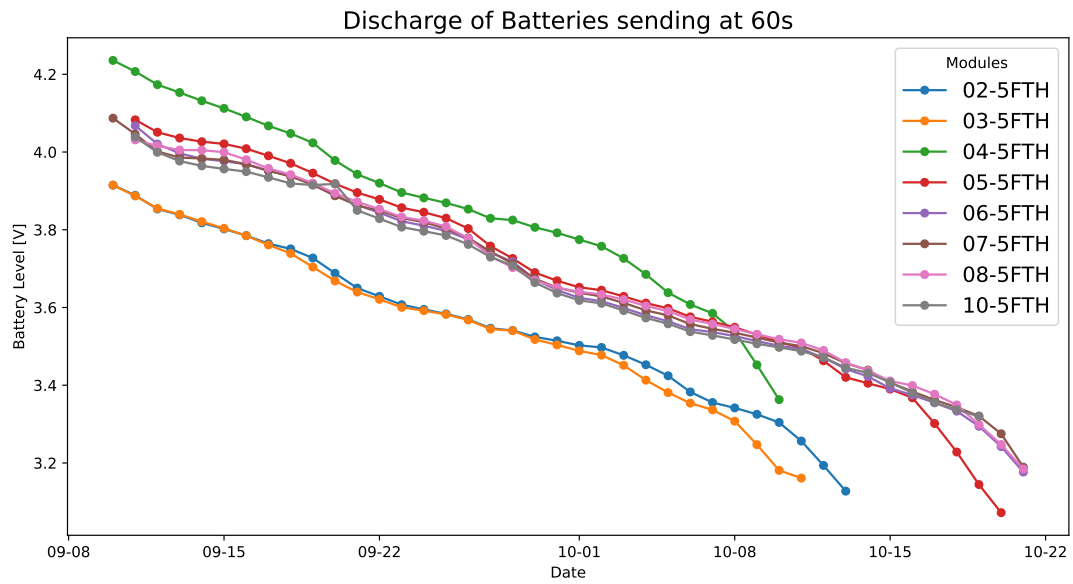


Figure 3.21: The battery discharge graph over 44 days of a WSN with eight modules, a transmission rate of 60s, and no sleep mode.

The WSN was activated on September 8th, and the last received data was on October 22nd, totaling 44 days. It was observed that some modules ended their transmission before others. The graph in Figure 3.22 shows the total amount of data sent by each module and the days on which each one remained online. The days of operation for each module were calculated by the difference between the date of the last and first recorded data. It is essential to highlight that this test did not evaluate transmission continuity, that is, whether each module sent data strictly every 60s.

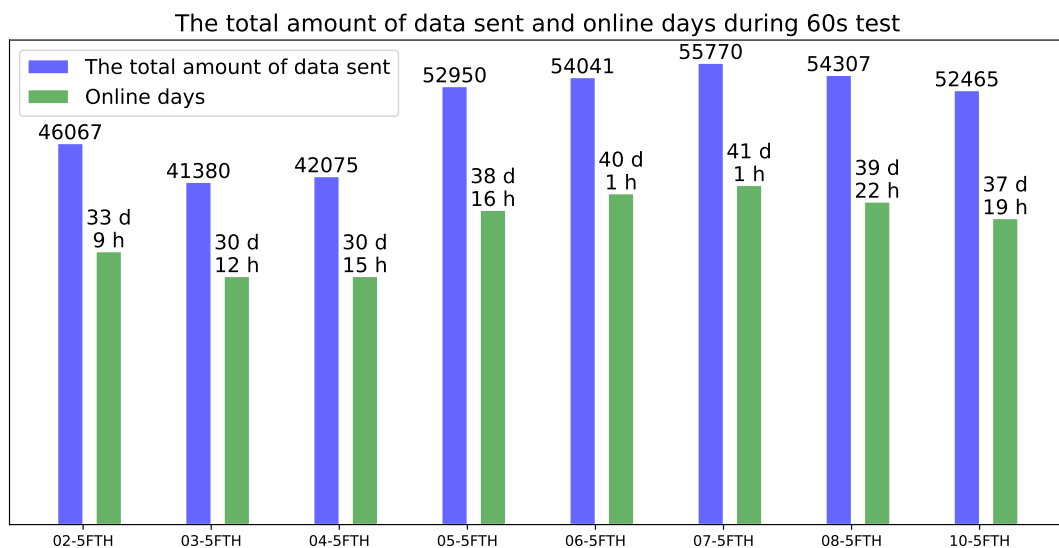


Figure 3.22: The total amount of data sent and online days during the test of a WSN with eight modules, a transmission rate of 60s, and no sleep mode.

The analysis of the collected data and the observation of battery discharge over the tested period

indicate the energy efficiency of the modules and the suitability of the configurations used for the operational reality in the field. However, as previously evidenced, each module can be configured to enter into standby mode between each data send. In other words, the operation would be repetitive and involve reading the data, sending it, and entering a sleep mode until the 60s were complete.

In this sense, the second test focuses on evaluating the impact of using a sleep mode after each data transmission, maintaining a transmission rate of 60 s. The scenario of this test was identical to the first, using the same WSN with the same eight modules positioned outdoors on the IPB campus. It also used the same activation sequence and transmission configuration through the same gateway on SF7 and frequencies from the first group of channels.

The objective of this test was to evaluate how implementing sleep mode immediately after each transmission could extend the battery life of the modules compared to the previous test, where the modules remained active at all times. As in the last test, a daily average filter was applied to facilitate the analysis of battery decay, shown in the graphs in Figure 3.23.

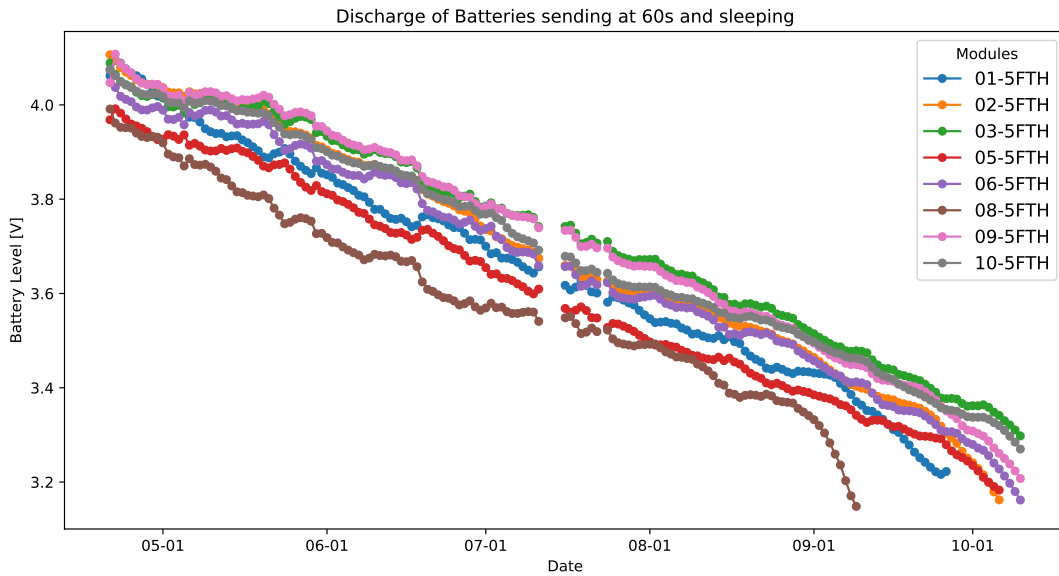


Figure 3.23: The battery discharge graph over 172 days of a WSN with eight modules, a transmission rate of 60 seconds, and sleep mode activated.

The test period is from April 21st to October 10th, totaling 172 days. During this period, there were two interruptions in the gateway service due to technical maintenance on the IPB infrastructure, specifically from July 11th to 16th and from July 23rd to 24th. These interruptions, seen in the two gaps in the graph in Figure 3.23, however, did not affect the battery consumption of the modules, since the data sending procedures were maintained according to the **Activation By Personalization (ABP)** configuration. Which does not depend on the connection network for the operation of the modules. Therefore, even though data was not recorded, all modules consumed battery power through data transmission throughout the entire period of this test.

Similar to what was observed in the previous test, during the second test, it was also observed

that some modules ended their transmission before others. Thus, in the graph presented in Figure 3.24, it is possible to visualize the comparison between the modules through the total amount of data sent during the test period and the number of days that each module was online. Also, it is possible to determine that sleep mode, when activated, can significantly extend the operability of the modules, reducing the frequency of maintenance and battery recharges.

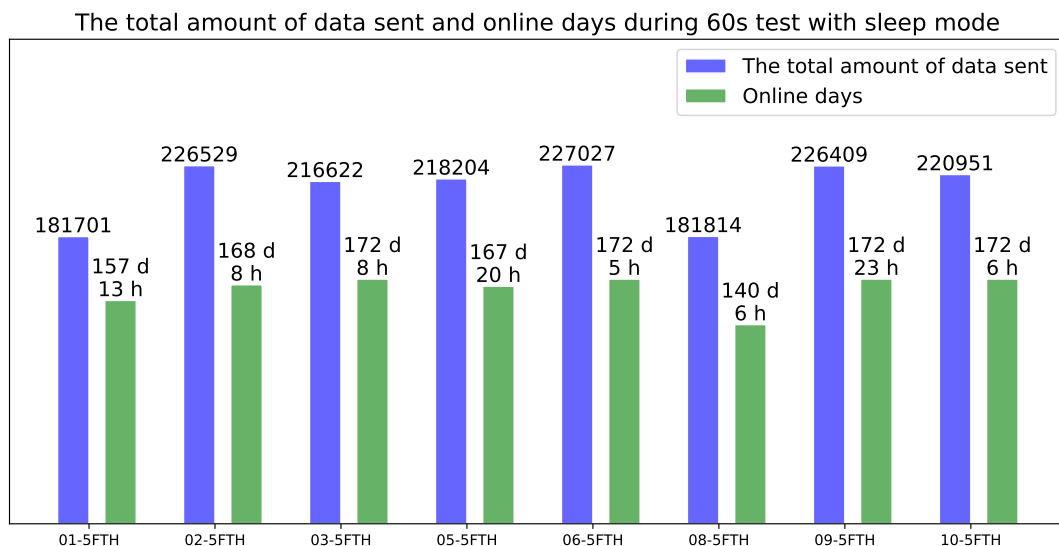


Figure 3.24: The total amount of data sent and online days during the test of a WSN with eight modules, a transmission rate of 60s, and sleep mode activated.

After activating sleep mode, the data transmission process became the main energy consumer in the modules. During each cycle of reading, sending data, and sleep mode, an average consumption of approximately $672.30 \mu\text{A}$ was observed. Notably, the peak current consumption, reaching up to 132.21 mA , occurs during the transmission phase, highlighting the significant energy demand of this process.

Given this finding, a strategy was defined to reduce energy consumption through data pre-analysis before sending it. It is proposed that the modules transmit data under two main conditions: the locally detected abrupt variation in the data, which may indicate a critical condition such as a fire, or after an established minimum period without transmissions, thus ensuring continuity of data collection.

This approach inspired the development of an innovative digital filter named **DROWSIER**, which has the ability to reduce module energy consumption by up to 53%. The filter was created after analysing the data behavior obtained through controlled and supervised fire tests (all fire tests are detailed in the Chapter 5). The development of the **DROWSIER** filter, from the identification of variations to the definition of the applied methods, is discussed in [30].

However, the mentioned publication does not analyse the discharge of the modules' batteries from the beginning to the end of their operational lifetime. Therefore, Figure 3.25 illustrates the battery discharge of a given WSN, indicating the real **DROWSIER**'s filter performance on the battery life of outdoor modules.

The third battery test was conducted in *Serra da Nogueira*, a Portuguese national reserve, where the WSN has twenty modules for forest monitoring. It is important to highlight that these devices were installed in the forest with the authorization of all entities involved, respecting the relevant legislation and environmental guidelines.

For this test, the entire WSN was configured to operate on SF7. Regarding the allocation of frequency channels, the WSN was divided into two distinct groups: ten modules operated in the first group of channels, while the other ten used the second group of frequencies. Thus, in addition to avoiding some possible collisions during data transmission, it is possible to evaluate the performance of modules in different frequency bands under the same environmental conditions.

All twenty modules were configured to operate with the DROWSIER filter, which adjusts data transmission based on analysis of significant events. Each module was programmed to send data every 15 min. Additionally, every 60s, each module wakes up to check the collected data. If the DROWSIER filter identifies an abrupt variation in the data, which indicates a critical environmental condition, the module immediately sends this information. Otherwise, the module returns to sleep mode, and transmission occurs only after the 15 min interval, unless a new alert condition is detected.

The test began on September 5th and lasted until July 26th, totaling 324 days of continuous operation. The results of this monitoring period are presented in Figure 3.25, which shows the average daily discharge of the WSN batteries.

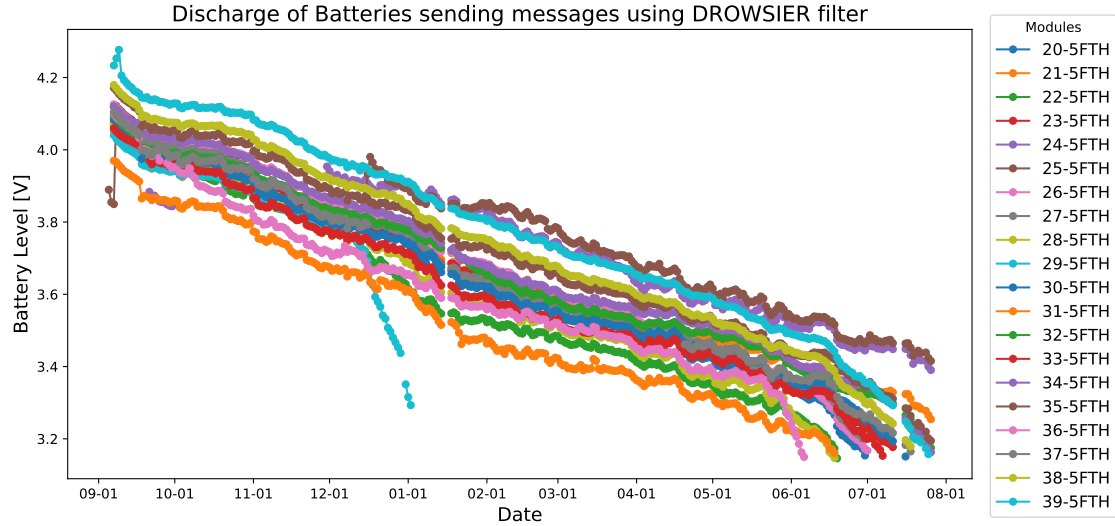


Figure 3.25: The battery discharge graph over 324 days of a WSN with twenty modules using the DROWSIER filter.

Figure 3.26 shows the total amount of data transmitted by each module and the days each module remained online. This graph shows that the number of days online increased significantly after implementing the DROWSIER filter. On the other hand, the volume of data sent by the modules was also reduced. In other words, there was an exchange of data sent for days online. By

reducing the frequency of unnecessary transmissions, the modules are able to save energy, thus extending their operating time without recharging.

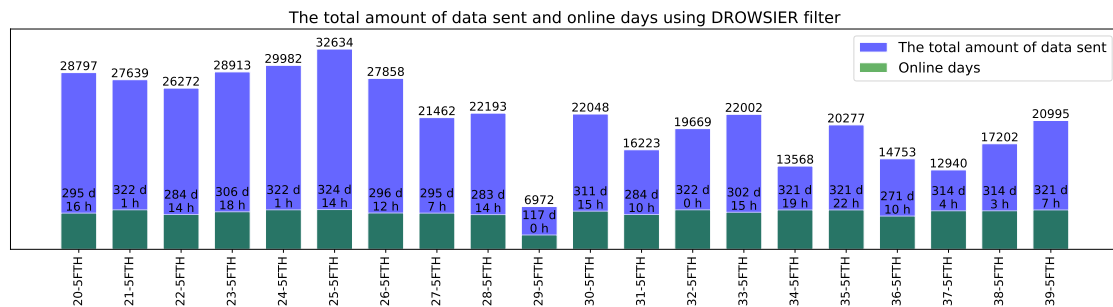


Figure 3.26: The total amount of data sent and online days during the DROWSIER's test in a WSN with twenty modules.

During this test, it was also possible to observe that some modules remained active for longer than others, and transmitted more data, which may indicate greater adaptation of these devices to the specific conditions of their installation environment or greater efficiency in energy management.

3.4.2 Testing WSN's communication

The test described in this subsection was designed to verify the regularity of data transmission cycles of the developed modules when configured with the **DROWSIER** filter. In this sense, this communication test will not focus on signal quality. This experiment focuses on verifying the constancy of transmission intervals, ensuring that the modules transmit data at least every 15 min, as programmed. Therefore, the main objective of this test is to ensure that there are no significant interruptions in the transmission flow that could compromise the integrity of the collected data. Therefore, this test intends to confirm the regularity of the modules in transmitting data.

A module was fixed close to the gateway to ensure that the data flow was not affected by distances or physical obstacles, as shown in Figure 3.27. This specific module was equipped with a 5W solar panel and a voltage regulator to increase autonomy and ensure continuous operation. This is because a conventional power supply is not possible at the installation location, the **School of Technology and Management of Bragança (ESTiG)-IPB's** rooftop.



Figure 3.27: Fixation of the module outdoors close to the Gateway.

The use of the solar panel, although previously considered not essential for the operation of the modules, in this specific case, aims to maximize the module's lifetime under test. Furthermore, as shown in Figure 3.27b, this module was constantly exposed to sunlight, which is different from the cases in which the modules is fixed to tree trunks and, consequently, without intense sunlight. It is crucial to highlight that the application of these extra components is exclusive to this module, and therefore, the implementation of these electronic components will not be discussed.

The transmission regularity test was conducted over 365 days, starting on March 1st and ending on February 29th. The graph in Figure 3.28 illustrates the data collected during this period. This data was generated by the time difference between actual message and the previous message. According to the **DROWSIER** filter, these time intervals are expected not to exceed 15 min. Differences of less than 15 min mean that **DROWSIER** identified abrupt variations in the data, resulting in additional transmissions.

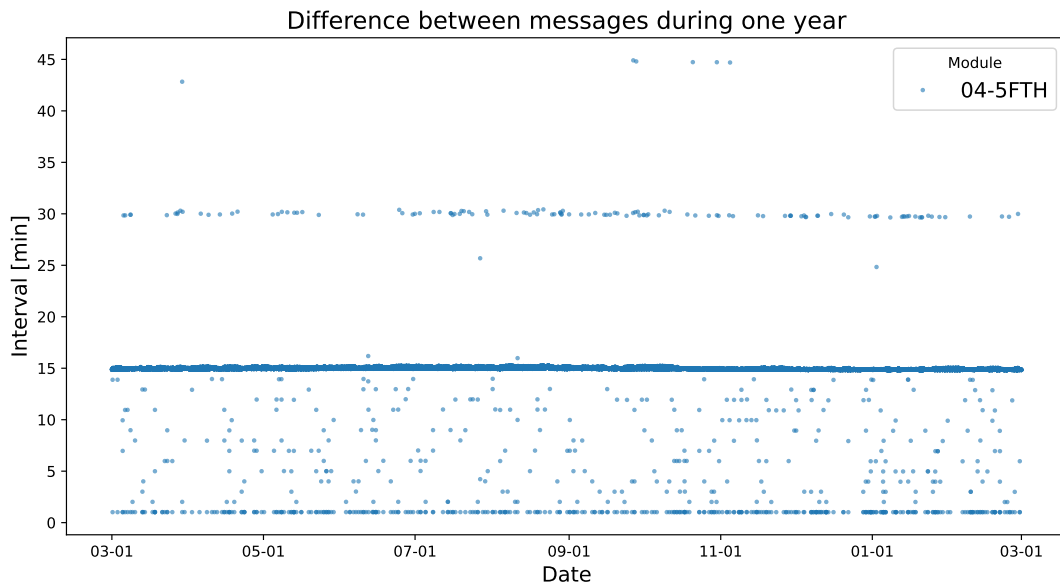


Figure 3.28: Time interval over 365 days of a module using the DROWSIER filter.

The graph in Figure 3.28 indicates that most values are concentrated in intervals of 15 min or less, with some values between 15 and 30 min and a few points identified above 45 min. Still, in this graph, it is possible to notice that the variation observed in the line formed by the dots suggests that the messages are sent with differences of milliseconds. This is because the points do not create a completely straight line. This floating can be attributed to the airtime required to send messages, which may change due to environmental conditions. These fluctuations are evidenced in Table 3.6.

Table 3.6: Example of the dataset stored with floating millisecond differences in time intervals.

Time [HH:mm]	Diff. [min]
Day 01 00:13	<i>NaN</i>
Day 01 00:28	14.884883
Day 01 00:43	14.885650
Day 01 00:58	14.885783
Day 01 01:13	14.883400

During the test period, at least 35136 transmissions were expected to be recorded, which would be the number of 15 min intervals over a year (including the hours of the last day). And the total of message stored was 35376. In Figure 3.29, four histograms are presented, all normalized concerning the bin density to facilitate interpretation. The first histogram shows the general distribution of time intervals between transmissions using 45 bins. As the data is concentrated in the 15 min range, three more histograms were generated to better detail this distribution. Therefore, the second histogram covers the range from 0 to 14.5 min, with 15 bins. The third histogram focuses on the critical range between 14.5 min and 15.5 min, using 156 bins. Finally, the fourth histogram presents intervals greater than 15.5 min minutes, with bins set at 30.

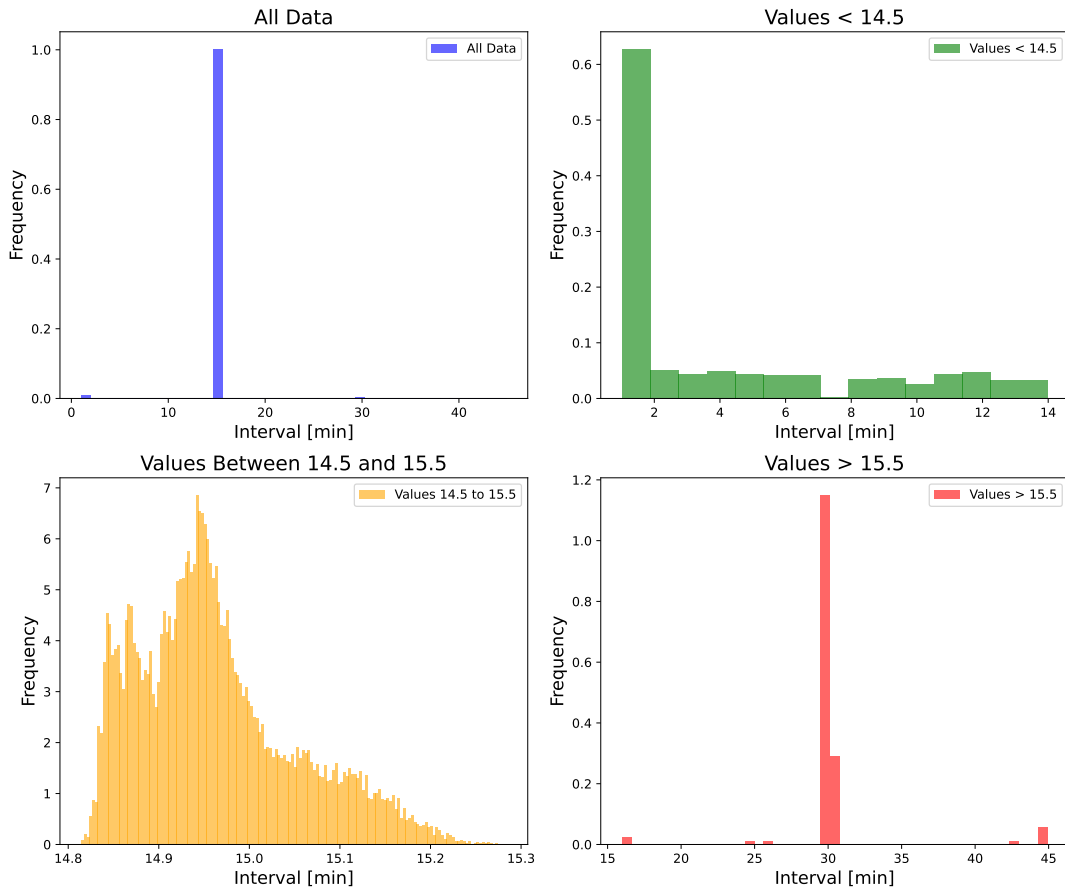


Figure 3.29: The histograms with the time intervals' distribution from different perspectives.

Therefore, to determine how many transmissions exceeded 15 min, a filter was set at 15.5 min. This adjustment takes into account that airtime rarely exceeds half a minute and also avoids the accumulation of additional **DROWSIER** messages if they are sent in the next minute. Based on this analysis, 134 values were obtained that exceeded the 15.5 min interval, representing 0.38% of the total messages.

3.5 Discussion of Nodes' Results

After developing the wireless module to monitor forests and identify fires, tests without fires were performed to verify the 5FTH's performance in terms of battery life and message delivery rate through **LoRaWAN** communication.

The non-fire tests aimed to examine battery durability and the effective implementation of power-saving modes. Using a **WSN** with eight modules configured to send data every minute, different sleep mode configurations were explored to identify which maximize energy efficiency without compromising operational performance. In addition to battery durability, reliability in data transmission was also evaluated. As shown in Table 3.5, each transmitted message is stored in time series format, which allows the verification that the 60 s intervals between messages are maintained

consistently. This analysis allows for checking the occupancy of the **LoRaWAN** network while operating in 60s cycles and, consequently, developing strategies for better data traffic management and communication stability.

Battery durability data indicates a considerable performance of the modules even without any software improvements, with some operating for up to 44 days without sleep mode (Figure 3.21) and 172 days with sleep mode activated (Figure 3.23). These results show that sleep mode can significantly extend the operability of modules by reducing the frequency of maintenance and battery recharges. Furthermore, implementing the **DROWSIER** digital filter proved to be an effective strategy for reducing energy consumption by up to 53% by adapting the data transmission frequency based on the analysis of detected critical events. The test carried out in Serra da Nogueira, which lasted 324 days, confirmed the effectiveness of the filter, showing a significant increase in module operating time and a reduction in the volume of data transmitted (Figures 3.25 and 3.26).

During the fireless tests, the communication test was designed to verify the regularity of the data transmission cycles of a module configured with the **DROWSIER** filter and under good **LoRaWAN** coverage (without possible interference from obstacles). This test confirmed that the module transmits data at intervals of up to 15 min, as programmed, exceeding 15 min in a few transmissions, guaranteeing the integrity of the collected data (Figure 3.28). Few interruptions were identified over 365 days of operation, representing only 0.38% of total messages (Figure 3.29).

3.6 Final Considerations

Test results indicate that the developed modules are effective in detecting forest fires and have considerable durability, especially when configured with energy-saving modes. **WSN** proved to be robust and reliable both in continuous operating conditions and in simulated fire scenarios. Implementing the **DROWSIER** filter has proven to be an efficient strategy to extend battery life and improve module energy efficiency.

Additionally, testing evidences that fire identification may appear relatively simple when evaluating data from a single module at night. However, the situation becomes significantly more complex when considering a network of multiple modules operating during the day. Variations in environmental conditions, such as sunlight, can introduce noise into the data, making it challenging to identify fire events.

In this context, analysing data collected from an extensive network of modules under different conditions requires advanced data processing techniques. Applying **ML** methods becomes essential to assist in decision-making, allowing the accurate and rapid identification of forest fires. These techniques can filter noise and identify complex patterns in data, improving detection capabilities and reducing the possibility of false positives or negatives.

Before implementing **ML** techniques, designing a robust system architecture is critical. This includes creating an infrastructure that supports data collection, storage, and processing efficiently and securely. A well-planned architecture will allow the integration of a dashboard for real-time visualization, facilitating continuous monitoring and analysis of collected data. Furthermore, the

architecture must guarantee the scalability and resilience of the system, ensuring that it can handle large volumes of data and different operating conditions. In this way, it will be possible to maximize the potential of **WSN** in preventing forest fires, using **ML** techniques to transform raw data into actionable information.

Chapter 4

An Integrated Approach to Data Acquisition, Storage and Visualization Systems

Amid technological advancements and the increasing demand for sustainable environmental monitoring, integrating low-cost systems in natural environments, such as forests, emerges as a vitally important field of research and development. This chapter is dedicated to exploring the development of an innovative system based on **IoT** technologies, which aims to provide an effective and affordable solution for collecting, storing, and visualizing environmental data. This study's relevance is based on its contribution to more efficient monitoring and its ability to provide a valuable system for detecting wildfires early. In this sense, and through the implementation of low-cost sensors and advanced communication technologies, a method capable of operating autonomously and collecting essential data is proposed. This chapter is structured as follows:

- Section 4.1 – provides a general contextualization of autonomous systems and the justification for their use in monitoring forest environments. The final part of this section outlines the research objectives, focusing on identifying efficient **IoT** systems, searching for open-source and flexible solutions, and integrating new technological trends.
- Section 4.2 – presents the bibliographic review and related works, providing the theoretical basis necessary for understanding and developing the proposed system.
- Section 4.3 – details the creation of the system, including the definition of the system architecture, the selection of tools for data acquisition, the implementation of data storage, and the development of process management tools. The integration of dashboard visualization tools, essential for analysing and interpreting collected data, is also explored.
- Section 4.4 – describes the system's practical implementation, including testing, validation, data collection, and analysis, as well as a final description of the system and the minimum hardware requirements for its operation.

- Section 4.5 – discusses the scalability, flexibility, and broader applications of the proposed system, concluding with the final considerations about the proposed system.

4.1 General Contextualization of Systems

In an increasingly connected world and facing growing environmental challenges, such as early detection of forest ignitions, the importance of integrated and autonomous systems for monitoring and managing natural ecosystems has become more necessary than ever [147]. Often, isolated systems or systems that operate independently face significant limitations, such as the inability to process data globally, difficulties in adapting to constantly changing environmental scenarios, and challenges in effective communication between different technologies. These limitations can lead to gaps in monitoring and critical delays in responding to environmental events such as wildfires, highlighting the necessity for more integrated and collaborative systems [148, 149, 150].

When well orchestrated, these integrated systems represent the confluence of multiple technologies and approaches, where each component performs a distinct task in creating a global and effective solution [151]. Combining different systems with distinct tools and purposes can overcome the limitations of isolated systems, providing a more comprehensive view and allowing a faster and more efficient response to critical situations. From this perspective, the collaboration between systems can result in a united system, powerful in combating environmental challenges (such as wildfire prevention and early detection) by combining real-time data collection, advanced analytics, and autonomous capabilities.

Choosing a suitable wireless communication technology is not just about achieving maximum range or minimizing power consumption [152, 153], but also about ensuring that the data collected has safe and efficient airtime, overcoming natural obstacles and environmental variability. This strategic choice of technology is fundamental to maintaining the integrity and continuity of the data flow, certifying that crucial information is transmitted reliably for analysis and decision-making [154].

In the sphere of data management, the system architecture must be capable of handling the large volumes of data generated continuously. The ideal solution requires a platform that can not only store this data efficiently but also offer advanced capabilities for its processing and analysis. The dynamic and temporal nature of the data collected requires a specialized time series approach, guaranteeing accurate and timely analysis. Furthermore, data visualization is a critical component, demanding a platform capable of transforming massive raw data into understandable and actionable information [155]. The ability to create intuitive and interactive visualizations is essential to allow users to analyse trends, detect anomalies, and make informed decisions when required. Thus, the effectiveness of data visualization is to transform complex data into simplified information without losing quality, in other words, acting as a translator of the numerical data.

The scalability of an autonomous monitoring system is also an important aspect to be considered during its construction [156]. Therefore, the system for monitoring forest ignitions must have the ability to integrate several tools and technologies, each with its specific purpose, into a single

scalable architecture. The integration is essential for the system to operate cohesively, adapt, and scale according to the project's evolving needs.

This scalability is achieved by implementing a modular architecture [156], where each component can be managed and scaled independently. Adopting a module-based approach ensures that the system remains resilient and flexible, capable of incorporating new technologies and adjusting to changes in the operating environment or project requirements. This method also facilitates system maintenance and updating, as each module can be improved or replaced without affecting overall operation.

In this type of system architecture, it is essential to use technologies that allow virtualization and containment of operational environments. Such technologies provide an efficient means to encapsulate different tasks and services [157], allowing them to operate isolated and securely while sharing resources optimally. Additionally, choosing open-source solutions maintains the project's costs low, promoting an innovation and continuous improvement ecosystem powered by a global community of developers and experts [158, 159].

Given the advantages of such technologies, integrating **IoT** technologies into environmental monitoring systems presents a significant opportunity for several reasons. For example, with **IoT** technologies in environmental monitoring, it is possible to obtain real-time data collection, advanced data analysis, cost reduction, informed and fast decision-making, automation, and control, among others. Therefore, the general objective of this research is to develop an innovative system with multiple integrated tools, with the purpose of creating an autonomous (or semi-autonomous) real-time monitoring system - especially focused on forest areas. In this sense, the following topics describe some research sub-objectives that, taken together, will complete this study's main objective.

Identification of Effective IoT Systems

As **IoT** is a rising technology that is in continuous transformation, the first step for developing an integrated **IoT** system is identifying existing systems that have demonstrated effectiveness and those that have faced challenges in implementation. Continuous literature review and critical analysis of available technologies are fundamental to developing more efficient solutions. The first objective is to identify **IoT** systems that have proven effective in different environmental and geographic conditions and have successfully addressed limitations such as data processing capabilities and communication reliability.

Open Source Technology Integration and System Architecture Development

Applying flexible, open-source solutions in environmental monitoring systems ensures affordability, continuous innovation, and global collaboration. Using the latest software trends, such as container platforms for virtualization, allows independent and secure management of system components. Then, a sub-objectives is to identify relevant open-source technologies and software trends to create a modular and scalable system architecture. Additionally, the sub-objectives must to develop and

demonstrate a system architecture that attend the requirement of real-time data collection, advanced processing, visualization, and ML analysis. Each component, including data acquisition, storage, and visualization tools, must be detailed. To ensure scalability, a Docker version of the system must be developed, enabling broader access and implementation by other researchers and developers.

4.2 Bibliographic Review and Related Works: IoT for Monitoring Systems

Building an efficient, robust, and scalable monitoring system for early detection of forest fires involves collecting, processing, and analysing data in real-time, ensuring rapid response to critical events. Therefore, this section presents a review of the literature and related works to technologies referring to all the processes that this system must have, that is, the components and applications of the IoT to environmental monitoring. This review provides the necessary theoretical basis for the proposed system's development, covering everything from fundamental concepts to specific data collection, storage, and visualization technologies.

IoT has revolutionized data collection, processing, and application in several areas, including environmental monitoring. Using IoT in natural environments (i.e., as forests) offers significant advantages, such as monitoring in real-time, quickly detecting environmental changes, and providing support decisions. To build an effective environmental monitoring system based on IoT, it is essential to understand the fundamentals of this technology, as well as the tools and methodologies that have proven to be most effective in related works. However, IoT has a vast volume of published works, demonstrating the diversity of methodologies and applications available. In this sense, the concepts of each system's component will be based on six selected books [160, 161, 162, 163, 164, 165]. These books provide a comprehensive and detailed overview of the tools and methodologies for developing a modular IoT-based environmental monitoring system. In the following, the specific contributions of each book to this work are discussed, highlighting their relevance and application to the development of the proposed system.

The data flow, or data pipeline, has its first stage in the process of capturing information through sensors distributed throughout the forest. As highlighted in [160], the data acquisition should be carried out as close to the source as possible, preferably by electronic methods, to ensure accuracy and efficiency in collection. Using specialized hardware devices to automate data collection is a best practice as it minimizes human error and increases the reliability of a system.

Sensors used in forestry environments must be capable of operating in adverse conditions and providing accurate data on multiple parameters, such as temperature, humidity, and other factors. In [161], the importance of using robust and reliable sensors that can operate continuously without maintenance is highlighted. Sensors distributed at different strategic points can provide a comprehensive view of the forest, allowing early detection of fires.

The data collected communication by sensors to the central system is a critical challenge in forest environments. Using low-power communication networks, such as LoRaWAN, can effectively ensure data transmission over long distances with low power consumption. In this

sense, after the transmission, to manage communication between edge devices and the data center, in [160], it is suggested to use message brokers such as **Message Queuing Telemetry Transport (MQTT)**. Furthermore, gateways (which may be connected to the internet via cellular networks or satellites) for data aggregation and transmission may be necessary in areas that are difficult to access.

In environmental monitoring systems, where data is transmitted via the Internet to a control center, Node-RED can be helpful for configuring data processing workflows and system integrations. This is because Node-RED works as a Middleware, which, according to [164], is able to connect hardware or communication to software applications. Furthermore, this tool, which is based on visual programming, is notable for its easy interconnection of **IoT** components and web services. Therefore, the flexibility and ease of use of Node-RED are emphasized to create modular and integrated data pipelines, allowing the construction of complex systems without extensive coding.

According to [161], to store data in time series format, it is preferable to use a tool based on streaming data, which allows quick access to historical data. In this way, through efficient storage of the collected data, it is possible to carry out subsequent analyses based on the historical of each node. In environmental monitoring systems, such as forest fire detection, the data collected is often time series data, in which the data acquisition is performed at regular time intervals. **Time Series Database (TSDB)** systems are specifically designed to handle large volumes of time series data, offering robust performance for read and write operations, as well as advanced query and aggregation functionality.

Among the diverse **TSDB**, [165] highlights two of them: InfluxDB and Prometheus. The first, InfluxDB, is one of the most popular **TSDB**, it combines in-memory caching and disk storage to ensure durability and performance - it can handle up to 250000 writes per second. The second, Prometheus, is another **TSDB** that is widely used, especially in monitoring and alert systems. It offers a robust query interface and native support for real-time aggregations and operations.

NoSQL databases, such as Cassandra and MongoDB, are recommended to store large volumes of unstructured data and respond to an analytical query in real-time. According to [163], these databases offer the flexibility and scalability needed to store and manage **IoT** data. Furthermore, cloud storage can be a helpful tool as it provides scalability and high availability, allowing data to be accessed and analysed from anywhere. In [164], the cloud services (i.e., AWS S3 and Google Cloud Storage) are presented to store large volumes of data securely and efficiently. These cloud storage services also offer backup capabilities, ensuring continuity of operations even in cases of failure.

Fire monitoring systems can process real-time data by accessing data through database queries. In this sense, [162] describes how streaming processing systems, such as Apache Spark Streaming and Apache Kafka, are essential for supporting real-time queries and processing streaming data. These systems allow real-time data analysis and correlation, quickly identifying anomalies that could indicate the start of a fire. Processing frameworks can be scalable when applied to data pipelines. According to [163], these frameworks support the execution of complex multitasks in real-time or on a schedule, allowing continuous analysis and the generation of immediate alerts.

Furthermore, processing frameworks can facilitate integration with other data analysis and storage systems.

The implementation of anomaly detection algorithms in real-time data can be an effective solution to identify forest ignition situations quickly. Additionally, [162] states that it is possible to integrate ML and deep learning algorithms to improve accuracy in fire detection using supervised, unsupervised, and reinforcement learning techniques. These algorithms, commonly implemented in Python, can be trained with historical and real-time data (this data can be from a node or a group of nodes within a WSN), continually adapting to new conditions and patterns detected in the forest. Then, alert systems based on rules or ML, as described by [160], can be configured to trigger alerts when certain thresholds are exceeded. These alerts can be integrated with notification platforms like PagerDuty or through text messages and email.

The importance of an intuitive and responsive Graphical User Interface (GUI) to facilitate data interpretation is described in [160], where it is highlighted that a good User Interface (UI)/User Experience (UX) is crucial for the effectiveness of monitoring dashboards. In addition to sensor data, it is essential to monitor the health of each system component, including sensor connectivity status, server load, and network performance. This ensures that all components operate correctly and allows quick identification of failures or performance degradations.

Data visualization can be performed through interactive dashboards that display sensor information and the system status in real-time. Tools similar to Grafana are widely recommended for this purpose, as they allow the creation of custom charts and dashboards that facilitate visual data analysis. According to [165], Grafana integrates well with TSDB, such as InfluxDB and Prometheus, providing a robust solution for monitoring and visualizing metrics. An example of a visualization dashboard might include graphs that show sensor data over time - this data might be raw or from some analysis by an automatic algorithm. In addition, status indicators for each sensor and system component can also be displayed. This allows operators to view current conditions and identify any potential issues quickly.

Data security is a central concern in distributed monitoring systems. In this sense, [162] emphasizes the necessity for Cybersecurity to protect data, such as encryption, password security, and access levels. To achieve this, [162] describes the possibility and importance of applying encryption for the whole data pipeline, as well as implementing access controls to limit who can create, read, update, or delete anything. Similarly, [161] points out the importance of using encrypted connections and authentication mechanisms to guarantee data integrity and confidentiality. Furthermore, it is crucial to implement access control policies to ensure that only authorized users can access each system component, access data, or settings.

According to [161], it is crucial to adopt a scale-out architecture to ensure that the system can multiply as needed. This involves adding new servers or processing instances as demand increases, allowing the system to maintain performance even with a significant increase in data volume. Therefore, [161] discusses the use of scalability techniques, such as creating multiple instances or message brokers, to support high data ingestion rates. As an example of a scalability solution, containerization and orchestration technologies, such as Docker and Kubernetes, can facilitate the

implementation of a scalable and resilient architecture.

Based on the analysis of nine books, the recommended architecture for an effective environmental monitoring system integrates the following key components: Transmission, Storage, Processing, Visualization, Security, and Scalability. The selection of frameworks to implement each of these key components can be achieved by examining how other related works have overcome the challenges of integrating systems. In this sense, a study using these key components for environmental monitoring is necessary to identify best practices and potential improvements for the proposed system.

4.2.1 Related Work on Key Components of Environmental Monitoring Systems

To find related works that address these key components, the following search terms were used in academic databases between 2018 and 2022: ("environmental monitoring system" OR "system architecture") AND ("fire" OR "wildfire") AND "Lorawan" AND "wireless sensor network" AND "low-cost" AND "open-source". By alternating the key components in each query and using academic search engines, including Scopus, Web of Science, IEEE Xplore, Science Direct, and Google Scholar, a total of 59 distinct papers were obtained from the five most cited of each year.

The following subsections explore the applications and tools each of these works employed to develop the key components in its approach. By examining these implementations, this study aims to identify best practices and potential improvements for the proposed system.

4.2.1.1 Transmission

To analyse the Transmission component, 44 papers were reviewed. After careful analysis, 32 were discarded due to incomplete descriptions of data acquisition, transmission, and reception processes in a central system. Therefore, the remaining 12 papers presented a detailed description of how **LoRaWAN** monitoring systems collect, transmit, and send data to a central system.

All selected works emphasize the efficiency and simplicity of **LoRaWAN** networks. According to [166], data collected by sensor nodes is transmitted to a gateway, which then forwards it to a central server (often the cloud). The server manages communication, filters duplicate packet copies, and handles queries from multiple sources. In [167], it is stated that the primary function of a gateway is to convert data from radio frequency to **IP** format.

There are differences in server configuration among the selected papers, particularly in the choice of data management service after reception by the **LoRaWAN** gateways. This analysis categorized the services into three groups: **TTN**, ChirpStack, and other methods.

Applications using TTN

Five papers describe the use of **LoRaWAN** combined with the **TTN** service. In these cases, data is transmitted from sensors to gateways and then forwarded to the **TTN** web service via **MQTT**. For instance, [168] highlights the efficiency of **LoRaWAN** for long-range communication and low

power consumption, with data formatted in JSON for intuitive front-end visualization. Although this combination enables efficient and secure communication, the authors do not detail other tools or platforms used for data analysis.

Similarly, [169] uses **TTN** to manage **LoRaWAN** sensor networks, transmitting agricultural data such as soil temperature and humidity via **MQTT**. The paper underscores the simplicity and effectiveness of these technologies but lacks details on further data processing and visualization tools.

Two other works extend the use of **TTN** and **MQTT** by integrating cloud services for data processing and visualization. For example, in [170], **LoRaWAN** sensor data is sent to **TTN** via **MQTT** and then to cloud platforms such as ThingSpeak and Virtuino, simplifying analysis and visualization of air quality parameters such as nitrogen dioxide and temperature. Similarly, [171] describes data transmission to cloud storage services such as Elastic Stack, enabling continuous environmental monitoring, including temperature, humidity, and soil stability.

Finally, [172] discusses a system where data is transmitted to the cloud or stored locally, with **LoRaWAN** concentrators operating in edge or cloud modes. Data is published to **MQTT** topics and can be processed locally or in cloud centers, integrating with visualization and analysis tools. This setup allows for simplified analysis of parameters such as temperature, humidity, and gas levels.

Applications using ChirpStack

Two papers discuss the use of ChirpStack. In [173], a private **LoRaWAN** architecture was implemented using ChirpStack, including its Network and Application Servers, Redis and PostgreSQL databases, and the Mosquitto **MQTT** broker. The system transmits **GPS** data and text messages from sensors to a central gateway (static or mobile), with data managed by a custom server and a web interface for network management. The system supports two-way communication and integration with several sensors for flexible data collection.

Similarly, in [174], ChirpStack is used as the **LoRaWAN** network server for an environmental monitoring infrastructure. Sensor data, including temperature, humidity, and ground deformation, is transmitted via **LoRaWAN** to the ChirpStack server. The data management system incorporates the Elastic Stack Suite for storage and analysis, enabling robust data collection and export via **MQTT** to other applications.

Applications using Other Methods

Five works discuss custom data transmission services from gateways to central systems. In [175], data from sensor nodes is collected via **LoRaWAN** and transmitted to the cloud through an ESP8266 WiFi module. This setup was tested on a farm, successfully transmitting data from nodes to the gateway. In [176], the gateway sends data to a big data analytics platform via a USB serial port, where it is stored and processed in real-time. Similarly, [177] configures the gateway to relay data to a central system via RS-232 communication, with data processed and stored on an industrial computer.

Through a proprietary storage service, in [178], a smart greenhouse monitoring system uses an MQTT server to store and process data from the LoRaWAN gateway, which is then displayed on a dashboard for analysis. A similar approach is taken by [179], where coordinators based on small computers such as Raspberry Pi receive data via LoRaWAN and retransmit it through microservices using WiFi.

4.2.1.2 Storage

A review of 28 papers related to the term “Storage” after LoRaWAN transmission identified that many lacked specific technical details, leading to the exclusion of 18 papers. The remaining 10 papers provided detailed examples of storage implementations, including frameworks such as InfluxDB and Prometheus, which are discussed below to study the Storage key component in depth.

Data in LoRaWAN-based monitoring systems is often stored in distributed databases and big data platforms for efficiency and scalability. For instance, [180] describes storing IoT data in a MongoDB database locally on the gateway, with cloud transmission via standard Application Programming Interface (API). Similarly, [181] stores sensor data in a Structured Query Language (SQL) database, accessible via RESTful APIs.

The ThingSpeak platform is used for storage in [169] and [170], handling data such as soil moisture and air quality metrics. In [179], data from smart beehive systems is stored in Microsoft SQL Server and Neo4j databases, managed via a microservices architecture. Another approach described in [182] involves converting data for compatibility with the oneM2M platform for standardized access.

Cloud-based storage solutions are detailed in [171], where an AWS Elastic Compute Cloud instance hosts components such as Node-RED and the Elastic Stack. For large-scale data storage and processing, the framework Hadoop Distributed File System (HDFS) and Spark are used, as seen in [183] and [184], integrating Hadoop for storage and MapReduce for distributed processing.

The database for monitoring systems, described in [185], is based on Cyber-Physical Systems (CPS) integrated with Building Information Modeling. The system uses a Raspberry Pi to collect data from sensors that measure temperature, humidity, air quality, noise, vibration, and lighting. This data is transmitted in real-time via the MQTT protocol (which can also be sent via LoRaWAN) to a TSDB named InfluxDB. Then, InfluxDB stores sensor metadata, including timestamps and observation data, ensuring efficient and organized storage of the collected data. Other system components can access the stored data via queries.

Similarly, in [168], the developed system aims to provide early warnings for COVID-19 infections based on air quality sensors. This system uses a network of LoRa sensors to measure particulate matter levels and monitor Indoor Air Quality (IAQ) in real-time. The data collected by the sensors is saved in an InfluxDB database, which is optimized for storing and querying large volumes of time series data, enabling efficient analysis of air pollution levels over time.

4.2.1.3 Processing

In **IoT**-based monitoring systems, data processing transforms raw data into refined information. A review of 42 articles on “Processing”, were excluded 32 due to insufficient detail. Accordingly, the remaining articles were selected and analysed individually with the aim of obtaining explicit examples and detailed methodologies that may be applicable to data processing in systems proposed in this work. The selected works presented different approaches to face the challenges of data processing, highlighting the use of advanced frameworks and tools such as Apache Spark, MATLAB, Python, and several **ML** libraries. Below, each of these selected works is highlighted, exploring the leading technologies and methodologies that drive the processing component of **LoRaWAN**-based monitoring systems.

Key examples include [171], where a robust landslide monitoring system uses Node-RED for initial data capture, Kafka for real-time processing, and Python **ML** libraries like Pandas, Scikit-learn, and TensorFlow for predictive modeling, all hosted on AWS for scalability. Similarly, [179] describes a beehive monitoring system using a microservices architecture with C#, Python, and Kotlin for multiple components. In [170], an air quality monitoring system processes data using ThingSpeak, integrating several sensors for real-time analysis. Meanwhile, [186] uses WEKA software to create predictive models to detect falls and other user positions. With WEKA, it is possible to apply algorithms such as J48 (decision tree algorithm’s Java implementation).

A system in [187] integrates local and cloud processing for search and rescue, using LabVIEW for real-time data analysis. MATLAB is used in [169] for analysing humidity data in an irrigation system, optimizing water use based on real-time analysis. Other notable systems include [188], which implements a hybrid processing architecture to maximize energy efficiency, and [168], which uses PHP scripts for real-time data processing and visualization in an air quality monitoring system. The Fuzzy method is applied in [189] for real-time fire index prediction, combining sensor and satellite data. Lastly, [180] describes a system using FIWARE Orion Context Broker for data management, Elasticsearch for analysis, and a complex event processing engine for real-time decision-making.

4.2.1.4 Visualization

The visualization component of an **IoT**-based environmental monitoring system is vital for interpreting data and making informed decisions. Of the 24 papers initially identified, 8 were selected for detailed analysis due to their robust visualization implementations. These selected works highlight the application of advanced frameworks and tools for effective data visualization in **LoRaWAN**-based monitoring systems. Below are summaries of each selected paper, showcasing diverse methodologies and tools employed in data visualization for environmental monitoring systems.

In [181], a web application using RESTful **APIs** presents real-time water level data on interactive dashboards, including graphical visualizations and maps. Similarly, [167] developed a web and mobile interface for monitoring olive trees, displaying real-time data transmitted via **LoRaWAN**.

A web application in [171] visualizes high-precision sensor data using Django and Bootstrap, integrating interactive maps and charts through Leaflet JS and Kibana. In [176], Node-RED is used to develop a platform for real-time irrigation monitoring, featuring an interactive graphical interface.

Grafana is employed in [168] for visualizing air quality data, offering customizable dashboards and visual alerts for critical pollution levels. Similarly, [179] uses ASP.NET and Kotlin for web and mobile applications to visualize honeycomb conditions, supporting multiple data formats for transparency. In [170], ThingSpeak and Virtuino are used to create dashboards for real-time air quality monitoring, with color-coded visualizations representing different air quality levels. The system in [178] uses ThingsBoard for greenhouse monitoring, featuring interactive graphs that facilitate decision-making. Finally, [173] proposes using the TeamUp platform for real-time data visualization in emergency scenarios, integrated with ChirpStack, Redis, PostgreSQL, and Mosquitto for comprehensive network management.

4.2.1.5 Security

Security in IoT platforms is essential to guarantee data integrity, confidentiality, and availability. Out of 43 reviewed papers, 11 were selected for their explicit focus on security measures within LoRaWAN-based monitoring systems.

Security vulnerabilities in long-range systems are discussed in [190], such as Man-in-the-Middle and replay attacks. Which highlights availability, authentication, and integrity attacks on end nodes during Over-The-Air-Activation (OTAA) and ABP activation. To solve these problems, encryption in LoRaWAN is performed using the Advanced Encryption Standard (AES)-128 algorithm in counter mode. LoRaWAN is noted for being the only long-range technology with AES-128 encryption, as supported by [191]. Encryption security during transmission is also distinguished in outdoor monitoring systems, as discussed by [167, 168, 170].

According to [192], IoT devices face security issues due to inadequate security designs. The vulnerabilities include eavesdropping, spoofing, and skimming attacks at the perception layer, DDoS attacks at the network layer, and SQL injections at the application layer. In [176], data security in smart agriculture systems is ensured through password-authenticated sessions, allowing only authorized users to access and modify system data. Similarly, [36] suggests using blockchain to secure communication in smart homes based on LoRaWAN, emphasizing the importance of data privacy.

A cloud-based platform for environmental monitoring is described in [180], where data security is maintained through user authentication and encrypted Hypertext Transfer Protocol (HTTP) protocols, with access control based on user roles. In [193] is addressed the security of WSN-based precision agriculture systems, and implemented encryption and secure HTTP communication to protect data integrity and employing multi-factor authentication for access control.

The LoRaWAN-based pipeline monitoring system, described in [194], details a monitoring system using encryption functions such as AVR-Crypto and XXTea, alongside a frequency agility-based interference avoidance algorithm, to ensure secure and uninterrupted communication.

4.2.1.6 Scalability

Scalability is a key feature in **IoT** monitoring systems, especially those based on **LoRaWAN**. It refers both to the system's ability to handle increasing numbers of devices and data, and to the potential for duplicating the central system in new locations. Out of 31 papers reviewed, 7 provided explicit strategies for scaling system architecture.

Aiming to identify vulnerabilities and secure architectures, [170] discusses a modular system integrating fog computing, Software-Defined Networking, and blockchain to enhance scalability and security, applying Docker for containerization. Similarly, [176] explores scalability in smart irrigation systems through a modular architecture combining **LoRaWAN**, Node-RED, and **MQTT** on Raspberry Pi devices, enabling efficient data collection across large areas.

A low-cost **IoT** platform for environmental monitoring in African countries is presented in [180], emphasizing modular and scalable architectures with technologies such as **LoRaWAN**, Node-RED, and Docker to manage large data volumes and integrate new components. On the other hand, [195] compares Docker Swarm and Kubernetes for orchestrating containers in fog and edge computing, focusing on improving network efficiency and reducing latency, particularly in smart healthcare systems. The study highlights Docker's role in securing and managing additional devices.

The use of **Google Kubernetes Engine (GKE)** as a fundamental platform for providing crop-related advice in agricultural **IoT** systems is emphasized in [196]. It emphasizes the use of **GKE** for implementing microservices in agricultural **IoT** systems, facilitating data analysis and scalability in adapting to different scenarios. Similarly, [179] discusses microservices architecture for beehive monitoring, highlighting its scalability and modularity in handling multiple simultaneous requests. And [192] proposes a Docker-based architecture, NETRA, for enhancing smart home security through edge traffic analysis, offering superior performance and scalability compared to virtual machine-based platforms.

4.2.2 Summary of Key Components of Environmental Monitoring Systems

This section reviewed several related works and the literature on the main components of an **IoT**-based environmental monitoring system. The system's main components were based on the concepts and frameworks presented in nine books. Additionally, 59 papers from different academic databases provided a detailed analysis of practical implementations and methodologies used for monitoring systems: transmission, storage, processing, visualization, security, and scalability.

- **Effectiveness of transmission:** The effectiveness of using **LoRaWAN** for long-distance data transmission in real applications was highlighted.
- **Data Storage:** Databases such as InfluxDB are used for efficient storage of time series. Middleware Solutions: Middleware, such as ThingSpeak and Node-RED, can act as a bridge between different systems or applications, facilitating communication and data management between them.

- **Data Visualization:** Data visualization was emphasized as an essential component, with tools such as Grafana providing intuitive and interactive interfaces.
- **Data Processing:** Data processing was highlighted, using Python and libraries such as Pandas, NumPy, and Scikit-learn to enable advanced analysis and ML.
- **Platform Security:** Platform security was considered vital, with robust access controls and data encryption to ensure the integrity and confidentiality of information.
- **Scalability:** The architecture's scalability can be achieved with the use of Docker containers, which allow easy replication and distribution of services in different environments.

The contributions of the reviewed articles were crucial in identifying the best practices and most efficient technologies for each component of the system. The analysis of the books provided a solid theoretical basis and allowed a more structured and cohesive approach to the development of the proposed system. The efficient integration of all these components is essential for the success of an environmental monitoring system.

None of the reviewed papers explicitly mention that a system using the composition of all mentioned components can be maintained reliably at critical moments, such as wildfires. Therefore, the next section is dedicated to creating, configuring, and detailing the architecture to ensure no failures during monitoring. The specific configuration is presented, addressing the redundancy mechanisms and resilience practices necessary to operate the system continuously in adverse situations.

4.3 Development of the Monitoring System Architecture

Based on the in-depth study performed in the previous section, which reviewed several works and literature on the main components of an IoT-based environmental monitoring system, this section is dedicated to detail the creation and configuration of an integrated architecture. The proposed architecture is described with all the necessary and pertinent details to ensure its proper functioning in emergencies, addressing the redundancy mechanisms and resilience practices required to operate the system continuously in adverse conditions. This initial configuration is based on best practices identified in the literature and aims to serve as a solid foundation for the monitoring system.

To ensure that the proposed system works as expected, an initial assessment of the components, without any changes, must be carried out. This approach allows for the validation of the effectiveness of the components in a controlled environment before any modifications or optimizations. Therefore, an in-depth understanding of each component can contribute to a critical and objective analysis of system performance, enabling the identification of potential failures or strengths in each layer.

After this initial assessment, changes could be incorporated as necessary to optimize system performance and reliability. Adjustments can be included at any point in the data pipeline. Thus, they can consist of anything from changes to sensor parameters or payloads, changes to gateway configuration, or even cloud data management and visualization tools.

The environmental monitoring system is designed to integrate multiple technologies and components that work together collaboratively. The system architecture is composed of sensor modules, **LoRaWAN** gateways, cloud **IoT** components, communication protocols, microservices, web services, user interfaces, and a centralized control center. Figure 4.1 illustrates the system architecture in diagram mode, showing how the layers are configured and connected.

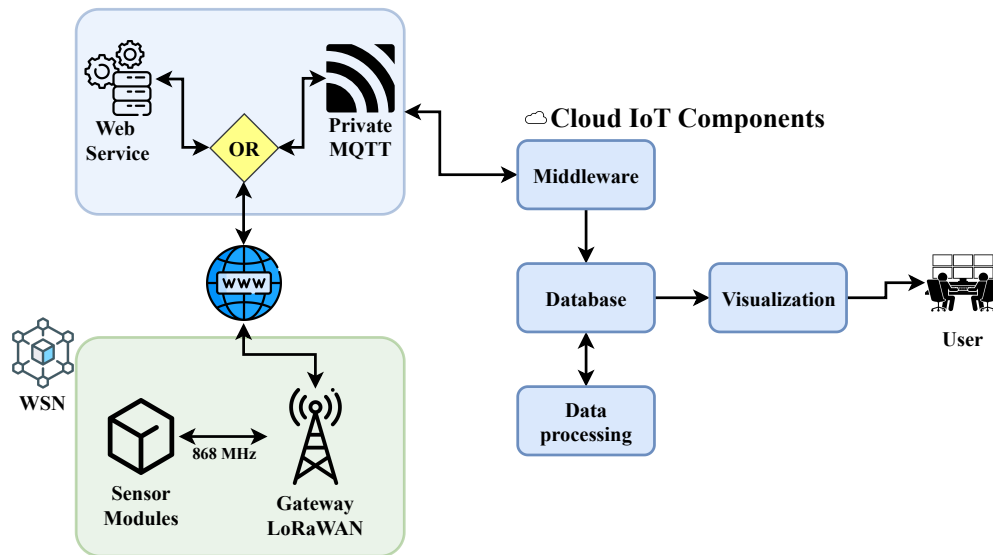


Figure 4.1: Base System Architecture for Environmental Monitoring, derived from literature review.

The **WSN** responsible for identifying forest ignitions is composed of several sensor nodes. It is strategically distributed in areas at risk of fire to collect environmental data, such as temperature, humidity, and **IR** intensity. Each **WSN**'s element transmits data via **LoRaWAN** at a frequency of 868 MHz, ensuring long-distance communication with low power consumption.

LoRaWAN gateways must be strategically positioned, considering the characteristics of the terrain (relief, forest density, among others) and the available infrastructure close to the monitoring location (internet connection via mobile network or ethernet, and energy sources such as power grid, solar panels, or wind generators).

The collected data is received by the **LoRaWAN** gateways, which retransmit the payload with the information acquired by the **WSN** to the **IoT** components in the cloud. Therefore, the gateway is responsible for transforming the radio frequency data into **IP** data. Furthermore, data can be integrated into the cloud through a web service that acts as a bridge. The **MQTT** protocol facilitates communication between these different system components, guaranteeing the transmission of data from the gateway to the cloud. Additionally, both **TTN** and ChirpStack can be used as services, and a proprietary **MQTT** service can also be useful.

Consequently, the web service, as soon as it receives the data, publishes it to an **MQTT** topic. In the cloud, the **MQTT** topic is constantly monitored by middleware, such as Node-RED or ThingSpeak. When listening to new data on the **MQTT** topic, this middleware sends it to storage

in JSON format. An specialized **TSDB** need to be applied, which is responsible for managing all the data in the system. And, therefore, it must transform the data from JSON format to time series format, that is, it assigns a timestamp each new data is recorded.

Once the data is stored, microservices are employed to process and analyse it in real-time through **ML** libraries. As each component of the system operates independently and scalably, microservices can perform functions ranging from managing data ingestion to processing, storage, and visualization. Thus, microservices can be used in any part or layer of the system.

The processed data is returned to **TSDB** to be stored in a different database. In this way, the entire history of processed data can be accessed through a web service, allowing users to access and view the information collected and processed in real time or on any recorded date. For visualization, tools are integrated into the system to function as a dashboard web service. With these dashboards, it is possible to detect real-time patterns and make informed decisions quickly.

The system is supervised by a centralized control center, which integrates all the layers mentioned, allowing continuous management and rapid response to critical events. The architecture also includes robust security mechanisms, such as access control and data encryption, to ensure the integrity and confidentiality of information.

In the following subsections, the specific configurations of each component chosen to form the system architecture is detailed, including the tools and technologies used, as well as the redundancy mechanisms and resilience practices implemented to ensure the system's continuous operation in adverse situations.

4.3.1 Selection of Data Acquisition Tools

Ensuring efficient and safe data collection requires selecting data acquisition tools that attend to system installation requirements. Therefore, this subsection discusses the use of the **MQTT** protocol and the **TTN** platform [197] to be inserted as web server tools in the data transmission layer. Additionally, the possible configuration in the system's context and the security and benefits of these two tools are detailed.

Although it is possible to implement a proper **MQTT** server using tools such as Eclipse Mosquitto [173], it was chosen to use **TTN** due to its ease of implementation and its consolidated infrastructure in terms of security, best practices and policies, privacy and authentication. **TTN** offers a reliable and well-established platform that facilitates the configuration and management of **LoRaWAN** networks, as well as providing additional benefits such as interoperability and scalability.

The main advantage of using **TTN** instead of a private **MQTT** service is the possibility of sharing the **LoRaWAN** network with multiple applications and users. A proprietary **MQTT** server would have exclusive access to its owner, which can complicate sharing the installed infrastructure for other purposes. With **TTN**, any application configured on the platform can uplink or downlink real-time data through any **LoRaWAN** gateway configured with a **TTN** server. This allows different projects and users to benefit from the same infrastructure without dependence on a developer.

On the other hand, according to the related work presented in the last section, it would still be possible to use the ChirpStack service, as it can function as a **LoRaWAN** web server. However, certain factors were considered when implementing **TTN** and ChirpStack services:

- **Ease of Implementation and Use:** **TTN** offers a more user-friendly interface and is easier to configure, especially for first-time users. **TTN**'s comprehensive documentation and active community provide additional support, facilitating troubleshooting and rapid implementation [169].
- **Scalability and Interoperability:** **TTN** is highly scalable and interoperable, allowing different applications and users to share the same **LoRaWAN** network infrastructure. While ChirpStack is equally powerful, it may require more complex configurations to achieve the same level of interoperability and scalability [171].
- **Community Benefits:** While ChirpStack also has an active community, it does not offer the same breadth of support and resources available on **TTN**. **TTN** has an extensive global community that contributes resources, solutions, and support, simplifying the implementation of new features and resolving issues. This can contribute to the scalability of the system developed in this work, as future users could overcome difficulties or add or modify features [168].

The **TTN** platform offers an environment for managing **IoT** devices for each user, which can also be collaborated with other users. In this way, users can create application layers, which facilitates the organization of projects that use the platform as a web server. Within each application layer, it is possible to assign each node sensor a unique `End device ID` that is registered on the **TTN** platform.

This registration allows authentication and encryption of transmitted data, ensuring the security and integrity of the information. Furthermore, it is possible to identify which node sensor the metadata in **TTN** services belongs to through its **ID**. Consequently, each metadata is available with the integration of the **MQTT** service and can be “listened to” or “published to” the corresponding topic. Therefore, both the downlink and the uplink follow the same structure but with opposite directions in the data flow.

TTN uses application keys to encrypt transmission data, providing an additional layer of security. These switches can be used for two different transmission modes: **ABP** and **OTAA**. **ABP** method uses keys such as **Device Address (DevAddr)**, **Network Session Key (NwkSKey)**, and **Application Session Key (AppSKey)** for authentication and encryption. On the other hand, **OTAA** requires **Application Key (AppKey)** and uses **Device Unique Identifier (DevEUI)**, **Application Unique Identifier (AppEUI)** for secure joining and generation of **NwkSKey** and **AppSKey** dynamic keys.

The data sent or received by sensors through the **TTN** service is formatted in a couple of bytes, and, therefore, it needs to be encoded or decoded to be interpreted correctly. Then, users can assign an encoder or decoder directly on the **TTN** platform. The encoder is responsible for converting

data from human-readable JSON object format to couple-byte format before sending. On the other hand, the decoder interprets the received data and converts it into a human-readable JSON object format, allowing the information to be processed and analysed efficiently. An example of how the decoder is configured in the **TTN** can be seen in Algorithm C.1.

Another significant benefit available by **TTN** is the ability to fetch data from the last 24 h. This functionality can be helpful to ensure data continuity in the system, even if a service goes offline temporarily, whether due to maintenance or other reasons. **TTN**'s storage integration allows to store upstream messages in a persistent database, which can be later retrieved via the **HTTP API** or gRPC service. This feature is configurable per application or end device on the **TTN** platform, facilitating historical data management.

4.3.2 Development of Process Management Tools

Once the tools used in the data pipeline have been defined for the data to reach the Cloud **IoT** components, it is necessary to choose the tool that will act as middleware. In this context, two tools emerge as candidates: Node-RED [198] and ThingSpeak [199]. Therefore, this subsection is intended to compare these two tools based on the application of the monitoring system proposed in this work.

Although these two tools have different purposes and functionalities, comparisons can be made in several aspects, particularly in how each platform manages data flow. Furthermore, the decision between the two tools directly impacts the configuration and usage of other components. Therefore, it is essential to consider the paths and implications that each choice will entail to guarantee the efficiency, scalability, and robustness of the monitoring system.

Node-RED is a flow-based development tool that allows easy integration of different **APIs**, devices, and services. Its intuitive **GUI** enables rapid configuration of data flows, which is ideal for **IoT** applications where real-time data collection and processing are essential. In the context of forest monitoring, Node-RED can be configured to collect sensor data with environmental parameters and forward this data for storage and processing.

ThingSpeak is a cloud-based data analytics and **IoT** services platform that facilitates sensor data collection, visualization, and analysis. Its user-friendly interface allows quick configuration of channels for receiving data and offers integrated tools for processing and visualization. In the context of forest monitoring, ThingSpeak can be configured to receive data from sensors with environmental parameters, store this data in the cloud, and provide real-time graphics and basic analysis. ThingSpeak also supports integration with MATLAB for more advanced analytics, offering a simplified solution for prototyping and monitoring small **IoT** projects.

Based on the characteristics and capabilities of Node-RED and ThingSpeak, the following are the main aspects considered in this comparison, including ease of use, flexibility, integration capacity, development and maintenance, and scalability:

- **Ease of Use:** Both tools offer user-friendly interfaces. Node-RED has a drag-and-drop graphical interface, allowing users to create intuitive data flows. ThingSpeak also provides

a friendly user experience, enabling easy configuration of channels for receiving data and viewing graphs [164].

- **Flexibility and Customization:** Node-RED offers a vast library of plugins that can be configured to perform a wide range of tasks, from integrating with APIs to running custom scripts. This allows for extensive customization to attend to almost all project requirements. ThingSpeak, on the other hand, is more restrictive in terms of customization and is better suited for specific use cases [176].
- **Integration with Multiple Protocols:** Node-RED supports multiple protocols, including MQTT, HTTP, and WebSockets, among others. This facilitates communication between different system components and integration with external services. Although ThingSpeak supports some of these protocols, its flexibility is limited compared to Node-RED. This may restrict the ability to integrate ThingSpeak with several devices and third-party services [171].
- **Development and Maintenance:** Node-RED allows for more agile development and simplified maintenance due to its drag-and-drop interface for connecting functions and easy visualization of workflows. Node-RED's modularity also turns it simple to update and maintain data flows. ThingSpeak, while easy to use for basic setups, can require more effort to set up and maintain complex integrations, especially for handling large volumes of data or the need for advanced analytics [171].
- **Scalability and Performance:** Node-RED is highly scalable and can be deployed in diverse environments, from local devices to cloud servers. Its ability to manage large volumes of data and process information in real-time makes it ideal for environmental monitoring applications such as forest fire detection. ThingSpeak, although scalable, may not offer the same level of performance in high-demand scenarios as it is a tool that runs in the cloud and, by default, is only used online. This may limit its responsiveness in critical or high-load situations [180].

Considering all these factors, Node-RED was chosen as the process management tool due to its flexibility, ease of use, and ability to integrate with a wide variety of protocols and services. Although ThingSpeak is a powerful platform for IoT, Node-RED offers significant advantages that make it more suitable for developing this system.

4.3.2.1 Implementation of Process Management with Node-RED

Node-RED can act as a central component in the proposed system architecture, serving as a hub to coordinate communication between different components, direct tasks, and manage data flows. In this sense, the main process flows implemented in Node-RED are detailed in this subsection.

Node-RED can continuously listen to data received on the MQTT topic from TTN and manage this data in real time. Each incoming payload can be transformed into a suitable format and forwarded to a database. This process ensures that the environmental data collected by the sensors is stored efficiently and is available for future analysis. To ensure data continuity, Node-RED could

be configured to implement a service that requests historical data from **TTN** every 12h. This service uses **TTN**'s storage **API** to retrieve data from the last 24h, ensuring that no information is lost due to temporary communication failures or microservices maintenance.

To perform advanced analytics and data processing, microservices code in Python can be integrated with Node-RED. These microservices use libraries like Pandas, NumPy, and Scikit-learn to perform data analysis and **ML** tasks. Node-RED can facilitate the execution of microservices through terminal runs, allowing complex algorithms to run efficiently and scalably. To visualize the location of sensor modules, Node-RED can integrate a mapping service that uses map **APIs** such as Google Maps or OpenStreetMap. This service receives **GPS** data inserted by the user during the nodes' fixation process and sends this information to a database for storage. Thus, it is possible to visualize the location of the nodes on a dashboard, as well as the distribution of the **WSN**.

Signal quality is a parameter to be observed in the **LoRaWAN** network. In this sense, Node-RED can include a service that saves signal quality data during the transmission process of each node's payload through any **LoRaWAN** gateway. This data can be analysed to monitor network performance and identify possible changes in module locations, aiming to improve communication. To optimize sensor positioning and ensure adequate coverage, Node-RED can implement a **LoRaWAN** signal quality measurement service at different points in the monitored area. This service can collect **LoRaWAN** signal quality data obtained by test modules in different locations and generate a quality map that visualizes the areas with the best and worst or no signal reception, helping to plan the sensors' distribution effectively.

With the choice of Node-RED as a process management tool, the next step in the data pipeline sequence is to select a **TSDB** for integration into the system to ensure the storage of environmental data collected by the sensors. Choosing the appropriate database to store this data is essential to ensure the integrity, accessibility, and real-time analysis capacity of the collected data. Therefore, this subsection discusses the implementation of data storage, comparing different databases that support time series to determine the most appropriate one for the proposed system.

Following the direction pointed out by the applications in the related work in the previous section, four database service attend the requirements of being open-source and compatible with integration with Node-RED. These four options are Cassandra [200], InfluxDB [201], MongoDB [202], and Prometheus [203]. The characteristics of each database concerning implementation in an environmental monitoring system are described below.

- **Cassandra** is a NoSQL database that offers high availability and horizontal scalability. Designed to handle large volumes of data distributed across multiple servers, this platform is ideal for applications that require scalability and redundancy. One of Cassandra's main advantages is its peer-to-peer architecture, which eliminates single points of failure and ensures high availability. However, the complexity of configuration and maintenance can be a significant disadvantage. Additionally, although Cassandra is scalable, it is not specially optimized for time series data, which can complicate queries [163].

- **InfluxDB** is widely recognized for its efficiency in storing and querying time series data. It supports fast writes and queries, allowing the insertion of large volumes of data per second. This platform organizes data using a schema of “tags” and “fields”, where tags are indexed key-value pairs that allow efficient querying, and fields are the actual data values. For example, in a sensor monitoring application, a tag might be the sensor **ID** or location, while fields could be temperature, humidity, and **IR** intensity readings. InfluxDB is particularly suitable for environmental sensor data due to its ability to handle large amounts of real-time data. Furthermore, it offers integrated data compression and automatic retention policies, ensuring efficient data management. However, InfluxDB’s focus on time series data means it may be less flexible for other types of data [185, 168, 165].
- **MongoDB** is a NoSQL database known for its flexibility and scalability. It stores data in JSON documents, making it ideal for unstructured or semi-structured data. Although MongoDB can store time series data, its structure is not optimized for this type of data, which can result in lower efficiency for time series queries and storage. Flexibility in data modeling and the ability to perform complex queries and aggregations are significant advantages. However, MongoDB may require more resources for real-time queries and operations on large volumes of time series data compared to specialized databases [180].
- **Prometheus** is a popular choice for monitoring and alerting, especially in infrastructure systems and microservices. It offers a query interface and is excellent for real-time aggregations and operations. Prometheus uses a time series data model optimized for system metrics and applies pull-based collection. This architecture is highly efficient for real-time monitoring, but its long-term storage capacity is limited. To overcome this limitation, Prometheus must often be integrated with long-term storage systems such as Thanos or Cortex, especially when large volumes of historical data are needed. This need for additional integration can be a disadvantage for those looking for a more integrated solution [165].

Considering wildfire monitoring systems, which require the most reliable real-time storage and querying of time series data, InfluxDB is the most appropriate option. Its specialization in time series data, combined with its ease of use and ability to handle large volumes of data during both read and write operations, ensures it is an ideal solution for the proposed forest fire detection and monitoring system.

4.3.2.2 Implementing Data Storage with InfluxDB

InfluxDB is ideally suited for integration with Node-RED, providing an efficient and scalable solution for storing time series data. Using a plugin available in the Node-RED libraries, data collected by **WSN** is easily forwarded to InfluxDB in JSON format.

To save data from **WSN**, Node-RED could be configured to listen for data received on the **MQTT** topic from **TTN**. Upon receiving data, Node-RED might select, through a code function, which data to send to InfluxDB, forwarding it in JSON format, and indicating the tags, fields, and

database. With a pre-configured write query in the plugin, data in JSON format would be sent to InfluxDB for storage. An example of the JSON format to be applied in a write query is provided in Appendix C - Algorithm C.3.

WSN raw data stored in InfluxDB could be processed by microservices integrated with Node-RED. These microservices would obtain data through read queries, supported by Node-RED or by communicating directly with InfluxDB via API. Read queries should follow the established standard for a given database, including the database name, the desired fields and tags, as well as the time interval. After processing and analysis, such as predictions or risk indications, the microservice's output might be saved in a separate database. In this context, the microservice would perform the reverse process of the read query, using a write query to InfluxDB with the necessary information for storage. An example of constructing a read query to access data from an InfluxDB database can be seen in Appendix C - Algorithm C.4

It is possible to implement backup and data recovery strategies to ensure data integrity and continuity in the event of system failures. InfluxDB supports automatic and periodic backups, allowing to create incremental and full backups of time series data. Setting up regular backups ensures that critical data, such as environmental sensor readings and analysis results, is protected from loss. In the event of failures, data recovery procedures allow to quickly restore data from backups, minimizing downtime and ensuring that the monitoring system continues to operate efficiently. Implementing a strict backup policy is essential for system resiliency and real-time monitoring reliability.

Similarly, other services, such as TTN storage data collection services and the storage of modules' GPS positions, can also benefit from integration with InfluxDB.

4.3.3 Integration of Dashboard Visualization Tools

With the choice of combining InfluxDB as the TSDB and Node-RED as the process management tool, the next step is to select the data visualization tools that will be integrated into the system. Visualizing data in a real-time environmental monitoring architecture can add significant value. Through appropriate representation, environmental conditions can be interpreted graphically, and dashboards can assist in making informed decisions quickly. This subsection discusses the integration of dashboard visualization tools, comparing two open-source options highlighted in the literature and related work: Kibana [204] and Grafana [205], and evaluating how they can be integrated with Node-RED and InfluxDB.

Kibana is a data visualization and analysis tool belonging to the Elastic Stack. It is widely used to search, visualize, and interact with data stored in Elasticsearch. Kibana allows the creation of interactive graphs, maps, and complete dashboards, making large volumes of data more accessible and understandable. Among its advantages are seamless integration with Elasticsearch, powerful search capabilities, and advanced visualizations. However, limited integration with InfluxDB and configuration complexity can be significant drawbacks.

On the other hand, Grafana is an open-source platform for monitoring and observability, standing out for its real-time data visualization capabilities. It is highly flexible and supports data

sources, including InfluxDB and Prometheus. Grafana offers an intuitive interface, ease of use, built-in support for InfluxDB, and an alerting system based on colors. However, although excellent for visualization, Grafana does not have the same in-depth data search and analysis capabilities as Kibana.

Considering the scope of monitoring real-time environmental data and the integration with InfluxDB, Grafana stands out as the most appropriate choice for the environmental monitoring system. Grafana allows the creation of highly interactive dashboards that can be customized to display the environmental data collected by **WSN**. With real-time visualization capability, users can continuously monitor environmental conditions and quickly detect any anomalies or significant changes.

Additionally, users can configure colored alerts based on predefined thresholds for parameters such as temperature, humidity, and **IR** intensity. Grafana can be easily integrated with Node-RED to receive processed data and with InfluxDB to query and visualize stored time series data. Dashboards to be viewed in Grafana can include: Line Graphs (Time Series), Bar Charts, Gauge Indicators, Heat Maps, Geolocation Maps (Worldmap Panel), and Colored Alert Panels, among others.

Integrating Grafana with Node-RED and InfluxDB ensures that data is continuously updated on dashboards, allowing for efficient monitoring and rapid response to any critical event detected by environmental sensors. With this approach, Grafana's use in the environmental monitoring system provides a powerful and efficient solution for visualizing real-time data - essential for detecting and responding to forest fires.

4.3.4 Security and Scalability Considerations

Security and scalability issues must be addressed when technologies and components are integrated into the environmental monitoring system's architecture. In this sense, this subsection discusses the security practices and scalability strategies implemented to protect data and ensure the system's ability to handle future growth.

With the union of these four main elements — **TTN**, Node-RED, InfluxDB, and Grafana — some security measures are obtained from these tools throughout the data pipeline, from data transmission to visualization system dashboards:

- **Data Encryption:** All flux data is encrypted using the **AES-128** or **Transport Layer Security (TLS)** protocol. First, **TTN** uses **AES-128** encryption for data transmitted between sensors and the **LoRaWAN** network, ensuring that information is protected against interception and man-in-the-middle attacks. Second, the **TLS** protocol guarantees that information transmitted between each component in the cloud (Node-RED, InfluxDB, and Grafana) is protected against interceptions.
- **Access Control:** **TTN** provides authentication and authorization for devices, ensuring that only registered devices can send data. Furthermore, access to system components (Node-RED, InfluxDB, and Grafana) is restricted to authorized users, using key-based authentication

and access control. This ensures that only individuals with appropriate permissions can view or modify the data.

4.4 System Architecture and Implementation

This section describes all the practical implementation steps of the environmental monitoring system using the tools and technologies selected in the previous section. Similar to the previous section, the system's implementation is detailed in stages and described according to the order in which the data flows from **WSN** to the visualization on the dashboard. To provide a clear view of the system as a whole, Figure 4.2 illustrates the system's general architecture, showing how each component interacts within the data pipeline with the selected components.

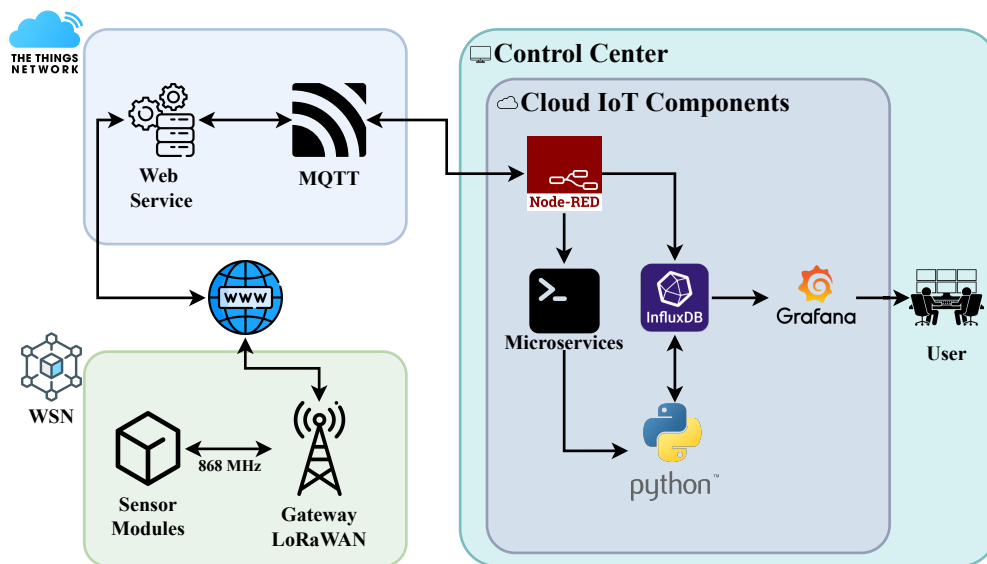


Figure 4.2: Environmental Monitoring System Architecture Components and Integration.

Starting from the **WSN** layer, the first stage of development and description addresses the configuration of **LoRaWAN** authentication: the registration of devices in **TTN** and the configuration for data transmission via **LoRaWAN**. Next, the configuration of each Node-RED service is described, including the creation of data flows and the description of the services created for:

- Service for saving signal quality data from sensors.
- Service to get data from **TTN** storage.
- **Network-attached storage (NAS)** service for backup InfluxDB.
- A **GUI** to assist in the identification of modules' locations.
- Service to measure **LoRaWAN** signal quality at different points.

- Service for real-time data processing and analysis.

The configuration of InfluxDB is covered, emphasizing the setup, database configuration, and the method used through examples of write and read queries. An example of data restoration is also presented.

Grafana's application is presented next, including creating dashboards, configuring colored alerts, and visualizing data in real-time. In addition, data processing and analysis is detailed too, focusing on microservices that should be integrated with Node-RED to perform analysis and processing tasks using **ML** libraries and data analysis tools.

A final step aims at system scalability approaches, demonstrating how the system can be replicated through installation in a **Virtual Machine (VM)** or containerization via Docker.

4.4.1 LoRaWAN Configuration on Modules and TTN.

Adding each node to the **LoRaWAN** network through an account on the **TTN** portal involves accessing the console after login, navigating to the `End Devices` section, and selecting `Register End Device`. On the subsequent page, the option `Enter end device specifics manually` should be activated, and the parameters configured according to Table 4.1.

Table 4.1: LoRaWAN Configuration Table.

Filed	Selection
Frequency Plan	Europe 863-870 MHz (SF9 for RX2 - recommended)
LoRaWAN Version	LoRaWAN Specification 1.0.2
Regional Parameters Version	RP001 Regional Parameters 1.0.2 revision B
Activation Mode	ABP
Device Address	Generated ID
AppSKey	Generated a Key
NwkSKey	Generated a Key
End device ID	Node's name

When the registration is performed by clicking the `Register end device` button at the bottom of the page, it is necessary to configure some parameters in the 5FTH firmware to obtain authentication on the **TTN** network. These parameters must be copied from the corresponding fields on the **TTN** portal: **DevAddr**, **NwkSKey**, and **AppSKey**. Then, each **TTN** parameter must be inserted into the corresponding variables in the firmware, as shown in Appendix C - Algorithm C.2.

After entering these values, the firmware can be uploaded to the module's microcontroller, and the 5FTH can be installed on tree trunks to begin monitoring. It is crucial to ensure that each module has a unique set of parameters. If two identical uploads are performed, it could cause conflicts on the network, resulting in duplicate data, loss of information, and difficulty in correctly identifying the origin of the data. Therefore, the configuration process must be repeated for each module to be added to the **WSN**.

Upon installation of 5FTH on-site, data will begin to be sent. As previously mentioned in Section 3.3.5, the payload will contain information about Battery, Temperature, Humidity, five **IR** intensities from different positions, and an **AUX** variable for communication control and DROWSIER filter output. When the payload arrives at **TTN**, it is decoded using the Appendix C

- Algorithm C.1 code. The data is then forwarded to integration with TTN's MQTT, allowing Node-RED to process and manage this data.

4.4.2 Node-RED Configuration

After configuring the sensors and ensuring that the data is transmitted securely to the TTN network, the next step is to receive it in the cloud components for storage, processing, and transformation into useful information for environmental monitoring. With its ability to integrate different APIs, devices, and services in a visual and intuitive method, Node-RED becomes a central piece in the system architecture.

The choice of Node-RED, as discussed in the previous section, is due to its flexibility, ease of use, and ability to integrate with multiple technologies, including MQTT, HTTP, and third-party APIs. Its development GUI allows the creation of work environments, designated as “flows”, which can be built with simple drag-and-drop operations, enabling quick configuration and modification of processes as needed. Moreover, Node-RED provides a feature to export flows, which can be shared and reused. In this context, Appendix C.3 includes all the flow codes in JSON format. Once Node-RED is installed and configured securely, with a login and password for access control, the configuration of the process flows necessary for the system to work as expected can be built. Therefore, each of the services implemented in Node-RED is described below.

Reading WSN Raw Data

Reading the raw data transmitted by the WSN is the first step in the Node-RED process flow to be detailed. To read the payload sent by the WSN, a subscription block using the MQTT plugin is used to subscribe to the TTN topic. This topic is responsible for receiving or sending any information between the Cloud and WSN layers.

When a payload from any WSN element is received, metadata is obtained from the topic. This metadata contains the identification for each 5FTH module, transmission and signal quality, timestamp, and sensor data. To process each piece of this information, new function blocks must be prepared. In Figure 4.3, the function block is responsible for processing a specific part of the payload and forwarding the data to the InfluxDB database.

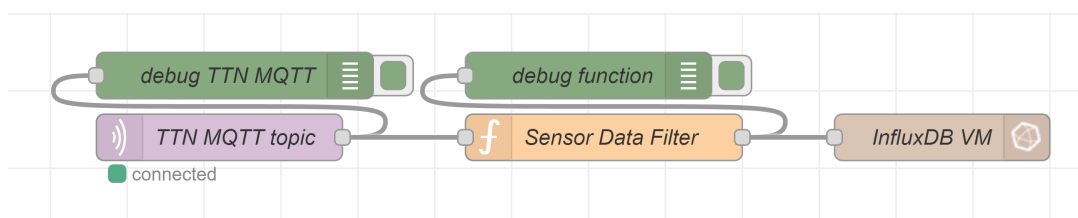


Figure 4.3: Node-RED flow for processing data from the TTN MQTT topic. Available in Appendix C - Algorithm C.5.

The selected data represents the ID of each module and the sensor data, which is forwarded to InfluxDB, where it is stored in a time series database for later analysis and visualization.

Service for saving signal quality data from sensors

Signal quality is an important parameter that must be observed in the **LoRaWAN** network. In this sense, signal quality data can be extracted and processed similarly to raw sensor data using the same subscription block for the **MQTT** topic of **TTN**. Therefore, this service stores signal quality data during each transmission, allowing network performance to be monitored and possible adjustments to the location of modules can be identified.

Each module's signal quality is saved individually, ensuring that data specific to each device is recorded. Furthermore, as modules can transmit using more than one gateway, a division process is performed (illustrated in Figure 4.4) to save the signal quality of each gateway separately. This makes it possible to obtain signal quality data from different gateways and analyse this data separately, allowing a detailed view of network performance at different access points.

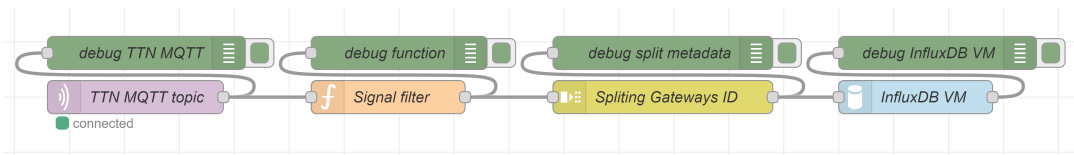


Figure 4.4: Node-RED flow for processing and storing signal quality data. Available in Appendix C - Algorithm C.6.

Service to GET data from TTN storage

Once the sensor data and signal quality are stored, it is necessary to GET the historical data from **TTN** storage. This service ensures that data is recorded continuously, even during temporary communication or system maintenance failures.

The flow shown in Figure 4.5 is an automation that handles JSON messages, interactions with an InfluxDB database, and queries to an **HTTP API**. This flow is structured to operate cyclically and process device data periodically.

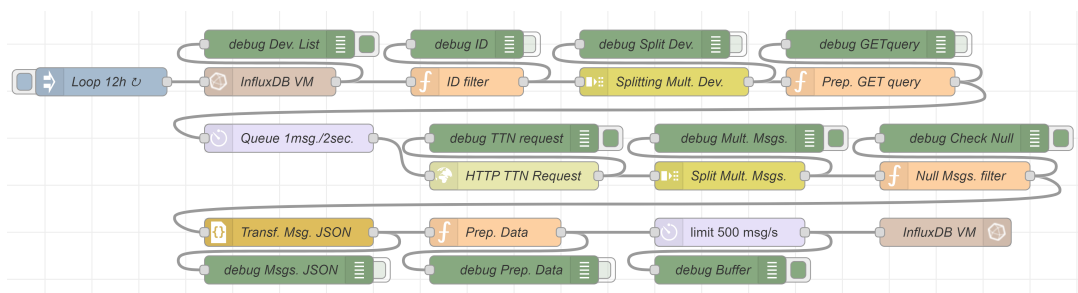


Figure 4.5: Node-RED flow for GET data from TTN storage. Available in Appendix C - Algorithm C.7.

The process starts with an injection block `Loop 12h`. This block is configured to trigger the flow every 12h, sending an initial message that starts the entire subsequent process.

The first important step is to query the InfluxDB database, performed by the `InfluxDB VM` block. This block runs a query to get a list of devices from the bucket. The response to this query is then sent to the `ID filter` block, which processes the response, extracting the `ID` of the devices.

Then the `Prep. GET query` prepares the URL and `HTTP (GET)` method to query `TTN API` using the `ID` of filtered devices. This block constructs the URL needed to get data from each specific device.

The `HTTP` request is executed by the `HTTP TTN Request` block, which sends the request to the `TTN API` and waits for the response. The received response is processed by the `Split Mult. Msgs.`, which divides messages into multiple parts in order to be processed individually.

After splitting, the `Null Msgs. filter` filters any null messages, ensuring that only valid messages are processed. The `Prep. Data` prepares the received data by extracting the device `ID`, decoded payload, and timestamp. This data is formatted appropriately for insertion into InfluxDB.

The processed data is finally stored in InfluxDB by the `InfluxDB VM` block, which inserts the data into the specific bucket. To avoid system overload, delay blocks such as `Queue 1msg./2sec.` and `Limit 500msg./1sec.` are used to control the message sending rate, ensuring that the system can handle the volume of data efficiently.

NAS service for backup InfluxDB

Ensuring the integrity and security of data stored in InfluxDB guarantees the continuity and reliability of the environmental monitoring system. Therefore, implementing a regular backup service is essential, and thus, this service ensures that data is saved periodically, allowing recovery in the event of failures or data loss.

The flow shown in Figure 4.6 is an automation that performs periodic backups of the entire InfluxDB every 24h. The process starts with a time injection block `Loop 24h`, configured to trigger the flow daily at 0h01, and, consequently, this starts the backup process.

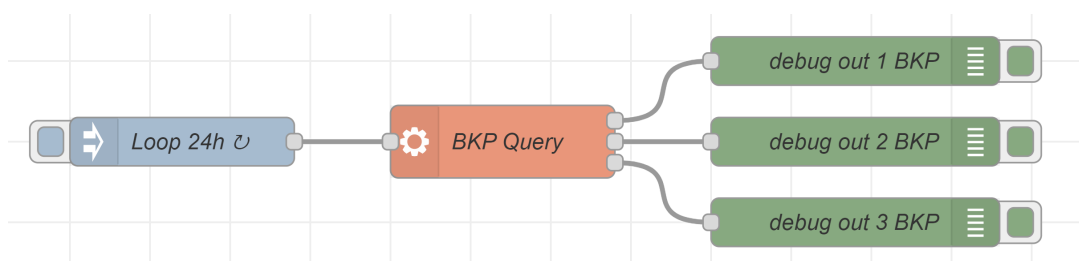


Figure 4.6: Node-RED flow for periodic InfluxDB backup. Available in Appendix C - Algorithm C.8.

Next, the `BKP Query` block executes a system command to perform a backup of the InfluxDB database. The command used is:

```
influx backup /mnt/nfs/BackupsInflux/backup_$(date '+%Y-%m-%d_%H-%M')
-t eyJrIjoizmFrZXRVCI6IkZBSImMiOiJNSVRFIn0==
```

This command creates a backup of the entire InfluxDB and saves it to a specified directory with a name that includes the current date and time, ensuring that each backup is unique. This directory can be a **NAS**. Note: the token used above is fictitious and has been changed for security purposes.

A GUI to assist in the identification of modules' locations

Following the implementation of services for reading raw **WSN** data, signal quality and all backup involved, it becomes essential to accurately map and track the locations of each sensor module within the monitoring area. Node-RED offers the capability to create a **GUI** that simplifies the management and visualization of module locations, allowing users to manually input and store geospatial data.

The flow shown in Figure 4.7 is an automation that enables users to select an InfluxDB database, list available devices, manually input coordinates, and store the coordinates in a geospatial database.

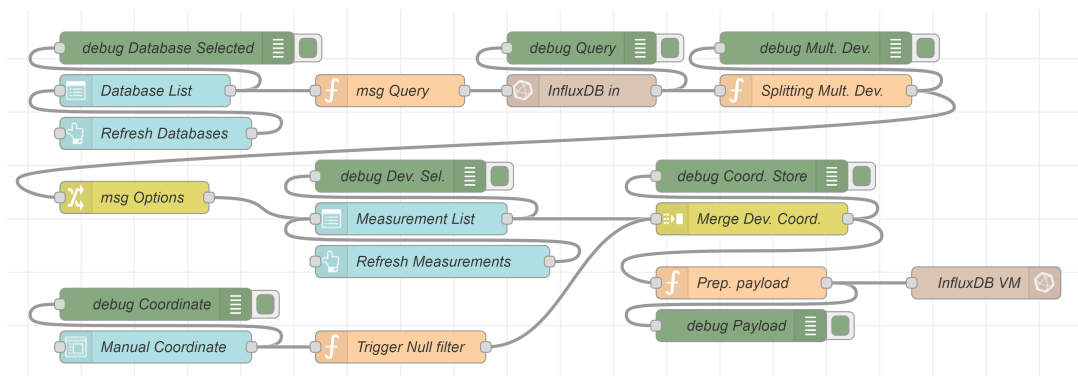


Figure 4.7: Node-RED flow for managing and storing module coordinates. Available in Appendix C - Algorithm C.9.

The process starts with the `Database List` block, a dropdown menu where the user can select one of the available databases. Upon choosing a database, the `msg Query` block builds a query to receive a list of devices from this database and sends it to the `InfluxDB in` block, which executes the query.

The query response is then processed by the `Splitting Mult. Dev.` block, which extracts the devices' **ID** and prepares them to be displayed in another dropdown menu named `Measurement List`. This menu allows the user to select the specific position measurement to be updated in the database (or create a new entry if the module is brand new).

Additionally, the flow includes a form named `Manual Coordinate`, where the user can manually input coordinates (latitude, longitude, and altitude). When the form is submitted, the data is sent to the `Trigger Null filter` node, which checks if the coordinates are null before triggering the merge message.

The selected device **ID** and the manually entered coordinates are then merged by the `Merge Dev. Coord.` block, outputting this information into a single message. This message is

processed by the `Prep_payload` block, formatting the data into an appropriate payload for database insertion.

Finally, the `InfluxDB_VM` block stores the formatted payload in the geospatial database, ensuring that the manually entered coordinates are associated with the device `ID` and timestamp to be stored correctly for later analysis.

The flow also includes two refresh buttons in case a user misclicks or needs to access the most updated information about databases and devices:

- **Refresh Databases:** This button, when clicked, updates the list of available databases, allowing the user to obtain the latest list for selection. It is connected to the `Database List` block.
- **Refresh Measurements:** This button, when clicked, updates the list of available devices in the selected database. It is connected to the `Measurement List` block.

This flow provides an interactive `GUI`, displayed in Figure 4.8, for selecting databases and available devices, manually entering coordinates, and efficiently storing this data.

Figure 4.8: GUI for storing module coordinates.

Service to measure LoRaWAN signal quality at different points

Following the configuration and use of the `GUI` to store module locations, it is essential to assess the `LoRaWAN` signal quality across different points within the monitoring area. This ensure network performance and identifies areas with weak signal coverage that may require additional attention.

For this purpose, a service to measure `LoRaWAN` signal quality was implemented. The process involves using a mobile device with `GPS` capabilities to collect signal quality metrics as it moves through the monitoring area. Alongside this mobile device, a module with `LoRaWAN` transmission is employed. By combining data from these two devices, its possible to determine the available `LoRaWAN` signal strength and the corresponding `GPS` positions without using the `LoRaWAN` bandwidth.

The merged data from these devices includes `GPS` coordinates and signal quality parameters, which are then stored and analysed to generate a comprehensive signal quality map. This service

enables the collection of signal-quality data from several locations, providing valuable outputs into the network's performance. The collected data can be used to create signal quality maps, aiding in the optimization of module placement and network configuration.

This methodology was detailed and published in a previous work, which can be accessed for a more in-depth explanation and implementation specifics [206]. The paper, titled “Map coverage of LoRaWAN signal's employing GPS from mobile devices”, provides a thorough overview of the techniques and outcomes of the signal quality mapping process.

Service for real-time data processing and analysis

Executing Python-based microservices can be implemented for real-time data processing and analysis, supporting decision-making in environmental monitoring. This step leverages Node-RED's capabilities to integrate ML algorithms and data analysis processes, improving the system's ability to provide accurate and timely information.

To achieve this, Node-RED can be configured to run .py scripts, which contain ML models and data analysis routines. These scripts can be triggered based on specific events or schedules, ensuring data processing occurs in real-time. The flow in Figure 4.9 illustrates an example of implementing this service. The process starts with a trigger action, which initiates the execution of Python scripts. This trigger can be event-based (e.g., new data received) or time-based (e.g., periodic analysis).

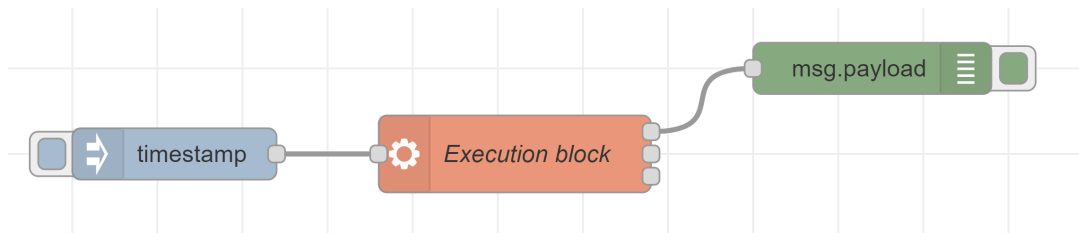


Figure 4.9: Example of Node-RED flow for real-time data processing and analysis. Available in Appendix C - Algorithm C.10.

Multiple scripts can perform distinct tasks, including data preprocessing, feature extraction, and applying ML models for predictions or anomaly detection. Microservices, when activated, can access data stored in InfluxDB (through read queries via Python) to fulfill their predetermined tasks. After the script execution is completed, the results must be stored in InfluxDB, and a write query via Python can also be performed to store the microservices' output.

Although it is possible to run some analyses using the function block, running Python files through the Node-RED execution block is recommended to avoid possible crash or freezing issues in Node-RED that data processing and analysis could cause. In this sense, the microservice being launched via the execution block will be run in the operating system layer, thus avoiding conflicts with Node-RED.

Implementing these microservices can be a complex and critical task, especially when the WSN is dense, as identified in the previous chapter. The requirement to perform ML in a high-density

WSN makes these microservices vital to ensuring system efficiency and accuracy. For convenience, the next chapter is dedicated to the elaboration of each of these scripts, in which the **ML** techniques used and how they are executed will be explored in detail based on the example indicated above.

4.4.3 InfluxDB Configuration

According to the data pipeline, after configuring Node-RED to manage real-time process flow, the next step is configuring InfluxDB to store time series data efficiently and in a structured method. As previously mentioned, InfluxDB was chosen due to its specialization in storing and querying time series data, offering a powerful solution for environmental monitoring systems.

The choice of InfluxDB is fundamental to the system architecture, as it allows the organization and management of large volumes of data from environmental sensors. Its integration with Node-RED facilitates real-time data transfer, enabling fast and accurate analysis. The installation of InfluxDB must follow the developers' recommendations, as described in the official documentation. Once installed according to official guidelines, InfluxDB needs to be configured, ranging from creating databases to data storage modes and creating tokens for different uses, such as reading and writing.

Within InfluxDB, data is organized using measurements, tags, and fields. This structure allows flexible and detailed queries, which is essential for monitoring environmental variables provided by a **WSN**. The "measurements" represent different categories of data, such as the module's name that performed the acquisition data; "tags" are metadata that help identify the origin or other characteristics of the data; and "fields" contain the values of the measurements.

At least five types of buckets must be implemented in the proposed environmental monitoring system. This is because, depending on the type of application, the data needs to be stored with some specific considerations. For example, tags should only be used when necessary or require careful strategies. Tags are indexed to enable fast and efficient queries, which increases disk space consumption. Therefore, it is essential to interpret each process flow that requires data storage to determine the best way to structure it, ensuring efficiency and space savings.

According to the applications managed by Node-RED, as discussed in the previous subsection, writing queries are performed in five different applications. Starting with storing the raw data from **WSN**, it is decided to use an exclusive bucket named `SAFe_project`. The `SAFe_project` bucket has a specific structure for store the data, where the measurement of this bucket represents the module's **ID** that acquired the data, and the field is used to store the sensor read values. This same structure is followed when creating the bucket that stores data from the **TTN** Storage service, as the data has similar characteristics. Therefore, the bucket created is `SAFe_storageTTN` for writing queries from this service.

The modules' location service used during the installation on the tree trunk, through the developed **GUI**, stores the data following a similar logic, only changing the field values. In this case, the `SAFe_positions` bucket is responsible for storing the geospatial coordinates of the modules, which structure the node's **ID** as measurement, and the latitude, longitude, and altitude coordinates are written as fields. The reason for creating a separate bucket for **GPS** positions

is to avoid unnecessary disk space consumption increases and improve data organization. **GPS** coordinates, if added together with the raw data, could significantly increase the complexity and volume of the data, as well as complicate location-specific queries and analysis.

The `SAFe_metadata` bucket is used as a storage of signal quality data, such as **RSSI**, **SNR**, frequency, **BW**, **SF**, and channel, for each module and gateway that transmits the data. In this bucket, the measurement attribute is the gateway name used for data transmission. The tag is the module **ID** that made the transmission, and the field is the signal quality value. In this context, if a module transmits data through more than one gateway, it is possible to perform individual signal analysis, as the data is stored separately by the measurement attribute. A summary description can be seen in Table 4.2.

Table 4.2: Buckets' structure with their respective measurements, tags, and fields.

Bucket	Measurement	Tags	Fields
SAFe_project	Node ID	None	Timestamp, Battery, AUX, Humidity, Temperature,
SAFe_storageTTN			IR1 , IR2 , IR3 , IR4 , IR5
SAFe_positions			Latitude, Longitude, Altitude
SAFe_metadata	Gateway name	Node ID	RSSI , SNR , Frequency, BW , SF , Channel

The fifth bucket, declared `SAFe_analysis`, is dedicated to storing the output data generated by the real-time processing and analysis microservices. This data results from **ML** scripts and other advanced analyses performed by the system, which transform raw data and metadata into helpful information for decision-making.

The structure of the `SAFe_analysis` bucket must follow a similar logic to other buckets, with measurements, tags, and fields organized to optimize subsequent queries and analyses. Measurements can represent the names of different analyses or predictions performed, while tags can include relevant metadata to categorize and identify the analysed data. The field should contain the values resulting from the analyses, such as predictions of environmental parameters and anomaly detections, among others. As the output of analysis processes needs to be clearly defined to ensure the efficiency and accuracy of queries, the next chapter is dedicated to detail and describe how the output data is organized and stored in the `SAFe_analysis` bucket.

When performing read or write queries in InfluxDB, it is necessary to include the access token along with the bucket name and parameters: measurements, tags, and fields. This ensures that the system authenticates and authorizes the query operation. For example, the service that stores raw **WSN** data needs a write token for the `SAFe_project` bucket, while the service that stores signal quality data needs a write token for the `SAFe_metadata` bucket. In this sense, write tokens ensure that only authorized processes can add new data to the system, preventing unauthorized insertions. Similarly, read tokens are configured to allow query of data already stored in a bucket. Read tokens can be used by services or users authorized to access the data for analysis or visualization. For example, when configuring a visualization dashboard to monitor real-time environmental data, Grafana might have a read token to access the data in the `SAFe_project` bucket.

To perform read queries and access data stored in InfluxDB, a client must be used to communicate with the database. The example in Appendix C - Algorithm 1 indicates how to configure and run a query to obtain data from a specific module within a time range by using a function `DATA_GET(start, stop, module)`.

In addition to reading, to insert data into InfluxDB, it is necessary to use an API client that allows communication and the execution of write queries in the database. The example in Appendix C - Algorithm 2 shows the configurations to execute a query to write data to a specific module and bucket using the function `DATA_WRITE(data_to_write, module)`.

Through Python scripts, it is possible to read and write data in InfluxDB. This flexibility allows the microservices developed in the next chapter to integrate these database operations into a given data analysis seamlessly. Also, using Python to manipulate data from InfluxDB makes implementing ML algorithms easier.

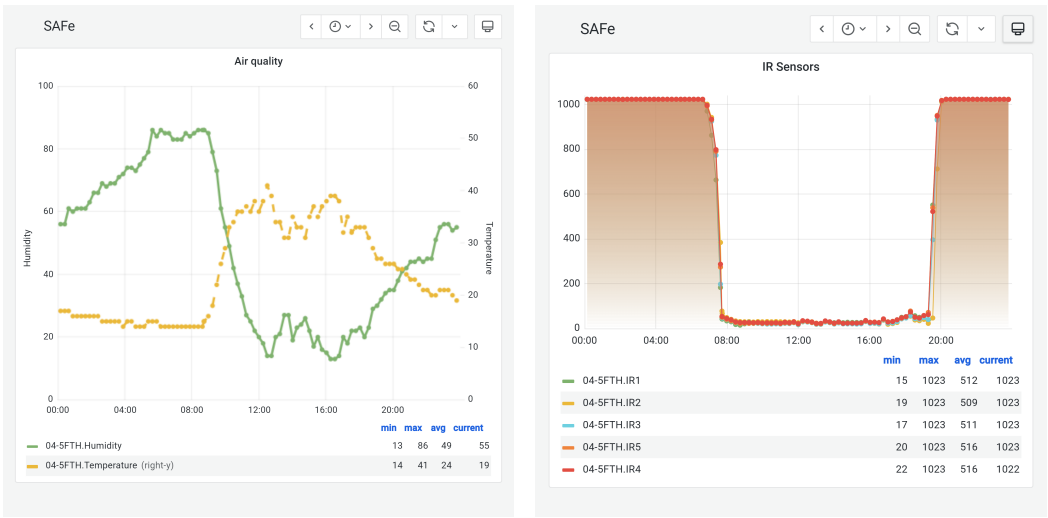
4.4.4 Grafana Configuration

Visualizing data is an indispensable feature of any environmental monitoring system. After configuring InfluxDB for structured time series data storage, the next step is to set up Grafana for real-time data visualization. As previously mentioned, Grafana was chosen for its powerful capabilities in creating interactive and customizable dashboards, which are fundamental for monitoring environmental parameters.

The choice of Grafana as a visualization component in the system architecture allows the creation of detailed and dynamic visual representations of the data collected by the WSN. Its integration with InfluxDB ensures that real-time data can be displayed in different compositions, facilitating quick decision-making.

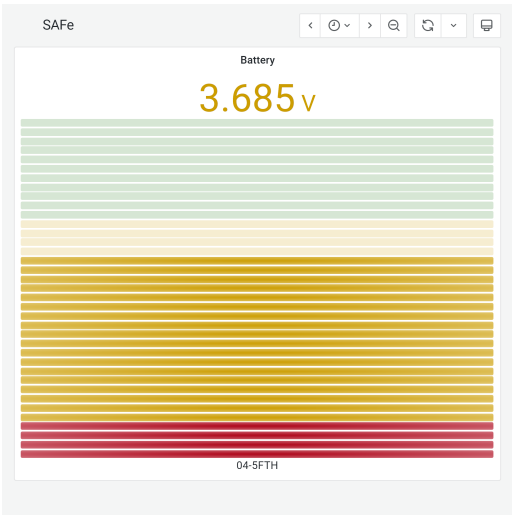
Adding InfluxDB buckets as a database source requires Grafana to be setting up to access data with the necessary authorization to read any bucket. By dedicating a token with read permissions, five buckets can be added as data sources. In this context, it is possible to create several types of dashboards by simply selecting the data source by its name and its original application. Despite the possibility of changing the name of each data source during the configuration, it was decided to remain bucket names as the data source names. Therefore, the five data sources in Grafana are `SAFe_project`, `SAFe_storageTTN`, `SAFe_positions`, `SAFe_metadata`, and `SAFe_analysis`.

Creating dashboards in Grafana involves configuring several types of graphs and panels to visualize the data collected from the sensors. Grafana supports real-time monitoring by configuring dashboards to continuously update with new data from InfluxDB. This setup ensures that the latest data is always displayed, providing an up-to-date view of environmental conditions. For example, in Figure 4.10, three panels represent the temperature, humidity, IR sensors, and battery levels. In Figure 4.10a, the time series graph displays the temperature and humidity data collected from the sensors over 24h. Similarly, in Figure 4.10b, the time series graph shows the data from the five IR sensors. The battery levels, which use colors to indicate the minimum, median, and maximum values of battery levels, are presented as a gauge bar chart in Figure 4.10c.



(a) Temperature and Humidity in Time Series Graph.

(b) IR Sensors Time Series Graph.



(c) Battery Levels Gauge Bar Chart.

Figure 4.10: Grafana Panels for Temperature, Humidity, IR Sensors, and Battery Levels.

In Figure 4.11, two panels are displayed: one indicating a regular status and the other showing a fire alert. In Figure 4.11a, the status indicator is colored blue with the text “Good”, representing regular conditions. This configuration helps quickly identify when the monitored area is within safe environmental parameters. Conversely, in Figure 4.11b, the status indicator is red with the text “Fire Alert”, signaling a critical condition that requires immediate attention. These visual indicators are configured to change color and text based on real-time data from the sensors, providing a clear and immediate understanding of the system’s status.

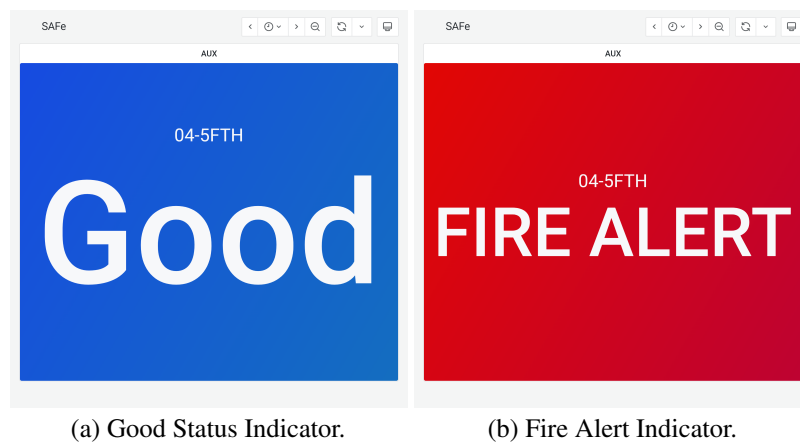


Figure 4.11: Grafana Panels for Status Indicators.

A map-based visualization can display the locations and status of sensor modules. This is particularly useful for visualizing the distribution and condition of modules across the monitored area. The map panel displays the locations of all selected sensor modules in the monitoring area, as shown in Figure 4.12. Each blue point on the map indicates that all modules are working correctly and are securely fixed. This visual representation helps operators quickly assess the geographical distribution and status of WSNs. The satellite view provides a realistic and detailed perspective, accurately identifying module positions and ensuring they are correctly placed.

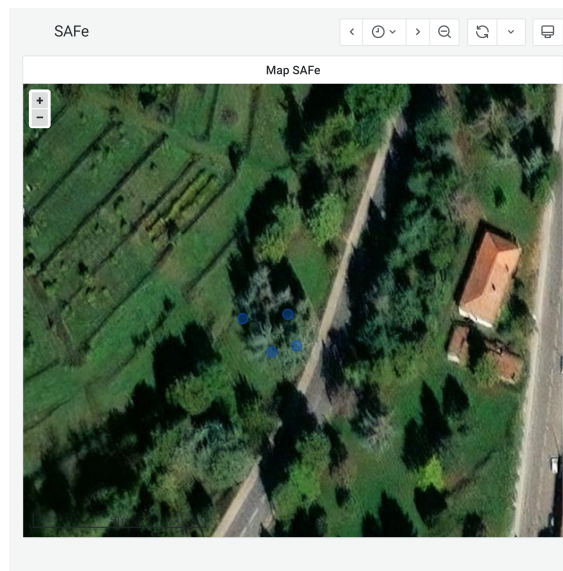


Figure 4.12: Grafana Map Panel with Sensor Module Locations.

Signal quality metrics such as SNR and RSSI are critical for assessing the performance and reliability of the LoRaWAN network. Grafana can be configured to display these metrics as heat maps, providing a visual representation of signal quality over time. In Figure 4.13, two heat maps are displayed to visualize SNR and RSSI metrics. Figure 4.13a shows the SNR values in dB over 24h, indicating the ratio of the signal power to the noise power.

Higher **SNR** values, close to 10dB, generally signify better signal quality, allowing for more reliable data transmission. The color intensity in the heat map represents the count of messages received within a one-hour interval with similar **SNR** values, illustrating the stability and quality of the signal. Similarly, Figure 4.13b presents the **RSSI** values in dBm over time, measuring the power level detected by the receiver. Higher **RSSI** values, close to 0dBm, indicate better signal strength. The heat map's color intensity reflects the count of messages received within a one-hour interval with similar **RSSI** values, demonstrating the consistency and quality of the signal.

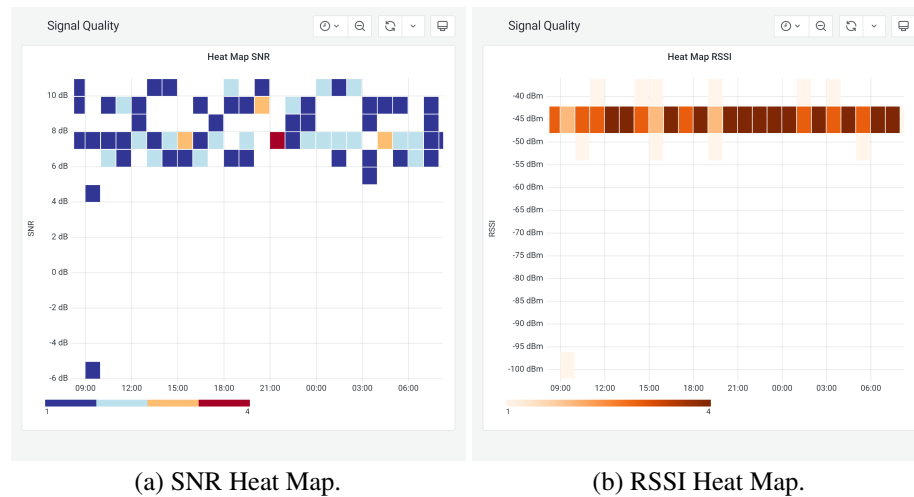


Figure 4.13: Heat Maps for SNR and RSSI Metrics.

4.4.5 Docker Implementation

With all the components configured and integrated, containerization can be applied to ensure scalability and ease of deployment. Docker, a powerful containerization platform, offers a solution to these requirements. By encapsulating applications and their dependencies into containers, Docker allows for consistent and repeatable deployments across a range of environments. In this sense, this subsection describes an approach based on Docker that can be configured to streamline the deployment and management of the system's architecture, including Node-RED, InfluxDB, and Grafana.

The main objective of adopting Docker in the system architecture is to simplify the deployment and scalability of the system without compromising the performance, execution, and security of each component. Docker offers a standardized environment, ensuring that the system works consistently across different platforms and environments. In other words, with Docker-encapsulated components, it is possible to run the system on any Docker-compatible hardware. Furthermore, it facilitates the maintenance and updating of system components, reducing downtime and configuration errors.

To better understand how Docker can be applied to the developed monitoring system, Figure 4.14 illustrates the system architecture with the layers of encapsulated components. Therefore, the added Docker layer in the system architecture starts communicating with the **TTN** and users, maintaining the same data pipeline standard as explained previously.

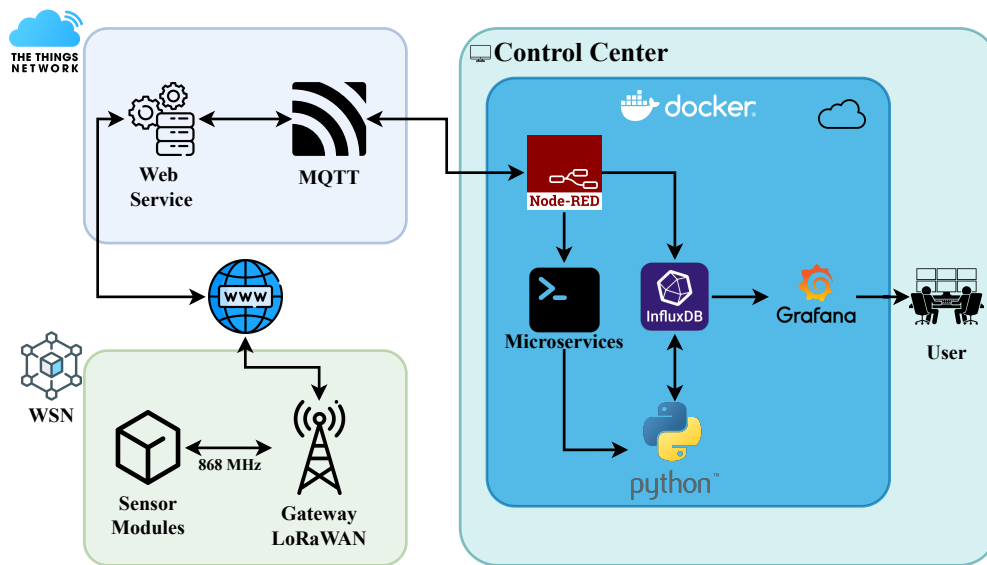


Figure 4.14: System Architecture Containerization with Docker implementation.

A `docker-compose.yml` file is configured to enable Docker to encapsulate each component. This file describes how Docker should configure, start, and interconnect Node-RED, InfluxDB, and Grafana component containers. In this sense, the `docker-compose.yml` file is presented in Appendix C - Algorithm C.11.

With the `docker-compose.yml` configuration, the Node-RED container pursues the same purpose mentioned previously - managing processes in real-time. Similarly, the InfluxDB and Grafana containers are also configured to work for the same purposes mentioned earlier - storing time series data and visualizing data, respectively.

Due to the microservices approach, running the Node-RED container requires preparing the environment to execute Python scripts. For this, a `Dockerfile` is dedicated to setting up the Python environment through external setup parameters. Therefore, the purpose of the `Dockerfile` is to install the necessary tools and libraries for data processing and analysis performed through pre-defined microservices, such as Pandas or Numpy. It is recommended that any library to be imported into the Python environment is added to the `requirements.txt` file.

The primary bucket of the InfluxDB container is `SAFe_project`. However, other buckets can be automated at creation through the `init-influxdb.sh` file, which launches four more buckets: `SAFe_positions`, `SAFe_storageTTN`, `SAFe_metadata`, and `SAFe_analysis`. If a bucket needs to be deleted or added, changes to the `init-influxdb.sh` file are recommended.

Grafana configuration focuses on pre-configuring buckets created during InfluxDB container initialization. In this sense, to avoid possible conflicts, any addition, renaming, or exclusion of buckets within the `init-influxdb.sh` file must be reflected in the `datasource.yml` file - which is responsible for pre-configuring Grafana data sources.

Launching Docker Compose prepares a virtual environment for each container, which requires maintaining some settings persistently even after reboots. To achieve this, folders are created

to store the containers' data for each component outside of the containers, ensuring that it is unnecessary to reconfigure everything from scratch when Docker is restarted. Then, the file `start` is launched to check the existence of these folders, and then Docker is launched through `docker-compose.yml`.

The environmental monitoring system was validated using three VMs with distinct hardware and software configurations. The system components were tested for durability, scalability, and performance under several conditions. The architecture supports flexible deployment on traditional Operating Systems (OS) or containerized environments, enhancing system adaptability and efficiency. In this sense, using Docker ensured standardized deployment and scalability, facilitating real-time data processing through microservices.

The VMs, configured with identical services and access to TTN, performed tasks such as data collection, backup, signal quality measurement, and real-time analysis. The analysis showed consistent data storage without service interruptions, with timestamp variations observed only on a millisecond scale during detailed inspection. This validation highlights the system's robustness and reliability across different hardware setups, demonstrating its capability to handle real-time environmental monitoring tasks with precision and minimal disruptions.

4.5 Discussion and Final Considerations

This chapter has meticulously detailed the development and implementation of the environmental monitoring system. By integrating multiple platforms such as Node-RED, InfluxDB, Grafana, and Docker, the developed system proved reliability, scalability, and efficient solution for real-time data monitoring. The following discussion elaborates on each part of the chapter, highlighting key aspects and discussing the implications of the results.

The system architecture was designed to facilitate the seamless integration of several components essential for environmental monitoring. The Bibliographic overview and related works provided a comprehensive understanding of the architecture, emphasizing the need for a modular and flexible design. By outlining the interactions between sensors, data processing units, and visualization tools, the architecture set the foundation for the detailed configurations and implementations described in subsequent sections.

The integration of MQTT and TTN services allows the gateway to be used by other users. If a proprietary service had been developed for IPB, additional means would need to be provided for other users to access the data. Another advantage of using TTN is the registration of the gateway, which, by adhering to best practices outlined by the TTN portal, can be configured with standards beneficial to other users. Moreover, relevant information about the gateway and its operations is made publicly available, fostering an environment of shared resources and collaborative development.

Node-RED is a fundamental platform in establishing a uninterrupted flow of data from the sensors to the storage and processing systems. Its visual programming interface facilitated the integration of different APIs and devices, making it an ideal choice for managing the workflow of

environmental data. The services implemented in Node-RED included reading raw data from the **WSN**, saving signal quality data, GET data from **TTN** storage, backup of InfluxDB data, assisting in identifying module locations, measuring **LoRaWAN** signal quality, and performing real-time data processing and analysis. Each of these services presented Node-RED's capability to manage real-time data and integrate with other components.

InfluxDB was chosen for its specialization in **TSDB**, which is essential for environmental monitoring systems. The configuration process involved setting up buckets, defining measurements, tags, and fields, and creating tokens for secure data access. Five distinct buckets were created to organize and store different types of data, ensuring efficient data management and retrieval. Creating read and write tokens enhanced the security of the InfluxDB setup, minimizing the risk of unauthorized data access and maintaining data integrity. Detailed algorithms for reading and writing data to InfluxDB were provided, demonstrating how Python scripts could interact with the database, enabling advanced data processing and analysis.

Grafana was configured to visualize the data stored in InfluxDB, providing an interactive and customizable interface for real-time monitoring. Each InfluxDB bucket was added as a data source in Grafana, allowing for flexible and detailed visualizations. Many dashboards were created to display data such as battery levels, signal quality metrics, and sensor readings, facilitating quick and informed decision-making. The dashboards were configured to update in real-time, ensuring that users always had access to the latest data.

Docker was used to encapsulate the system components, guaranteeing consistent and efficient deployment across different environments. The `docker-compose.yml` file described how to configure and interconnect the Node-RED, InfluxDB, and Grafana containers, simplifying the deployment process. A dedicated `Dockerfile` was created to configure the Python environment within the Node-RED container, enhancing the system's capabilities through pre-defined microservices. The `init-influxdb.sh` script automated the creation of additional buckets in InfluxDB, and the `datasource.yml` file pre-configured Grafana data sources, simplifying the visualization setup. Docker volumes ensured that data and configurations were persisted outside of the containers, maintaining system integrity and facilitating easy updates and maintenance.

The performance of the system was validated by comparing three different **VMs**, each with specific hardware and software configurations. The system successfully storage data without any interruptions in all **VMs**, validating the efficacy of the implemented services and configurations. Moreover, the comparison of the three **VMs** demonstrated the system's ability to operate effectively across different hardware configurations. Each **VM** performed the same tasks, allowing for an accurate assessment of performance.

In conclusion, this chapter has comprehensively detailed the configuration, implementation, and results of the environmental monitoring system. By leveraging Node-RED, InfluxDB, Grafana, and Docker, the system achieved robust, scalable, and efficient solutions for real-time data processing and visualization. The detailed comparison of three different **VMs** validated the system's performance and reliability across different applications, highlighting its potential for effective environmental monitoring. The integration of Docker significantly simplified the deployment and

maintenance processes, ensuring that the system could be easily scaled and updated. Additionally, the system's modular design allows for flexibility and adaptability, making it suitable for multiple environmental monitoring applications. Overall, the results indicate that the system is well-equipped to handle the demands of real-time environmental monitoring.

Additionally, the system's modularity and flexibility allow **ML** models to be easily incorporated, offering support for advanced and predictive analytics. Therefore, the necessary framework for the development and integration of **ML** techniques is complete. Thus, the natural next step is to explore the potential of these techniques to further enhance the system's ability to provide accurate and timely information for environmental decision-making.

Chapter 5

Data Analysis and Machine Learning

This chapter explores the application of **ML** techniques to **WSNs** for anomaly detection, specifically in the context of wildfire detection. It provides a comprehensive analysis of four **ML** models and their effectiveness in detecting fire-related anomalies in WSN data, with an emphasis on real-world scenarios. The structure of this chapter is as follows:

- Section 5.1 – provides a general contextualization of the application of **ML** to **WSNs**. The section highlights the integration and the challenges of **ML** algorithms with **WSNs** to improve their performance in real-time monitoring, especially in dynamic environments.
- Section 5.2 – reviews related works on applying **ML** to **WSN** data, focusing on studies in anomaly detection and fire monitoring. It identifies common methodologies, challenges in model performance, and the gaps in applying low-cost sensors for wildfire detection.
- Section 5.3 – presents the datasets used for training and validation, detailing the conditions under which the data was collected, the features used in the analysis, and the importance of the datasets in testing the models' performance under both controlled and real-world wildfire scenarios.
- Section 5.4 – performs an **Exploratory Data Analysis (EDA)** on the datasets used, focusing on the distribution of variables, missing data, correlations, and initial insights. The analysis provides a foundation for understanding the data characteristics and informs the selection of appropriate **ML** models for anomaly detection.
- Section 5.5 – details the preprocessing steps applied to the datasets, including a comparison between versions of modules, developing a strategy for filtering data, and transforming raw data into formats suitable for **ML** model training and evaluation.
- Section 5.6 – discusses the different **ML** models implemented in this work, including **RWR**, **ARIMA**, **RNN**, and **TimesFM**, explaining their underlying principles, strengths, and limitations, as well as their adaptation to **WSN** data for fire detection.

- Section 5.7 – explores the benefits of combining the **RWR**, **ARIMA**, **RNN**, and **TimesFM**, for anomaly detection in **WSN** data through unweighted and weighted aggregation strategies to improve detection accuracy in fire scenarios.
- Section 5.8 – evaluates the performance of various **ML** models in identifying fire-related anomalies across multiple datasets, including controlled fire events in low and high **Tree Cover Density (TCD)** environments, and assesses their effectiveness through real-world testing.
- Section 5.9 – provides a comprehensive conclusion to the chapter, summarizing the effectiveness of the **ML** models applied to **WSN** data for fire detection.

5.1 General Contextualization of Machine Learning Applications for WSN

WSNs represent one of the most significant innovations in the **IoT** era. Composed of several distributed sensors, these networks are commonly designed to monitor environments, collect data, and transmit information to a central station [207]. Their versatility allows applications ranging from environmental monitoring [208] to industrial automation [209], health care [210], and water pipeline leakage [211]. However, large volumes of data are obtained as **WSNs** become denser, that is, with a high number of elements and/or variables collected [212].

ML, a sub-area of **AI**, involves the development of algorithms capable of learning from historical data, allowing predictions or decisions to be made without explicit (re)programming for each task [213]. In other words, based on historical data, it is possible to continually improve algorithms' performance in specific tasks based on the acquired experience. Instead of following predefined rules or thresholds, which may not provide the best solutions because they do not consider history, these algorithms can also identify patterns and trends in the data [214]. With this information, it is possible to make predictions to support decision-making.

One of the most common **ML** applications in **WSNs** is anomaly and event detection [215, 216, 217]. Anomaly detection involves identifying significant deviations from standard operating patterns, indicating potential problems such as equipment failures, leaks, or invasions [218, 219]. Anomalies in environmental data can indicate unusual events, such as forest fires, one of the areas of interest for forest monitoring. In this sense, **ML** algorithms can be trained to detect anomalies in **WSN** data, in the historical data of each module and sensor parameter, and to identify patterns that precede the occurrence of fires. Therefore, with this information, it is possible not only to detect fires early but also to predict their occurrence, allowing the implementation of preventive measures.

On the other hand, applying **ML** to **WSN** systems also implies significant challenges. One of the main challenges is to ensure the quality of the data collected since the presence of outliers and incomplete data can compromise the effectiveness of **ML** models [220]. To overcome these issues, it is essential to perform adequate pre-processing of the data, which includes steps such as noise cleaning and the treatment of missing or anomalous data. One challenge is selecting and evaluating

the most appropriate **ML** models and their hyperparameters, considering the nature of the data and the objectives of the monitoring system [208, 214]. The choice of models should be made based on performance metrics such as accuracy and precision, which help assess the effectiveness of the models in identifying relevant patterns.

Many **WSN** applications require time series analysis, given the continuous monitoring of variables over time. **ML** algorithms, such as **RNNs** and **Long Short-Term Memory (LSTM)** networks, are particularly effective in analysing temporal data. These algorithms can learn complex, non-linear patterns in data, allowing for accurate predictions of future measurements, which is particularly useful in weather forecasts and early detection of wildfires [221].

Another crucial factor is data fusion and dimensionality reduction of the dataset [215]. **WSNs** often collect data from multiple sensors, which can be heterogeneous and redundant. Data fusion and dimensionality reduction techniques can be applied to combine information from different **WSN** elements, improving the information quality and reducing the data volume to be processed [222]. In this way, it is possible to minimize the processing time of **ML** algorithms while maintaining the essential data information and, consequently, without losing the dataset's quality. Furthermore, reducing processing time directs **ML** algorithms to provide outputs with a time interval at least shorter than the sampling of **WSN** elements. Otherwise, real-time data sent by the **WSN** may remain in a type of queue to be analysed, which could be problematic in forest fire situations where information is expected to be obtained as quickly as possible [223].

Given these specific challenges for the **ML** application in **WSN** systems, it is essential to establish research objectives that explore practical and innovative solutions to optimize data processing, ensure system scalability, and improve the detection of critical events, such as forest fires. The research objectives should guide the development and implementation of **ML** models capable of attending to the time and data processing limitations of the **WSN** formed by the 5FTH modules. In this sense, it was decided to divide the research into objectives, which are detailed in the following topics.

Identification and Evaluation of Effective ML Methods for Anomaly Detection in WSNs

One of the main challenges in applying **ML** to **WSNs** is the accurate detection of anomalies, such as atypical variations in temperature, humidity, and **IR** intensity, which may indicate the occurrence of forest fires. This is because, during data collection, anomalies in the data may represent the occurrence of fires, and therefore, it is necessary to remove outliers to avoid discarding useful information. Through a detailed review of the literature to identify successful strategies employed by other related works, it is possible to find solutions or adapt methods to overcome this challenge. Therefore, the first objective is to identify and evaluate the most effective **ML** methods for detecting anomalies in environmental data. This analysis will include a critical review of the literature and the practical evaluation of different algorithms in simulation scenarios and field tests.

Processing and Exploratory Data Analysis

Before applying the **ML** models indicated by the literature review, it is essential to understand the data collected by **WSNs** in depth. The second objective is to perform a preliminary assessment using traditional statistical techniques to verify the feasibility of fire detection based on the monitored variables. In addition, it is necessary to perform a detailed **EDA**, focusing on identifying patterns and relationships between each variable. **EDA** will serve as a basis for developing and tuning **ML** models, helping to determine which data features are most relevant for fire detection. With this objective, **ML** models can be trained with consistent data, improving their performance.

Validation and Scalability of ML Models in Real Conditions

To ensure that **ML** models effectively detect fires under real-world conditions, they must be validated and adjusted using data from supervised fire tests. In this sense, the third objective is to evaluate the performance of **ML** models in detecting anomalies and predicting fires, using datasets collected from a **WSN** installed in forest areas with and without supervised fires. This validation process is dedicated to verifying how the models respond to different fire scenarios, as well as optimizing their hyperparameters to improve their accuracy and response time. Therefore, the definition of the **ML** models to be applied must consider the ability to operate in real-time and the integration with all system components without compromising the execution of the other elements. Thus, **ML** models must be compatible and implemented through microservices.

5.2 Related Works: Machine Learning applied in WSN data

Data collection in **WSNs** is a critical aspect of modern sensor technologies, facilitating the acquisition and transmission of data across several applications, from environmental monitoring to healthcare. **WSNs** consist of distributed sensor nodes that gather data on diverse parameters, such as temperature and humidity. The significance of data collection in **WSNs** emerges from their ability to provide real-time information and enhance operational efficiencies across sectors, underscoring their growing importance in the **IoT** ecosystem [224, 225].

The methodologies employed for data collection in **WSNs** include multiple techniques, encompassing techniques from data dissemination to data gathering. Data dissemination involves efficiently distributing information or commands throughout the network, while data gathering focuses on the data transmission to the base station, focusing on accuracy and minimal latency [226]. In this sense, ongoing debates over the trade-offs between energy efficiency and data integrity mark the discourse surrounding data collection in **WSNs**. Multiple strategies, such as clustering, duty cycling, and network coding, are actively explored to optimize performance and extend the network's lifespan [225]. Additionally, ethical considerations concerning data privacy and security are paramount, especially as **WSNs** increasingly monitor sensitive areas, requiring robust frameworks to protect individual data privacy while ensuring information accuracy [227].

Despite their advantages, **WSNs** face notable challenges beyond energy consumption and network reliability. Complexities such as hidden terminal problems and real-time data processing consistently challenge **WSN** methodologies [225, 226]. In the context of data processing, emerging trends in data collection within **WSNs** promise to harness advancements in edge computing, **AI**, and collaborative sensor networks. These advancements promise to create more intelligent and resilient systems capable of addressing the increasing demands of data-intensive applications [228, 229]. As these technologies evolve, the landscape of data collection and description in **WSNs** will continue to adapt, presenting both opportunities and challenges for efficient data management.

However, as real-time sensor data becomes more complex, managing and interpreting these datasets demands advanced techniques. Methods ranging from classical approaches, such as **Linear Regression (LR)**, to advanced **ML** models are essential for real-time analysis [230]. These techniques enable systems to differentiate between typical fluctuations and significant events, such as environmental hazards, ensuring timely and accurate responses.

Accurate data interpretation is crucial in **WSN** approaches, making it essential to establish and implement best practices for data documentation and analysis. Poor data management and analysis practices can lead to inaccurate predictions and missed anomalies, undermining the potential benefits of **WSN** technology. Therefore, the following subsections present a comprehensive review of the related work in this domain. By exploring best practices for **WSN** data documentation, effective data collection techniques, and data structuring methods, and the application of **EDA** and **ML** models to enhance the accuracy and reliability of sensor data analysis.

5.2.1 Data Structure and Description in WSNs

The related works support some directions on effective strategies for documenting sensor configurations, data collection protocols, and metadata standards that enhance data quality and usability. This area of focus has gained prominence due to the increasing reliance on **WSNs** in diverse industries, which necessitates comprehensive and user-friendly documentation to attend to diverse stakeholders, including developers, system administrators, and end-users.

The challenges in data collection in **WSN** are addressed in [231], which highlights sensors' high energy consumption and bandwidth limitations. To mitigate these problems, a three-layer framework is proposed that uses mobile elements, such as vehicles, to obtain data collection by visiting strategically selected collection points. Fixed sensors in a planar area send data to the cloud via multi-hop transmissions, minimizing latency and maximizing throughput.

The data collection in **WSN** described by [232] shows the issue of unstable renewable energy sources installed in modules to be self-sufficient. To test the proposed approach, the work implemented the data collection in adverse scenarios. The sensor network model is structured as a graph, where sensors can harvest energy from the environment, and a time-slotted system is adopted for all sensors to report data periodically to the sink. The sensors are equipped with solar panels, and the collected energy is stored in batteries, with energy costs defined for sensing, transmission, and reception operations. Through the detailed description of the dataset and the modeled scenario, it was possible to compare the performance of the proposed solution with other approaches, such as

the minimum sparsity tree, and validate that the proposed algorithm can better distribute energy consumption, reducing the chance of sensor battery depletion, especially in situations of limited energy.

Another work that also details the description of **WSN** applications in real environments to verify an approach is presented by [233]. In this paper, the Packet-in-Packet technique is explored to optimize **WSN** data collection, focusing on temperature monitoring and other critical measures. Using radio frequency transmissions with constructive interference and the capture effect, it was possible to improve energy efficiency and reduce **WSN** latency. The collected data is structured as short packets, storing measurements such as temperature and binary states. Real-world tests validated the approach across different node densities, assessing reliability and energy efficiency.

Other works that emphasize the importance of detailing the scenarios in which a **WSN** is applied, whether to obtain data collection, aggregation, and dissemination, are [234, 235, 236]. Both works highlight that documentation and appropriate data structuring are essential to facilitate data analysis and network performance. In addition, it is discussed that aggregation techniques can be used to combine and process data collected from multiple sensors before sending them to the central node or sink. Therefore, the main objective of these techniques is to reduce the amount of data that needs to be transmitted, saving energy, minimizing network overhead, and increasing communication efficiency.

The implementation of an **IoT**-based forest fire detection system using **WSN** is described in [237]. The prototype system was tested in a natural forest environment, where six sensor nodes were fixed to trees, and a central node sending data to a server located 2.6km away. However, to avoid real fire risks, the tests were performed using smoke spray, and the sudden increase in temperature was performed by holding a lighter close to the nodes' thermistor. Furthermore, the distance of these disturbances to the modules is not explicitly described. Despite the description of the data-sending interval, there is also no mention of when the fire event simulations were performed.

A real-time fire detection system, named Fire Detection and Prediction Architecture, which uses **WSN**, is described in [238]. The system was tested in three different locations, with sensors deployed to collect data such as temperature and humidity. The data was processed every 30s to detect the presence of fires. Although the tests were conducted in natural forest environments, the paper notes that it was impossible to test with propagating burns for safety and regulatory reasons. The sensors were positioned in locations with different weather conditions to validate the system, detailing the number of modules, the distances between them, and some information about the duration of the fire.

In [239], it is mentioned the implementation of **WSN** in forest areas to collect real-time meteorological data, with sensors deployed at high density in areas of higher risk (every 100m) and lower density in areas of reduced risk (every 500m). The work focused on collecting and analysing environmental data, such as temperature, humidity, wind speed, and precipitation, to predict fire risks. However, there is no description of tests carried out with real fire or tests with supervised fire.

5.2.2 Exploratory Data Analysis in WSN Data

EDA in WSNs encompasses a range of techniques that extract meaningful information or simplify complex data from the vast amounts generated by sensor nodes deployed in diverse environments. The significance of EDA lies in its ability to uncover patterns, relationships, and potential anomalies within complex datasets, thereby enhancing decision-making and operational efficiency [225]. EDA techniques are classified into univariate and multivariate analyses:

- **Univariate Analysis:** Focuses on a single variable within the dataset, helping to summarize and understand its distribution and characteristics. Descriptive statistics summarize data using metrics such as mean, median, and standard deviation. Frequency distributions use visual tools such as histograms and bar plots to show how often each unique value or category occurs within the variable. Outlier detection techniques, such as boxplots and scatter plots, also help identify extreme or unusual values that deviate significantly from the dataset's overall trends [240].
- **Multivariate Analysis:** Examines the relationships between multiple variables, which is crucial for understanding the interaction within WSN data. Correlation analysis assesses the strength and direction of relationships between pairs of variables using correlation coefficients, such as Pearson's or Spearman's correlation. Visual tools, including scatter plots and pair plots, are used to identify potential patterns, aiding in exploring their relationships. Furthermore, dimensionality reduction methods help visualize high-dimensional data in lower-dimensional spaces, making it easier to uncover underlying structures [240].

Furthermore, effective EDA relies heavily on visualization tools and methodologies that assist analysts in interpreting the data accurately and efficiently [241]. Despite the benefits, conducting EDA on WSN data presents several challenges. Data quality issues, such as inaccuracies and missing values, can significantly undermine the reliability of analyses, necessitating robust preprocessing techniques [242]. EDA functions as a critical component in harnessing the potential of WSN data, enabling in-depth understanding and supporting informed decision-making across a myriad of applications. As technology evolves, the continuous adaptation of EDA techniques will be essential to address emerging challenges and take advantage of advancements in sensor technologies and data analytics methodologies.

Data from the WSN, which continuously transmits real-time monitoring data to track landslides, are interpreted using EDA techniques in [243]. The applied EDA includes using boxplots to detect outliers in GPS displacement data, which helps distinguish between anomalies caused by measurement errors and actual ground deformations. In addition, methods such as polynomial fitting and moving average for noise reduction are compared to smooth the data, with the moving average being preferred for its accuracy in retaining important information.

To understand complex environmental and industrial data patterns, [244] performs a detailed EDA. Techniques applied include descriptive statistics, such as mean, median, and standard deviation, to summarize and assess the overall data distribution and scatter plots to explore relationships

between continuous variables, such as temperature and pollutant levels. Boxplots are used to detect outliers that may indicate anomalies or measurement errors, while frequency histograms help visualize the data distribution by identifying possible asymmetries. In addition, temporal analysis examines the variation of the data over time, revealing seasonal trends and long-term effects, essential for effective environmental monitoring.

A plant disease prediction system using data collected by WSN uses EDA in its dataset [245]. The applied EDA includes statistical analysis and data visualization with the Matplotlib and Seaborn libraries, which generate correlation graphs to identify significant relationships between variables such as temperature, soil moisture, and relative humidity, helping to understand important patterns. Data preprocessing is another essential step, where the data is cleaned and transformed, removing missing values and preparing it for analysis and modeling. In addition, heat maps highlight the most influential variables in the emergence of diseases, helping select relevant attributes for predictive models.

In [246], the application of EDA techniques using computational methods to analyse and monitor environmental data is described. One of the main approaches is averaging the collected raw data stored in a matrix, facilitating the initial analysis. In addition, using derivative analysis in sliding windows allows for identifying significant variations in the data, highlighting relevant changes or patterns. These derivatives are compared with a predefined threshold, and when they consistently exceed this threshold, the system is dynamically adjusted to optimize performance. In this way, the paper illustrates how these EDA techniques can be used to adjust and improve the system response in real-time.

EDA techniques, in [247], are implemented to ensure that the selected variables are relevant and that trends and patterns are correctly visualized, helping in data-driven decision-making. Therefore, processing and understanding data related to forest fires is possible. The techniques include correlation analysis to identify significant relationships between meteorological variables, such as temperature and humidity, which are essential for fire prediction. The paper also uses the transformation of categorical data into numerical data to facilitate analysis and ensure compatibility with predictive models, in addition to methods for identifying hotspots from meteorological data and analysing regions with higher fire risk.

To process weather data collected by WSN, EDA techniques are applied to guide essential directions to follow [248]. The methods include data preprocessing, where missing values and outliers are treated to ensure that the data is ready for analysis. The outlier treatment method uses the Interquartile Range (IQR) to identify and replace extreme values that may distort the analysis. In addition, correlation analysis is performed to understand the relationships between variables such as temperature, humidity, light, and voltage. The paper also uses the linearity test, normality of residuals with QQ plots, and checks for homoscedasticity to ensure that the regression model's assumptions are accomplished.

In [249], the EDA techniques applied include data cleaning, which involves removing incorrect, redundant, and null values, ensuring that the data is suitable for analysis. Anomaly treatment is done through validation against known ratios, and data normalization is used to ensure that all

variables are on the same scale, which is crucial for the stability and performance of the models to be applied. The paper also details data transformation, where numerical values are preferred to facilitate subsequent analysis with deep learning algorithms.

In order to predict wildfire risk, [239] indicates the implementation of **EDA** techniques to analyse meteorological data collected from **WSN**. **EDA** techniques include data cleaning, where anomalies and missing values are treated, such as data outside the expected range of temperature and humidity, and replacing null values with identifiers to ensure the integrity of the dataset. The importance of detecting “geographic outliers”, which refer to deviations from expected climate parameters, such as temperature and humidity, in certain regions, is emphasized. These outliers are considered anomalous if the meteorological values fall outside a reasonable range for the target area, such as temperatures or humidity that do not have a typical weather pattern. This analysis ensures that the climate data remains within reasonable ranges for the region under study. According to [239], after the entire **EDA** process, data analysis became more practical and understandable, which increased the accuracy of wildfire prediction models.

5.2.3 Machine Learning Models for WSN Data Analysis

ML models have become indispensable for analysing the complex data generated by **WSNs**. These networks often operate under constraints such as limited resources (e.g., low power or bandwidth), dynamic network topology, and vulnerabilities to attacks. Additionally, the data acquisition process in **WSNs** can be challenging, especially in diverse environmental scenarios or when the network density complicates data interpretation. Given these challenges, this subsection reviews different **ML** models applied to **WSN** data analysis, ranging from simple regression techniques to more advanced methods, such as Foundation Models.

The **ML** process applied to the dataset in [250], by the paper [251], used several techniques focused on detecting events, such as forest fires. The work compares **Decision Tree (DT)**, **LR**, **General Regression Neural Network (GRNN)** and **Support Vector Machine (SVM)**. Therefore, this comparison analysed the models’ performance on environmental data collected by sensors, including temperature, humidity, wind, and rain. To measure the accuracy of the models, the R^2 and **Root Mean Squared Error (RMSE)** metrics were used, with **RMSE** standing out for being more sensitive to significant errors. The R^2 metric was used to evaluate the variability explained by the model. According to the results obtained from the paper, **LR** obtained low processing time, low **RMSE** value, and low R^2 , indicating that this technique can detect forest fires when environmental data are used.

In [252], several **ML** models are mentioned as possible implementations to indicate wildfire occurrence and risk of fires. Among the models mentioned, the main ones are:

- **Artificial Neural Networks (ANN)s**: Used extensively, including **Multilayer Perceptron (MLP)**, **Back Propagation Network (BPN)**, **Radial Base Function Network (RBFN)**, and **Dynamic Learning Vector Quantization (DLVQ)**. These networks have been applied to classify and process multi-sensor data to improve accuracy and reduce false alarms [252].

- **Fuzzy Logic:** Implemented for the fusion of data from multiple sensors, increasing reliability and minimizing communication activity and false alarms. This was done to integrate and process variables such as temperature, smoke density, CO, and humidity [252].
- **DT:** Used in comparative studies for wildfire detection, evaluating classifiers such as OneR and comparing their effectiveness on different datasets [252].
- **Naive Bayes:** Implemented for fire detection in multi-sensor WSNs, showing accuracy of up to 94% in some cases [252].
- **SVM and Bayesian Models:** These are used less frequently but with significant results regarding true detection rate. However, SVM has shown limitations for large fires [252].
- **Deep Neural Network (DNN)s and CNNs:** Explored for advanced fire modeling with reasonable accuracy rates, especially in WSN-based systems for fire detection [221, 252].

Additionally, in [252], the paper extends to metrics mentioned for evaluating each model. The RMSE is used to measure the mean squared error of model predictions and has been applied to evaluate the performance of MLP, RBFN, and SVM. Similarly, the Mean Absolute Error (MAE) is reported alongside RMSE to indicate the mean absolute error of predictions, with MLP and SVM being the focus of evaluation. The Mean Absolute Percentage Error (MAPE) measures the average percentage accuracy of models and has been used for MLP, RBFN, and SVM. For overall performance evaluation, the Mean Absolute Deviation (MAD) is also employed in several studies. Metrics such as Accuracy, Precision, Recall, and F-score are widely reported, particularly for neural networks. False Alarm Reduction metrics is implemented through image processing techniques and neural networks to minimize false alarms. Finally, the Area Under the Curve (AUC) metric is used to assess the quality of classifiers and has been applied across different models depending on the study.

A system described in [221] uses ML to analyse WSN data focusing on early detection of forest fires. The system was composed of distributed sensors that collected environmental data, such as temperature, relative humidity, atmospheric pressure, concentration of CO₂, CO, and amount of particulate matter. These data were then processed by deep learning models, such as RNNs, LSTMs, and Gated Recurrent Unit (GRU)s, configured with adjusted parameters to optimize prediction accuracy. The parameters used to evaluate the effectiveness of the models were Loss Function and Accuracy, in which the GRU model performed better than the LSTM and RNN models.

Temperature and Humidity variables, collected from a WSN designed to monitor forests, were analysed using a density-based ML scheme to detect anomalies in [253]. The approach focuses on outlier detection, where anomalous data is identified using clustering techniques. In this regard, K-Means and Mean Shift methods were initially considered, but the paper introduces a Density-Based Scheme (DBS). DBS creates high-density clusters, identifying outliers in low-density regions. Anomaly detection is performed by identifying data that deviates significantly from the standard. This includes incorrect data caused by faulty sensors, extreme weather conditions, or malicious

activities. The metrics used were Precision (proportion of true outliers correctly detected) and Accuracy (a measure of the system's overall accuracy in identifying normal and anomalous data).

Using multiple LR techniques, [254] discussed an ML-based approach to predict fire situations. Data was collected from sensors in different weather conditions, with 7000 samples divided into 80% for training and 20% for testing. The model was trained on a dataset of environmental variables such as temperature, relative humidity, light intensity, and CO levels. The system also applied threshold ratio analysis to detect when a fire situation was forming quickly. In addition, anomaly detection was performed using clustering techniques such as K-means to separate the data into clusters of fire and non-fire situations. The anomalies identified included readings outside the expected patterns, which could indicate sensor failures or atypical situations caused by fires. The approach achieved a theoretical accuracy of 81%, and the accuracy was calculated in field tests that included both fire and non-fire scenarios.

A detailed study for anomaly detection using the Isolation Forest (IForest) algorithm is presented in [255]. The main objective is to analyse high-dimensional data and real-time streams. The paper introduces the algorithm with an improved version, Extended Isolation Forest (EIF), and a new extension called Majority Voting Isolation Forest (MVIForest), which uses majority decisions among individual trees, reducing execution time without compromising accuracy. Metrics such as Receiver Operating Characteristic (ROC), precision, recall, F1-Score, and false alarm rate were used to evaluate the model, highlighting its robustness and efficiency. The method indicated scalability and applicability to large datasets, detecting anomalies based on their ease of isolation. However, detection in multiple dimensions presented challenges due to the density and proximity of actual data.

Using an open-access WSN dataset, in [256] is performed comparative analyses of four classification models: SVM, DT, Logistic Regression, and MLP. The dataset comprises sensor data measuring temperature, humidity, light intensity, and CO levels. The MLP model was developed with four hidden layers, each containing ten neurons, and the Rectified Linear Unit (ReLU) activation function. Training was done using 80% of the data (50104 samples), while 20% (12526 samples) were reserved for testing. The effectiveness of the model was validated using ROC curves, which indicated an accuracy of 99.7% after 10-way cross-validation.

A Cognitive Fire Alert Framework has been proposed for wildfire detection applying WSN [257]. This system collects environmental data, such as temperature, humidity, and pressure, and transmits it via ZigBee to a base station. The collected data is then analysed using the Random Forest (RF) method to classify fire events with high accuracy, reaching up to 99% accuracy. The framework also performs threshold analysis, comparing measured values with a continuously updated reference value. Anomaly detection is incorporated to quickly identify unexpected variations, such as erroneous sensor readings or extreme environmental conditions, thereby reducing false positives.

While these classification and anomaly detection methods have proven effective in many WSN applications, understanding and predicting the temporal patterns in sensor data requires more specialized approaches. In particular, time series analysis has emerged as a crucial aspect of WSN data analysis, enabling proactive decision-making in dynamic environments. Advanced time series

forecasting models address the challenge of capturing both the temporal dependencies and the native noise in WSN data.

Several innovative techniques and libraries have been developed to improve the accuracy of time series forecasting. For instance, the AIX360 library discussed in [258] combines deep learning with traditional statistical methods to enhance predictions in industrial settings. This library offers interpretable models tailored for time series analysis, including Time Series - Time Series Local Interpretable Model-agnostic Explanations (TS-LIME), Time Series Individual Conditional Expectation (TS-ICE), and Time Series SHapley Additive exPlanations (TS-SHAP). These techniques improve forecast accuracy and provide explainability, which is essential for understanding the underlying factors influencing predictions.

Furthermore, with the aim of improving the accuracy of short-term power forecasting and ensuring the safe and economical operation of power systems, [259] presents a short-term power forecasting model for centralized Photovoltaic plants. The dataset covers a specific period, with power measurements and weather forecasts every 15 min collected from a centralized Photovoltaic plant. The Long Short-Term Network with Attention (LSTNet-Attn) model is composed of several layers: the first uses a CNN to extract short-term dependencies; the second layer captures the long-term trends of time series through Long-term and LSTM, and a recurrent-skip structure is used to learn ultra-long-term repetitive patterns, with a fully connected layer and an Autoregressive (AR) model for combined forecasting.

Another example, in [260], is presented an innovative system for recognizing, diagnosing, and predicting anomalies in a large volume of time series data from energy automation systems. The system uses a combination of Time-series Generative Adversarial Networks (Time-GAN) and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) to identify, diagnose, and predict data anomalies in real-time, increasing the accuracy and efficiency of detection in power grids. Time-GAN generates large synthetic datasets from limited samples by balancing the distribution of different data types, while DBSCAN detects unknown anomalies based on the spatial density of the data. Dataset includes telemetry information and control messages, ensuring the accuracy of the analysis. The model was evaluated using metrics such as precision, recall, and F1-score, which obtained high effectiveness in detecting anomalies and surpassing traditional methods.

Although time series analysis with a single variable provides information about the behavior of a given phenomenon over time, analyses with multivariate approaches can offer a more robust and comprehensive perspective. analysing multivariate time series allows for evidentiatio of the interdependencies and correlations between multiple variables, providing a more complete and detailed understanding of complex systems. Including multiple variables can significantly improve forecast accuracy, identify hidden patterns, and reveal underlying dynamics that are not apparent when considering only one variable. However, according to [261], it is necessary to perform compression in the temporal dimension to avoid the Information Bottleneck (IB). In this sense, applying the Multiple Time Series (MTS)-IB to transform time series into two-dimensional representations through partial convolutions is possible.

In [262], a Seq2Seq model with multiple parameters is proposed for the short-term forecast of seawater temperature data. The main objective is to improve the accuracy of sea surface temperature predictions by considering highly correlated external factors. The work adapts the Seq2Seq model, commonly used in **Natural Language Processing (NLP)**, to achieve this objective for time series prediction. The model's effectiveness was compared with other models, such as **LSTM** and **GRU**, indicating superiority in predicting multivariate time series. The dataset was collected over a decade and includes variables such as temperature, pressure, humidity, wind speed and direction, wave height, and wave period. The model's effectiveness was evaluated using metrics such as **RMSE**, which showed an improvement in forecast accuracy when highly correlated variables were included.

Using a detailed dataset from 23 years of daily observations, [263] performs a comparison between the effectiveness of statistical models, XGBOOST and Transformers in predicting malaria incidence based on high-resolution climate data. The Transformers' superiority in several performance metrics is highlighted. An innovative loss function, named M-Delta, was developed to optimize model accuracy, resulting in an approximately 20% improvement over standard **Mean Squared Error (MSE)** loss. The Transformer model achieved high effectiveness and robustness with 98% accuracy in daily forecasts and a **ROC** of 0.83. In this way, Transformers can be promising tools for malaria prediction, offering significant improvements in surveillance and response to outbreaks in endemic regions.

A new time series forecasting paradigm using text-based prompts was introduced in [264]. Using pre-trained language models, PromptCast transforms numeric data into natural sentences to make predictions. The model was tested with the **Prompt based Time Series forecAasting (PISA)** dataset, covering temperature prediction, energy consumption, and customer flow. The PromptCast architecture employs encoders and decoders to convert numeric sequences into sentences and generate textual predictions. Evaluated with metrics such as **RMSE** and **MAE**, the model obtained accuracy in multivariate and single-step forecasting scenarios and generalization in zero-shot settings.

Combining **ARIMA** models and stepwise regression analysis to predict the occurrence of wildfires, [265] indicates that through climate data collected from 1991 to 2021, including temperature, precipitation, relative humidity, and wind speed, it is possible to identify the most influential meteorological factors in the occurrence of fires. The stepwise regression model determines that average annual precipitation and wind speed are the main predictors of forest fires, while the **ARIMA** model analyses the trend of occurrences. The combination of these models presented significant predictive accuracy.

Pre-trained language models for time series forecasting, which transform numeric values into discrete tokens through scaling and quantization, are described in [266]. Chronos trains transformer-based architectures (i.e., *T5*) to predict time series using cross-entropy loss. The effectiveness of Chronos was obtained on 42 datasets, where it outperformed existing methods and showed comparable or superior performance on new datasets in zero-shot configurations. Using data augmentation strategies such as **Time Series Mixup (TSMixup)** and KernelSynth, Chronos improves model robustness and generalization, becoming a viable and efficient alternative to

traditional time series models and significantly simplifies forecasting pipelines.

Tiny Time Mixer (TTM) are presented in [267], a compact and efficient multivariate time series forecasting model. Based on the TSMixer architecture, **TTM** uses **MLP** blocks interleaved with simple gated attention, avoiding the complexity of Transformers. Trained on multiple public datasets, **TTM** employs techniques such as adaptive patching and resolution prefix tuning to handle different dataset resolutions. Evaluated on 11 datasets, **TTM** achieved superior predictive accuracy and computational efficiency performance. Furthermore, **TTM**'s ability to capture correlations between channels and integrate exogenous signals is highlighted, presenting it as an ideal solution for environments with limited resources.

In [268], the **Foundation Model (FM)** is described as an advanced tool for time series analysis, categorizing it into three main approaches: Transformer-based models, which use attention mechanisms to capture long-range dependencies in sequential data; Non-transformer models, such as **RNNs**, **MLPs**, and **CNNs**, which are effective in classification and prediction tasks; and Diffusion-based Models, which manage the uncertainty by predicting future states through inverse noise processes. These approaches are applicable in several areas, including transportation, environmental monitoring, and healthcare, with examples such as traffic forecasting and climate analysis. To maximize the effectiveness of **FMs**, the paper recommends supervised and self-supervised pre-training strategies, such as contrastive learning and synthetic data generation. Adaptation techniques, such as fine-tuning and zero-shot learning, are suggested to customize models for specific tasks.

The use of **FMs** and deep learning for **MTS** forecasting, classification, and anomaly detection is explored in [269]. The paper highlights the methods **Seasonal Autoregressive Integrated Moving Average Exogenous (SARIMAX)**, **N-BEATS**, **LightGBM**, and **Transformers**, including **Large Language Models (LLMs)**, which enable robust analysis in diverse contexts, such as pandemic, traffic, and economic forecasting. Multimodal models, which combine temporal and textual data, increase the accuracy and interpretability of forecasts, especially when combined with Knowledge Graphs.

Inspired by recent advances in **NLP** models, the **TimesFM** model was developed as a fundamental decoding model for time series forecasting [270]. The main objective is to provide highly accurate predictions for time series in zero-shot mode, i.e., without additional training for each new dataset. Leveraging a patch-based processing method, **TimesFM** tokenizes time series data similar to **NLP** tokenization, enabling efficient handling of multivariate and univariate data. It is pre-trained on a large corpus of 3 million synthetic time series with multiple patterns – such as trends, seasonality, and **ARMoving Average (MA)** processes – the model generalizes well to real-world datasets. Therefore, it is expected that **TimesFM** can forecast values regardless of the origin or characteristics of the data present in a given time series dataset.

5.2.4 Summary of Data Structure and Descriptions, Exploratory Data Analysis and Machine Learning Models

The reviewed literature highlights significant advancements and challenges in applying **WSN** data across different domains. Key contributions were discussed in the areas of data structuring, **EDA**, and **ML**, providing a comprehensive understanding of the methodologies employed and their impact on real-world scenarios.

Data Structure and Description in WSNs: Studies emphasize the importance of robust data documentation and structuring practices in ensuring the usability and reliability of **WSN** data. Techniques such as efficient data dissemination and aggregation methods have been widely explored. These methods enhance network performance, reduce latency, and address energy constraints. However, some works highlighted gaps in describing experimental setups, particularly in detailing sensor placement and data collection intervals during real-world deployments.

EDA in WSN Data: The implementation of **EDA** techniques underscores their importance in interpreting complex datasets generated by **WSNs**. Univariate and multivariate analyses, such as descriptive statistics, and correlation matrices, have been employed to identify patterns, anomalies, and relationships between variables. Despite advancements, challenges such as managing missing values, treating outliers, and ensuring data quality persist, particularly in large-scale deployments. Visualization tools remain a foundation of **EDA**, aiding in uncovering underlying trends and facilitating informed decision-making.

ML Models for WSN Data Analysis: The application of **ML** models reveals their capability to address the complexity and dynamic nature of **WSN** data. Traditional approaches, such as **LR** and **ARIMA**, provide foundational information, particularly for simple time series forecasting tasks. On the other hand, advanced models, including **RNN** and **TimesFM**, showcase their adaptability and accuracy in capturing temporal patterns and multivariate dependencies, making them well-suited for real-time fire detection and environmental monitoring. The most common metrics employed to evaluate the **ML** models were **MAPE**, **MAE**, **MSE**, and **RMSE**, can determine the models' effectiveness across different scenarios. Notably, in the related works, anomaly detection techniques, such as **IForest**, were pivotal in identifying deviations in predicted versus actual values.

The **LR**, **ARIMA**, **RNN**, and **TimesFM** models indicates their potential for forecasting in **WSN** data. In this sense, for this work, if these models effectively capture the expected behavior of each variable, they could predict future values before data collection occurs. By predicting future values, these models can enhance anomaly detection systems, such as **IForest**, by identifying deviations from expected patterns, thereby serving as an early warning mechanism for fire detection. To evaluate these models in a real-world scenario, experiments with controlled fire were conducted, aiming to explore supervised strategies for anomaly detection in **WSN** data.

5.3 Datasets

This section is dedicated to presenting a detailed test performed to generate the dataset of the modules with fires in real scenarios. The focus is on evaluating the performance of the different versions of the 5FTH module, specifically in terms of material resistance and the sensors' effectiveness in detecting fires in forest environments with different tree densities.

To test the 5FTH versions for identifying wildfires, controlled and supervised fires were used to evaluate the performance of the proposed WSN for forest monitoring. These tests can be conducted in scenarios where the biomass was already intended to be burned, either due to land clearing (a practice commonly performed by firefighters to preserve certain areas), using tree pruning material, or in a space provided by IPB. Each of these scenarios is described in the following subsections.

The tests with supervised and controlled fires used in this work are called **Wildfire Supervised Events (WSE)**. These tests are specifically planned to replicate an emergency wildfire ignition scenario from the module's point of view. In other words, the tests involve real fires, carefully supervised and controlled by firefighters. The goal of each WSE is to evaluate the behavior of the data in conditions as close as possible to unsupervised events (such as natural wildfires caused by lightning or friction between dry materials in nature).

Before carrying out tests on each WSE, it is essential to ensure that all safety measures are in place. In this sense, the first step is to choose a selected area free of flammable materials and provide all necessary firefighting equipment available and working correctly. Furthermore, it is also essential to guarantee that all individuals are at a safe distance from the fire. Additionally, the combustion during WSEs must closely resemble natural forest fire conditions. Therefore, only forest biomass was used during the tests, such as branches, twigs, trunks, and brushwood, among others. To assist in the preparation of WSEs and for safety reasons, a checklist was created and is available in [30]. Moreover, all datasets made during the WSEs performed in this work were authorized, controlled, and supervised by the firefighters.

To facilitate the reference and analysis of the data collected during the fire tests, each dataset will be assigned a unique identifier, followed by a number. For example, the first test scenario described is labeled as Dataset 1, with the series continuing to Dataset n . This labeling helps to distinguish between the datasets generated from each test and when discussing the application of ML models to these datasets. Moreover, within the datasets, the column indicating the observed fire event timing will be labeled as **Fire Status (FS)**. This column specifies the timestamps for when the fire was observed to start and end during the tests.

In the Table 5.1, an example is provided to illustrate how the data from these fire tests is organized and stored. Dataset structure includes the Node ID and Node Version, alongside several sensor readings such as Humidity, Temperature, and five IR sensor outputs, as well as an indicator specifying whether a FS was present during the data collection, timestamp, and Battery levels. This approach builds upon the structure already described in Section 3.4, adding elements specific to the fire tests. Therefore, only the new variables introduced for these tests are highlighted here. Each dataset follows this consistent structure, enabling reliable analysis and comparison across different

tests and conditions.

Table 5.1: Dataset's structure generated by a given module stored on the server associated to a WSEs.

Time [HH:mm]	Battery [V]	Humidity [%]	Temperature [°C]	IR1	IR2	IR3	IR4	IR5	FS	Node ID	Node Version
Day 1 00:01	3.82	75	10	42	45	42	55	40	False	04-5FTH	1.0
Day 1 00:02	3.82	75	10	44	41	44	56	42	False	04-5FTH	1.0
...	...	...	...	...	...	...	...	...	...	...	...
Day 5 12:02	3.63	56	19	46	45	46	64	46	True	04-5FTH	1.0
Day 5 12:03	3.63	51	19	40	38	39	47	39	True	04-5FTH	1.0
...	...	...	...	...	...	...	...	...	...	...	...
Day 20 23:58	3.48	70	17	1023	1023	1023	1023	1023	False	04-5FTH	1.0
Day 20 23:59	3.48	71	17	1023	1023	1023	1022	1023	False	04-5FTH	1.0

In fire tests, four different scenarios are proposed to evaluate the performance and behavior of modules in fire conditions under different forest density conditions. These scenarios are designed to replicate real fire situations, ranging from areas with sparse vegetation to regions of high forest density. The objective is to verify the accuracy and speed of the sensors in detecting and reporting significant variations in temperature, humidity, and intensity of **IR**, which are critical elements in preventing and fighting forest fires. Each of these scenarios contributes to understanding the module's ability to operate effectively under extreme conditions, identifying the pros and cons of **WSN**.

5.3.1 Testing WSN at Low TCD Region - Dataset 1

The purpose of the first scenario is to apply a **WSE** in regions with low **TCD**, that is, small numbers of trees per square meter. The **TCD** (Tree Cover Density) is a value ranging from 0 to 100 representing the proportion of an area covered by trees, with 0 indicating no tree coverage and 100 indicating full tree coverage. This scenario was provided by **IPB**, and according to the Tree Cover Density 2018 (raster 10 m) map [56], the **TCD** for this location is 0. This type of region is characterized by receiving more intensity of sunlight since there is not much coverage of trees for the production of shadows. This effect can cause more noise when reading data from 5FTH modules. Figure 5.1 shows the placement of 5 modules in the first scenario. For this scenario, all sensor modules' versions used were 5FTH 1.0.



Figure 5.1: Placement of five sensor modules in the first scenario with low TCD.

The central part of this scenario is the biomass fuel that was burned to analyse WSN's behavior. This biomass has an ellipse shape with the following dimensions: $4.50\text{ m} \times 3.0\text{ m} \times 1.0\text{ m}$. In front of this central element are the five modules: 02-5FTH, 08-5FTH and 03-5FTH are about 5.50 m of biomass fuel (straight line); 09-5FTH and 05-5FTH are installed at 4.0 m and 6.0 m (straight line), respectively. All modules are at the height of about 1.50 m from the ground. Figure 5.2 shows the beginning of biomass burning, starting at 9h30 and ending at 10h50.



Figure 5.2: Five sensor modules acquire data of fire ignition in the first scenario with low TCD.

The five modules were fixed in the day before the test, ensuring stability and consistent data collection. The dataset for this scenario begins February 22, 2021, 15 h25 UTC, and ends February 25, 2021, 15 h24 UTC. The details of the module placements, along with the timing of the WSE, are summarized in Table 5.2.

Table 5.2: Modules placement details and fire observation timing for Dataset 1.

Node ID	Height [m]	Distance to Fire [m]	Dataset Start (UTC)	WSE		Dataset End (UTC)
				Fire Start (UTC)	Fire End (UTC)	
2-5FTH	1.5	5.5	February 22, 2021, 15:25	February 23, 2021, 09:15	February 23, 2021, 10:50	February 25, 2021, 15:24
8-5FTH	1.5	5.5	February 22, 2021, 15:25	February 23, 2021, 09:15	February 23, 2021, 10:50	February 25, 2021, 15:24
3-5FTH	1.5	5.5	February 22, 2021, 15:25	February 23, 2021, 09:15	February 23, 2021, 10:50	February 25, 2021, 15:24
9-5FTH	1.5	4	February 22, 2021, 15:25	February 23, 2021, 09:15	February 23, 2021, 10:50	February 25, 2021, 15:24
5-5FTH	1.5	6	February 22, 2021, 15:25	February 23, 2021, 09:15	February 23, 2021, 10:50	February 25, 2021, 15:24

Figure 5.3 presents, in Grafana panel, the raw data received by 09-5FTH along the day of the WSE. It is possible to notice that the IR sensors of this module exhibited oscillations, but it is not visually possible to identify the fire by comparing the oscillation in the values during the WSE with the values of before or after. On the other hand, the humidity and temperature sensors have significant fluctuations during the test, being possible to observe that something happened after 9h30. The acquired values of humidity and temperature between 9h30 and 9h35 decreased and increased faster than usual, respectively.

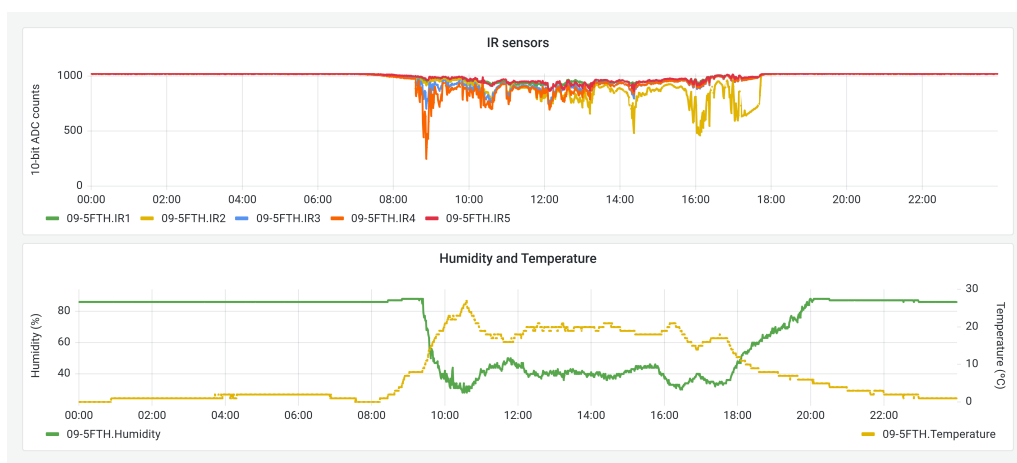


Figure 5.3: Data obtained by 09-5FTH during WSE in the first scenario.

The graphs of all the other four modules presented a similar behavior, and consequently, the graphs will not be shown.

For ease of reference, the data collected from this first scenario, which replicates a wildfire in a low TCD, will be referred to as Dataset 1. This naming convention will be used consistently in subsequent sections, particularly when discussing the application of ML models to these data. By clearly identifying this dataset, comparing and contrasting the results obtained from different fire tests and analysing the effectiveness of several ML models across different environmental conditions becomes easier.

5.3.2 Testing WSN at High TCD Region - Dataset 2

The scenario used for this test has a TCD value of 63 (on a scale from 0 to 100, where 0 represents no tree cover and 100 represents full tree cover), according to the Tree Cover Density 2018 (raster 10 m) map [56], as shown in Figure 5.4. Similar to the previous dataset, the WSEs used the same

low-cost modules (with the same configuration) to form a **WSN**. The scenario's central element remains the biomass fuel, which is equidistant from all modules (in a straight line concerning the ground), approximately 4.0m. The installation height of the modules depending on the number of modules installed on a single tree: 1.10m, 1.40m and 1.60m.

In the second scenario, in **IPB** facilities, the volume of the biomass was reduced due to safety measures, given the proximity to branches and tree trunks (some branches partially covered the upper fire location). Therefore, the biomass fuel has a circular shape with a diameter of 2.0m and a height no greater than 1.0m.



Figure 5.4: Placement of 10 sensor modules in the second scenario (with high TCD).

The test's duration also was reduced due to the smaller amount of biomass fuel used, ensuring the safety of the surrounding vegetation. Additionally, during the **WSE**, more biomass was not added to avoid compromising the sensors' readings. Increasing the biomass would have required more people to manage the fire, which could have obstructed the visibility of the **IR** sensors, leading to irregular oscillations in the readings. To mitigate this issue and ensure accurate data collection, it was decided to conduct additional **WSEs** at different hours of the day, including during the night (indicated by Figure 5.5).



Figure 5.5: Night-time test conducted as part of the additional WSEs to evaluate sensor performance under low-light conditions.

Both versions of the 5FTH modules described in the previous chapter were installed in this scenario to evaluate their performance under high TCD. In this sense, the differences in sensor configurations and hardware setups can be analysed to determine whether the upgrades offers better accuracy and reliability in detecting fire events. Figure 5.6 illustrates the distribution of both version modules. Additionally, the module 11-5FTH was positioned on Tree 1, module 14-5FTH on Tree 2, and module 16-5FTH on Tree 3. These last four module versions are 5FTH 2.0, allowing comparative analysis between different locations within the test environment.



Figure 5.6: The two versions of 5FTH modules being tested in the high TCD scenario.

The 5FTH modules were positioned at specified heights in different locations to assess their performance in detecting fire events. Data collection periods changed across modules, starting as early as May 4, 2021, 9h00 UTC and concluding by May 30, 2021, 10h03 UTC. The details of each module's placement, start and end readings, and the WSE count are presented in Table 5.3.

Table 5.3: Modules placement details, WSEs, and initial and ending time for Dataset 2.

Node ID	Height [m]	Dataset Start (UTC)	WSE	Dataset End (UTC)
01-5FTH	1.4	May 4, 2021, 09:00	1, 2, 3, 4	May 30, 2021, 10:03
02-5FTH	1.6	May 4, 2021, 09:00	1, 2, 3, 4	May 30, 2021, 09:55
03-5FTH	1.8	May 4, 2021, 09:00	1, 2, 3, 4	May 30, 2021, 10:02
04-5FTH	1.4	May 12, 2021, 07:21	1, 2, 3, 4	May 30, 2021, 10:02
05-5FTH	1.6	May 12, 2021, 07:19	1, 2, 3, 4	May 27, 2021, 20:22
06-5FTH	1.8	May 4, 2021, 09:00	1, 2, 3, 4	May 30, 2021, 10:02
07-5FTH	1.6	May 12, 2021, 07:30	1, 2, 3, 4	May 30, 2021, 10:02
08-5FTH	1.8	May 12, 2021, 07:16	1, 2, 3, 4	May 30, 2021, 10:02
09-5FTH	1.6	May 4, 2021, 09:00	1, 2, 3, 4	May 30, 2021, 10:02
10-5FTH	1.8	May 4, 2021, 09:00	1, 2, 3, 4	May 30, 2021, 10:02
11-5FTH	1.4	May 25, 2021, 16:15	4	May 30, 2021, 10:02
14-5FTH	1.4	May 25, 2021, 16:15	4	May 30, 2021, 10:02
16-5FTH	1.2	May 25, 2021, 16:15	4	May 30, 2021, 10:02

In total, four WSEs were conducted to assess the performance of the 5FTH modules under the second scenario, with each WSE labeled by its respective start and end times, as shown in Table 5.4.

Table 5.4: Detailed labels, and initial and ending time for WSEs in Dataset 2.

WSE	Fire Start (UTC)	Fire End (UTC)
1	May 12, 2021, 15:00	May 12, 2021, 15:45
2	May 19, 2021, 23:20	May 19, 2021, 23:40
3	May 20, 2021, 15:50	May 20, 2021, 16:10
4	May 25, 2021, 18:25	May 25, 2021, 18:50

To illustrate the data graphically, Figure 5.7 presents the sensor readings captured by Module 09-5FTH over several hours before and after the execution of the WSE 3. In this context, the fire test duration starts at 15h50 and ends at 16h10. A significant change in the data acquired by the module's sensors is noticed, with all IR sensor lines displaying an abrupt decline during the WSE period, and this pattern does not occur outside this timeframe. Additionally, data obtained for temperature and humidity show a behavior similar to the previous test – an significant increase in temperature and a corresponding decrease in humidity.

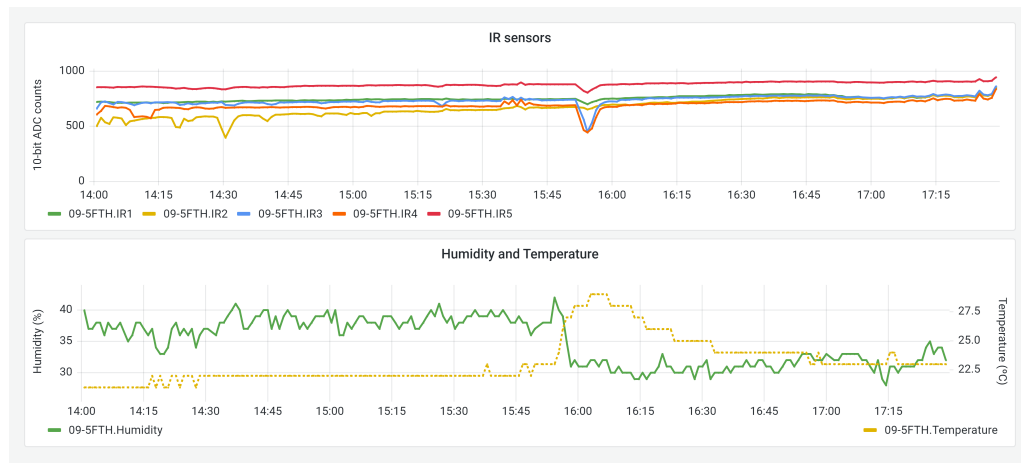


Figure 5.7: Data obtained by 09-5FTH during the third WSE in a high TCD scenario.

One of the WSEs was conducted at night to evaluate the performance of the modules under low-light conditions, with the test occurring between 23h20 and 23h40. Figure 5.8 illustrates the data collected during this night-time WSE. Notably, the IR sensors exhibited a distinct pattern with significantly reduced noise levels before and after the WSE 2. This lack of interference is due to the absence of sunlight, which can cause fluctuations in the IR sensor readings during daytime tests. Moreover, the acquired temperature and humidity data present an expressive variation during the WSE compared to the timeframe outside the WSE period.

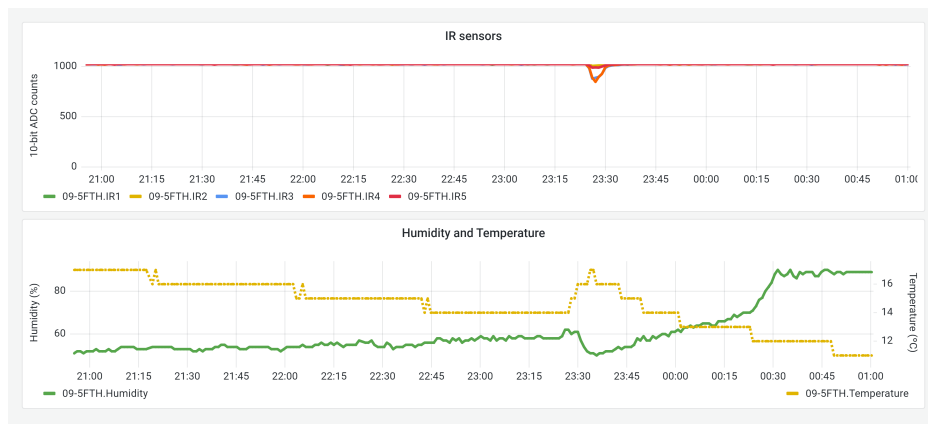


Figure 5.8: Data obtained by 09-5FTH during night-time (WSE2) in the second scenario.

The behavior observed in the remaining modules' graphs was similar, and these graphs are not included.

To facilitate reference, the data from this second test scenario – designed to replicate a wildfire under high **TCD** – will be identified as Dataset 2. This consistent naming convention will assist in the following sections, especially when examining the application of **ML** models to these data. By distinctly labeling this dataset, it is possible to compare the outcomes across multiple scenarios with **WSE** and evaluate the performance of multiple **ML** models under different environmental conditions.

5.3.3 Testing WSN at High TCD Region - Dataset 3

This test was performed in a high **TCD** scenario in the **IPB** campus, the same scenario mentioned in the last subsection. In contrast to previous datasets, this test was conducted during a month with high fire risk (according to [10]). Therefore, this test aims to create a dataset that not only reflects typical environmental conditions but also provides data patterns from a supervised fire under critical and high-risk scenarios. The **WSN** used for this dataset is shown in Figure 5.9



Figure 5.9: The scenario used for checking the ML models and strategies with new data.

The selected modules, labeled 12-5FTH, 14-5FTH, 17-5FTH, and 19-5FTH, were installed at the height of 1.40m above ground, referencing the tree trunk on which they were fixed. All modules are 5FTH 2.0. This data acquisition was conducted in September, a month typically associated with high temperatures and increased fire risk in the region [10]. Consequently, the data collected from this dataset are critical for verifying the behavior of the applied models and strategies and provide relevant seasonal characteristics due to the testing period. The details of the modules' placements and WSE timing for this dataset are summarized in Table 5.5.

Table 5.5: Modules placement details and WSE timing for Dataset 3.

Node ID	Height [m]	Dataset Start (UTC)	WSE		Dataset End (UTC)
			Fire Start (UTC)	Fire End (UTC)	
12-5FTH	1.4	September 14, 2021, 00:00	September 21, 2021, 11:20	September 21, 2021, 12:00	September 28, 2021, 23:59
14-5FTH	1.4	September 14, 2021, 00:00	September 21, 2021, 11:20	September 21, 2021, 12:00	September 28, 2021, 23:59
17-5FTH	1.4	September 14, 2021, 00:00	September 21, 2021, 11:20	September 21, 2021, 12:00	September 28, 2021, 23:59
19-5FTH	1.4	September 14, 2021, 00:00	September 21, 2021, 11:20	September 21, 2021, 12:00	September 28, 2021, 23:59

The data collected from this third test scenario – conducted during a period of heightened fire risk due to seasonal high temperatures – will be referred to as Dataset 3. Similar to the other datasets, this naming convention will assist in subsequent sections, particularly when discussing the evaluation of ML models on this data.

5.3.4 Testing WSN at Low TCD Region - Dataset 4

Dataset 4 was collected in a scenario with a TCD of 0, based on [56]. The primary objective of this test was to perform data acquisition during controlled and supervised fires by firefighters, conducted as part of land management practices to reduce the risk of large-scale wildfires. These controlled fires are essential for managing high biomass zones, which can fuel wildfires. Safely burning these areas under controlled conditions minimizes the potential for fire spread or intensity during wildfire. Figure 5.10 presents scenario during the test, showing the fire spread, the firefighters, and the modules.



Figure 5.10: The scenario with fire spread during data collection for Dataset 4.

This dataset provides an opportunity to observe and analyse the patterns in the **WSN** data during the fire spread over the terrain. The test involved placing four modules, labeled 12-5FTH, 14-5FTH, 17-5FTH, and 19-5FTH, along a predefined area. Module 12-5FTH was designated as the origin point (0 m), and the distances between subsequent modules were measured in a straight line relative to the preceding module. All modules were installed at a uniform height of 1.4 m, fixed securely using stakes, and positioned approximately 2 m from the fire path. Table 5.6 summarizes the placement details and timing of the **WSE** within Dataset 4.

Table 5.6: Modules placement details and WSE timing for Dataset 4.

Node ID	Distance [m]	Dataset Start (UTC)	WSE		Dataset End (UTC)
			Fire Start (UTC)	Fire End (UTC)	
12-5FTH	0	January 26, 2022, 11:40	January 27, 2022, 13:10	January 27, 2022, 16:20	January 28, 2022, 11:40
14-5FTH	120	January 26, 2022, 11:40	January 27, 2022, 13:10	January 27, 2022, 16:20	January 28, 2022, 11:40
17-5FTH	80	January 26, 2022, 11:40	January 27, 2022, 13:10	January 27, 2022, 16:20	January 28, 2022, 11:40
19-5FTH	118	January 26, 2022, 11:40	January 27, 2022, 13:10	January 27, 2022, 16:20	January 28, 2022, 11:40

For safety reasons, the exact location of this test is not disclosed. However, it is noted that the test was conducted within the Montesinho Natural Park. Figure 5.11 illustrates the spatial arrangement of the modules across the terrain. This layout was carefully designed to capture the fire spread since its direction is north to south.

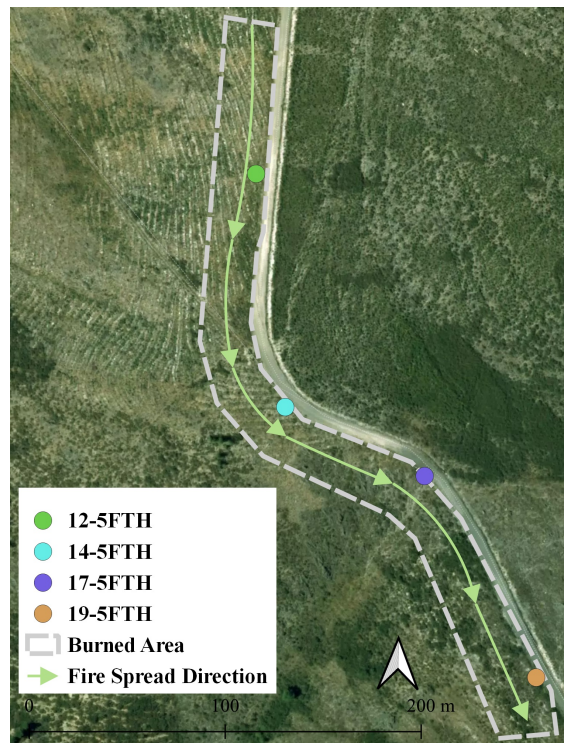


Figure 5.11: Spatial arrangement of the WSN modules across the test area in Dataset 4.

This strategic positioning ensures that the dataset captures critical environmental and sensor data as the fire spread through the terrain. By placing the modules equidistant from the fire path

and maintaining uniform height and alignment, the test setup provides an ideal scenario to evaluate the WSN's ability to monitor and detect fire-related events in a dynamic, controlled environment.

5.3.5 Testing WSN at Low TCD Region - Dataset 5

Dataset 5 was collected during a biomass clearance operation performed by firefighters, a routine practice to manage the volume of fuel material in specific areas. This process is essential for mitigating the risk of large-scale wildfires, as it reduces the fire spreading uncontrollably. During this operation, a WSN was deployed to monitor environmental changes and acquire data during the controlled and supervised fire propagation.

The chosen area contains variations in TCD, with values ranging from a maximum of 88 to a minimum of 42 and an average of 68, according to the TCD map [56]. These variations reflect the diverse vegetation density across the terrain, providing an opportunity to analyse how different biomass levels influence the fire's behavior and the sensor network's performance. Figure 5.12 shows the scenario in real-time during the fire.



Figure 5.12: The scenario with fire spread during data collection for Dataset 5.

The modules used for data collection – 05-5FTH, 06-5FTH, 08-5FTH, 10-5FTH, and 14-5FTH – were placed to stakes approximately 1.4m in height and positioned at an approximate distance of 2m from the fire's path in a straight line. The distances between the modules were measured relative to 05-5FTH, designated as the origin point (0m). The specific distances, as well as the dataset's start and end times and the fire's start and end times, are provided in Table 5.7.

Table 5.7: Modules placement details and WSE timing for Dataset 5.

Node ID	Distance [m]	Dataset Start (UTC)	WSE		Dataset End (UTC)
			Fire Start (UTC)	Fire End (UTC)	
05-5FTH	0	February 21, 2022, 15:00	February 22, 2022, 14:05	February 22, 2022, 15:55	February 23, 2022, 23:59
06-5FTH	26	February 21, 2022, 15:00	February 22, 2022, 14:05	February 22, 2022, 15:55	February 23, 2022, 23:59
08-5FTH	43	February 21, 2022, 15:00	February 22, 2022, 14:05	February 22, 2022, 15:55	February 23, 2022, 23:59
10-5FTH	47	February 21, 2022, 15:00	February 22, 2022, 14:05	February 22, 2022, 15:55	February 23, 2022, 23:59
14-5FTH	42	February 21, 2022, 15:00	February 22, 2022, 14:05	February 22, 2022, 15:55	February 23, 2022, 23:59

For safety and confidentiality reasons, the exact location of the test is not disclosed. However, the test was conducted in Rebordãoinhos. Figure 5.13 presents a satellite view of the chosen area, illustrating the layout and placement of the modules within the terrain.

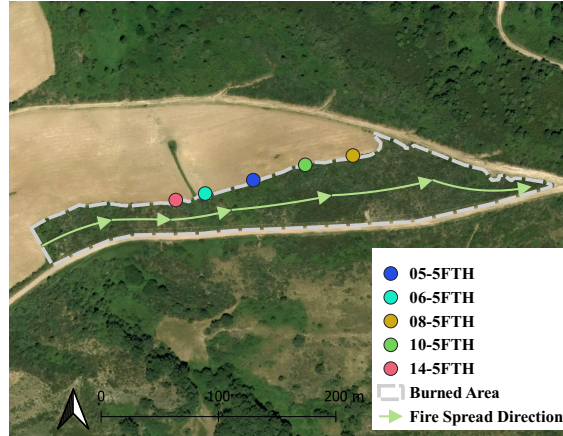


Figure 5.13: Spatial arrangement of the WSN modules across the test area in Dataset 5.

This dataset provides the **WSN** performance in an environment with different **TCD** levels during fire spread. The strategic placement of modules enables a detailed analysis of how vegetation density affects sensor readings and fire propagation dynamics.

5.4 Exploratory Data Analysis

In this section, the **EDA** is performed for each dataset collected from the **WSE** scenarios, referred to as Dataset 1 and Dataset 2. The purpose of conducting a detailed **EDA** on each dataset is to gain a comprehensive understanding of the data, identify patterns and anomalies, and assess the suitability of the data for subsequent modeling and analysis tasks.

For each dataset, the original data was plotted in scatter plots to visualize data distribution over time [271, 272]. Also, it is important to specify the start and end dates of the data acquisition period to provide context for the temporal scope of the data. An initial exploration is conducted using Python's data analysis libraries. By presenting textual summaries using the outputs of `.head()`, `.info()`, `.describe()`, and `.unique()` methods to understand the data structure, basic statistics, and uniqueness of values [273].

To understand the statistical properties of the data, it is indispensable to apply Skewness analysis [274], generate histograms [275], and use the Fitter library [276] to identify the best-fit distribution functions for each variable. This is crucial for selecting appropriate statistical models and detecting deviations from normality.

The Fitter library [276] can be applied to identify the best-fit distribution function for each variable without introducing bias. This can be achieved by systematically testing all distribution functions available within the package to determine the best fit for each dataset variable. Additionally, the selection method for the best-fit distribution must be based on the **Sum of Squared Errors**

(SSE) criterion, ensuring that the chosen distribution minimizes the error between the observed data and the fitted model.

Daily boxplots [275] can be helpful for visualizing variability and the daily distribution. In addition to the boxplot, the minimum, maximum, mean, and amplitude of daily values can help in understanding outliers and daily trends. A correlation matrix [275] needs to be generated to assess the interrelationships between the variables in the Datasets. This matrix quantifies the strength and direction of the linear relationships among the sensor readings. The correlation values range from -1 to 1 , where 1 indicates a perfect positive correlation, -1 a perfect negative correlation, and 0 no correlation.

Preliminary time series analyses can be implemented to examine the trends over intervals of 6h, 12h, and 24h. The trend component, derived using the **Seasonal-Trend Decomposition with Loess (STL)** method [277], captures the underlying long-term variations in the data by adaptively smoothing it through localized regression. This smoothing is performed using the Loess technique, which estimates the trend by fitting local regression models within sliding windows of the data. Each point is weighted based on its proximity to the center of the window, ensuring that nearby points have a more significant influence on the regression. This approach allows the trend to adapt to nonlinear patterns in the data while minimizing the impact of noise and transient fluctuations. This approach allows for evaluating the short-term and long-term variations in the datasets, which is essential for understanding the temporal dynamics of the monitored environment. In this context, the **Augmented Dickey-Fuller (ADF)** method [277] can be applied to assess the stationarity of each sensor variable.

ADF is a crucial property indicating whether the statistical properties of a time series, such as mean and variance, remain consistent over time. The outputs may suggest that most sensor variables are likely non-stationary, as their p values are significantly higher than the 0.05 threshold. A p value below this threshold indicates that the data are likely stationary, meaning the statistical properties of the time series remain stable over time, which is beneficial for time series modeling. On the other hand, detecting non-stationarity informs the necessity for data transformations or differencing in modeling.

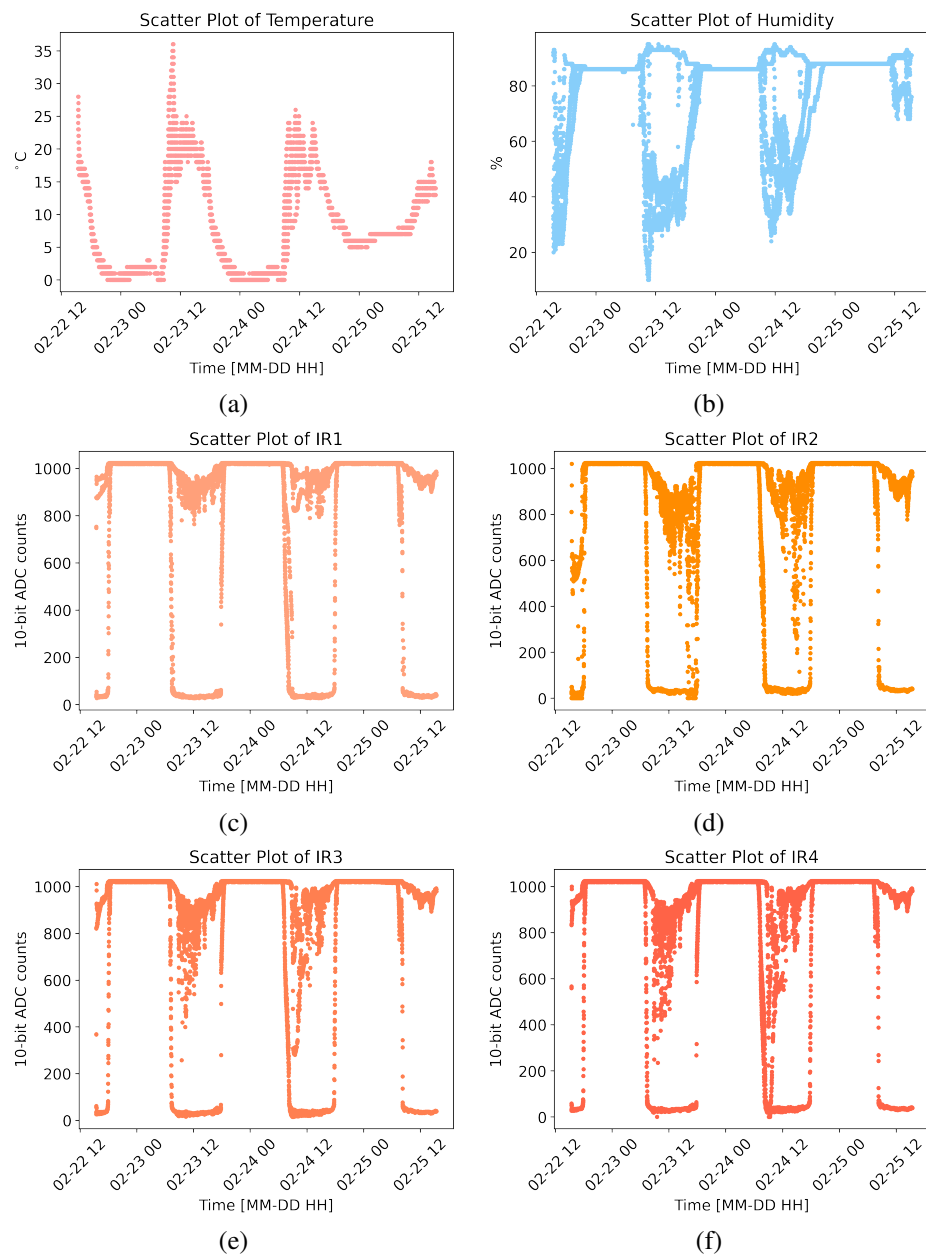
The **Autocorrelation Function (ACF)** [277] can indicate how each data point correlates with previous observations at several lags, while the **Partial Autocorrelation Function (PACF)** can indicate the direct impact of each lag after accounting for prior lags. This combined analysis is valuable for understanding underlying data patterns, detecting seasonal components, and identifying the order of autoregressive and moving average components (which can guide the subsequent analyses and implementation). Then, to evaluate both **ACF** and **PACF**, two red dashed lines were inserted into the plots. These lines were given by the equation of $\pm \frac{1.96}{\sqrt{n}}$, where n is the dataset length. Therefore, the lines represent the significance thresholds, helping to determine which lag values are statistically significant.

By conducting this comprehensive **EDA**, the aim is to fully understand each dataset and identify patterns or anomalies, ensuring the data is suitable for further analysis and modeling. This step is fundamental in the overall process, guiding the selection of appropriate modeling techniques and

the interpretation of results in the context of fire detection and monitoring using **WSN**.

5.4.1 Dataset 1 - Description and Analysis

The first dataset, referred to as Dataset 1, was collected over four days, from February 22, 2021, to February 25, 2021. During this time, sensor modules monitored environmental parameters and **WSE** under low **TCD** conditions. The modules involved in this dataset include 02-5FTH, 03-5FTH, 05-5FTH, 08-5FTH, and 09-5FTH, all of which were deployed to acquire readings of battery voltage, humidity, temperature, and data from five individual **IR** sensors. Figure 5.14 illustrates the raw data collected from the entire Dataset 1.



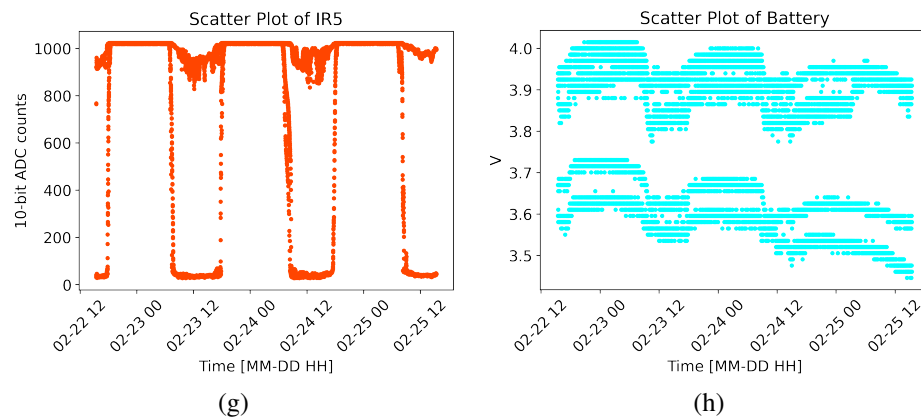


Figure 5.14: Temperature (a), Humidity (b), IR sensors 1-5 (c-g), and Battery (h) in Dataset 1 over the observation period.

A single **WSE** was observed in this dataset, lasting from 9h15 to 10h50 on February 23, 2021 – a duration of 1h35. This interval allowed for an analysis of the sensors’ responsiveness to fire conditions, as well as their behavior before and after the event. In this sense, a summary of Dataset 1 is provided next.

The dataset contains 18096 entries across 12 columns. Figure 5.15 illustrates the total count of entries for each sensor variable across all modules in the dataset. It presents the number of data points recorded for each sensor, including Battery, Humidity, Temperature, and the five **IR** sensors (IR1 to IR5), organized by module. Additionally, the daily data counts for the monitored period are as follows: 2264 entries on February 22, 6252 on February 23, 5937 on February 24, and 3643 on February 25.

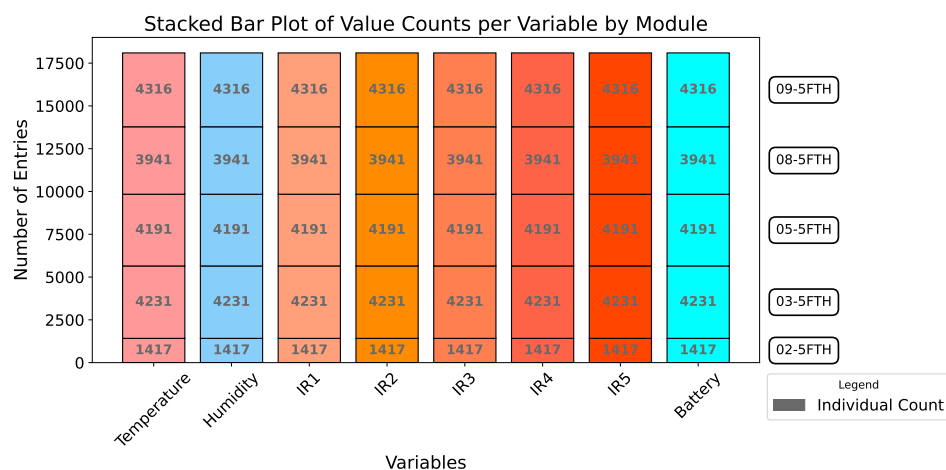


Figure 5.15: Total number of entries for Temperature, Humidity, IR sensors (IR1-IR5), and Battery recorded for each module in the Dataset 1.

Each row corresponds to a timestamp, capturing sensor values and module information. The columns include both categorical fields (Node and Node Version) which specify each module’s identification and configuration version. The `time` column uses the `datetime64` data type with

timezone awareness to record the date and time of each entry. Sensor readings, including `Battery`, `Humidity`, `Temperature`, and the five `IR` sensors (the columns' name are from `IR1` to `IR5`), are stored as `float64` values, while the `FS` column, which indicates the presence of a fire event, is of the `boolean` type. This dataset occupies approximately 1.5MB of memory and contains complete data with no null entries.

Descriptive statistics provide a summary of the central tendency, dispersion, and shape of the dataset's distribution. In this sense, starting by the `Battery` readings, the mean value is approximately 3.78 V, with a standard deviation of 0.17 V. The readings range from a minimum of 3.45 V to a maximum of 4.01 V. The 25th percentile is 3.61 V, the median (50th percentile) is 3.85 V, and the 75th percentile is 3.94 V.

`Humidity` values have a mean of 77%, with a standard deviation of 18.54%. The range extends from a minimum of 10% to a maximum of 95%. The 25th percentile is 73%, the 50th percentile is 86% (median), and the 75th percentile is 88%. `Temperature` values indicates a mean of approximately 8 °C, with a standard deviation of 7.06 °C, and values range from 0 °C to 36 °C. The 25th percentile is at 1 °C, the median (50th percentile) is at 7 °C, and the 75th percentile is at 14 °C.

The `IR` sensors, `IR1` through `IR5`, show mean values ranging from 760.95 to 798.68, with standard deviations around 400 for each sensor. The minimum values are close to zero, while the maximum values reach 1023 for all sensors, suggesting frequent high-signal readings. The specific values for each `IR` sensor are described below:

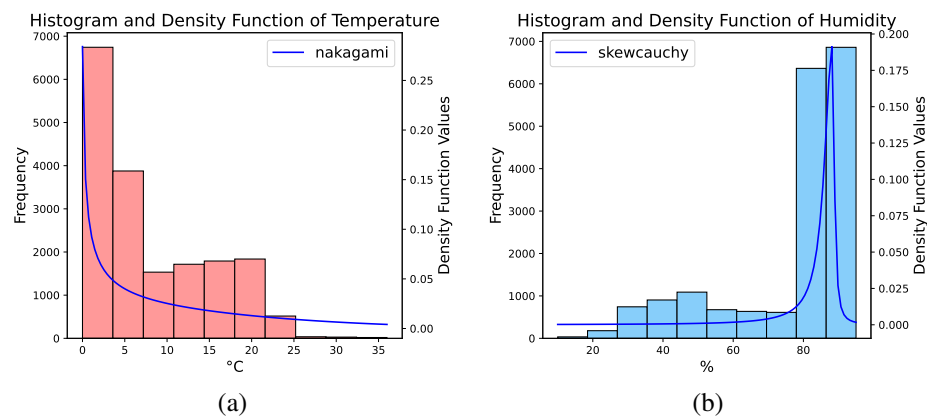
- **IR1:** Mean of 786.51, with a standard deviation of 399.61. The 25th percentile is 874.75, the median is 1022, and the 75th percentile is 1023.
- **IR2:** Mean of 760.95, with a standard deviation of 407.01. The 25th percentile is 614.5, the median is 1022, and the 75th percentile is 1023.
- **IR3:** Mean of 777.97, with a standard deviation of 400.70. The 25th percentile is 771, the median is 1021, and the 75th percentile is 1023.
- **IR4:** Mean of 776.19, with a standard deviation of 401.29. The 25th percentile is 741, the median is 1022, and the 75th percentile is 1023.
- **IR5:** Mean of 798.68, with a standard deviation of 393.36. The 25th percentile is 920, the median is 1022, and the 75th percentile is 1023.

The column `Time` has 18096 unique entries, confirming continuous monitoring. `Battery` and `Temperature` show moderate variability with 37 unique values each. While `Humidity` contains 86 distinct measurements. `IR` Sensors (`IR1` to `IR5`) exhibit a broader range of unique values, with: `IR1` contains 540 unique values, `IR2` registers the highest variability at 722 unique readings, `IR3` has 670 unique readings, `IR4` records 697 distinct values, and `IR5` presents 552 unique entries. `FS` presents 2 unique values, representing fire as (`True`) and non-fire as (`False`). `Node` includes 5 distinct modules, and `Node Version` confirms one version of the 5FTH module used throughout.

Of the 12 columns in Dataset 1, seven are explored in greater detail next, as they represent data directly acquired by the WSN monitoring system. These columns include sensor readings such as Temperature, Humidity, IR sensors (IR1 through IR5). The remaining columns, such as Battery, the Node Version and FS indicator, provide metadata about the modules and fire event occurrences. However, the focus of this detailed description is on the primary sensor data collected during the monitoring period.

To assess the distribution symmetry for the numerical variables in Dataset 1, the Skewness method was applied for each numeric column. The output reveal that the Humidity have a pronounced negative skew of -1.493 , suggesting that the data are left-skewed, with a longer tail on the lower end of the distribution. Conversely, Temperature exhibits a positive skew of 0.648 , indicating a slight rightward skew, with some values extending above the mean. The IR Sensors (IR1 through IR5) present significant negative skewness: IR1 at -1.298 , IR2 at -1.146 , IR3 at -1.244 , IR4 at -1.229 , and IR5 at -1.365 . This consistent negative skew across all IR sensors indicates that the majority of readings are concentrated at higher values, with fewer lower readings. These skewness patterns highlight asymmetries in data distribution that are essential to consider in subsequent analyses and model development.

The histograms in Figure 5.16 provide a visual representation of the distribution patterns identified through the skewness analysis for each sensor variable in Dataset 1. Along with the histograms, the analysis includes fitting distribution functions using the Fitter library to determine the best-fit distribution for each variable. The Fitter indicates that the Nakagami distribution fits the Temperature data, while the Skew Cauchy distribution best describes the other variables (Humidity and IR1-IR5).



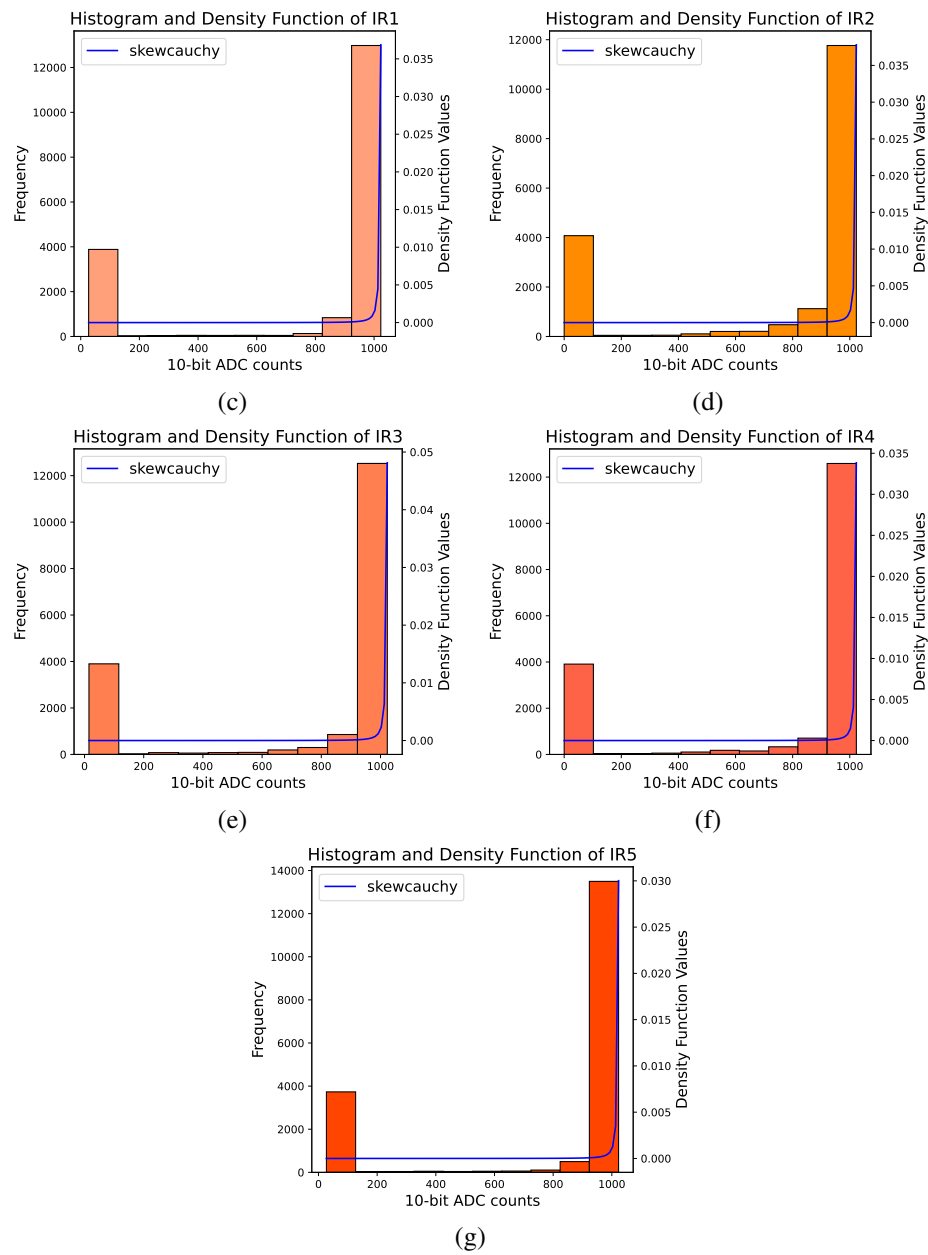


Figure 5.16: Histogram of Temperature (a), Humidity (b), and IR sensors 1-5 (c-g) in Dataset 1.

The distribution of sensor values over the monitored period can be visualized as boxplots, shown in Figure 5.17. Which is illustrated the minimum, maximum, mean, and range for each variable. These plots complement the statistical summaries described above, providing a visual inspection of daily variations.

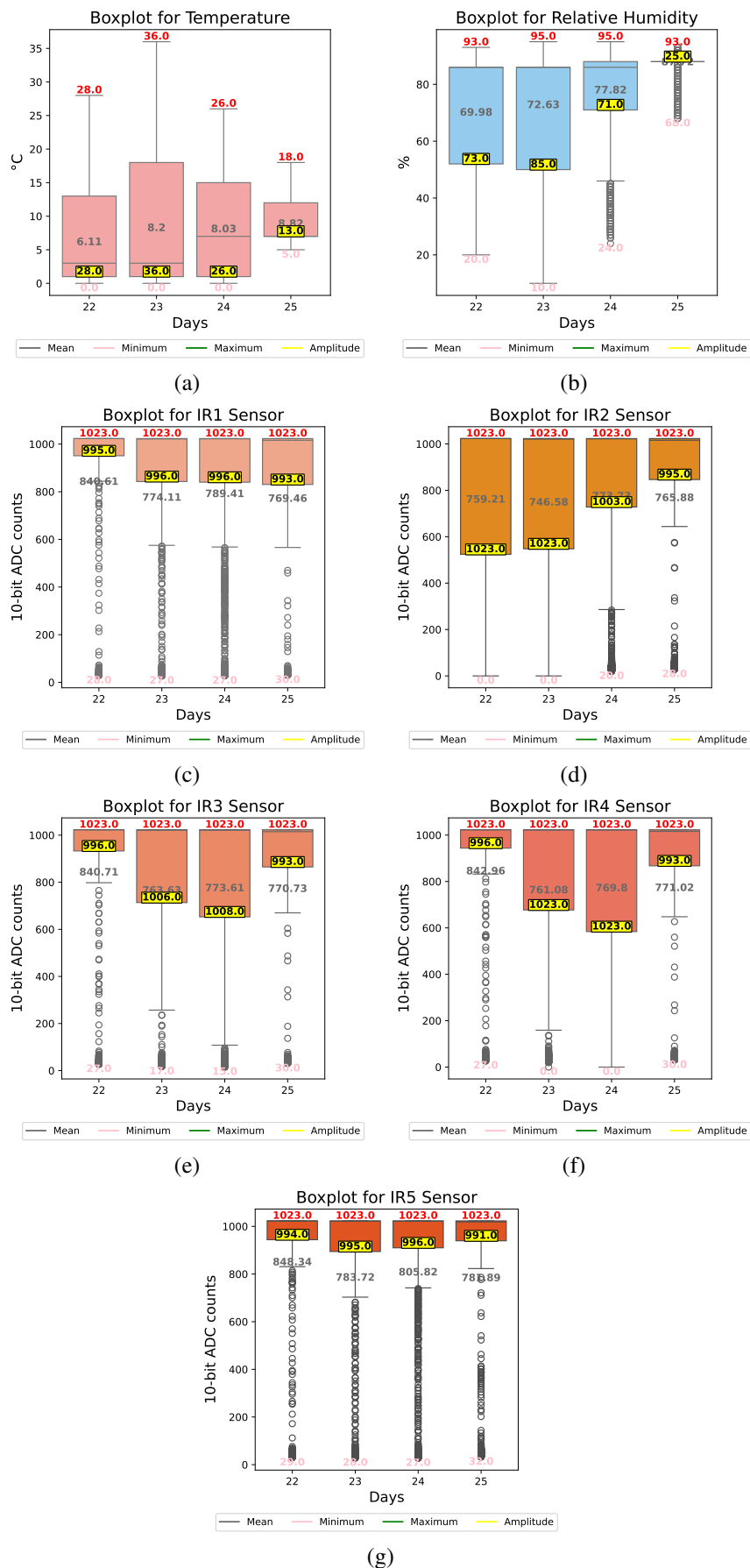


Figure 5.17: Boxplots illustrating the minimum, mean, maximum, and range for each variable (Temperature (a), Humidity (b), and IR sensors 1-5 (c-g)) in Dataset 1.

The boxplot illustrates the Temperature distribution over four days (22 to 25), emphasizing the minimum, maximum, and average values and daily amplitude. Day 23, when the fire test occurred, has greater amplitude and variability, indicating a possible impact on the measurements. In contrast, the other days show more stable temperature values, with minimums near 0°C and maximums ranging from 26°C to 36°C. While the boxplots suggest that the event on day 23 influenced the recorded readings, no outliers attributable to the fire could be identified.

The minimum, maximum, and average values and daily amplitude were highlighted in the boxplot Humidity distribution. Day 23, when the fire test occurred, shows a noticeable increase in variability compared to the other days, with values ranging from 10% to 95%. The other days exhibit more stable distributions, with minimums from 20% to 24% and maximums consistently around 93%. These boxplots suggest that the event on day 23 influenced the humidity measurements, yet no clear outliers associated with the fire were observed.

The boxplots of the IR sensors reading (IR1 to IR5) over four days range from minimum values of approximately 27 to 32 (with occasional zeros for IR2 and IR4) to maximum values of 1023, indicating full-scale measurements. Daily amplitudes are substantial, reaching up to 995 for IR1 and 1023 for IR2 and IR4. Day 23 does not present significant anomalies or outliers across any sensor, maintaining similar variability and amplitude compared to the other days.

A few notable patterns emerge from the correlation matrix in Dataset 1, as shown in Figure 5.18. The Humidity and Temperature variables show a moderately strong negative correlation of approximately -0.66 , indicating that higher temperatures generally coincide with lower humidity levels. The IR sensors (IR1 through IR5) display high positive correlations with each other, with values close to 1, suggesting that they respond similarly within this dataset. This analysis highlights the coherence among the IR sensors and the inverse relationship between temperature and humidity, which are essential aspects to consider in subsequent steps involving anomaly detection and model application for fire prediction.

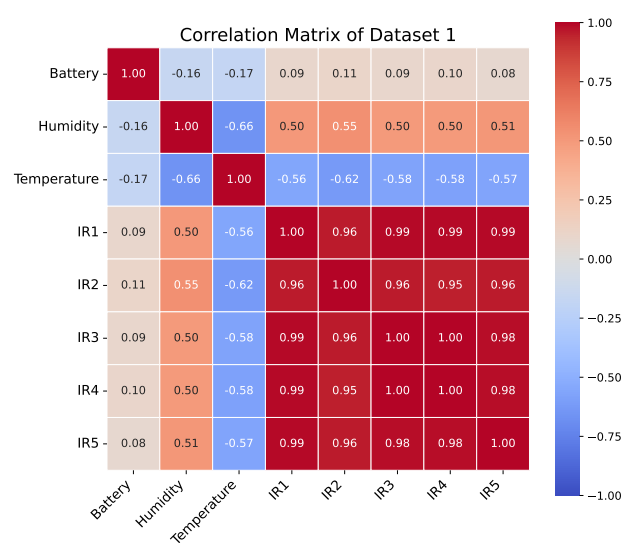


Figure 5.18: Correlation matrix of Dataset 1, illustrating the relationships among each variable.

In the trend analysis graph (outputted from **STL** method) shown in Figure 5.19, the original temperature data (top panel) is displayed alongside the following calculated trends. The 6-hour trend plot reveals more granular fluctuations, capturing rapid shifts in temperature that may correlate with immediate environmental changes or specific events within the observed timeframe. This interval exposes short-term variations in temperature, particularly in response to daytime heating and cooling patterns. The 12-hour trend smooths out some of these rapid changes, presenting a clearer picture of daytime versus nighttime patterns. This intermediate trend provides a balanced view, highlighting any persistent changes in temperature occurring across half-day intervals. On the other hand, the 24-hour trend plot highlights the overall daily temperature pattern. This trend illustrates the diurnal cycle, offering a general view of how temperature changes between each day without the influence of short-term fluctuations. This long-term trend is essential for understanding baseline temperature variations and provides a context for assessing unusual events.

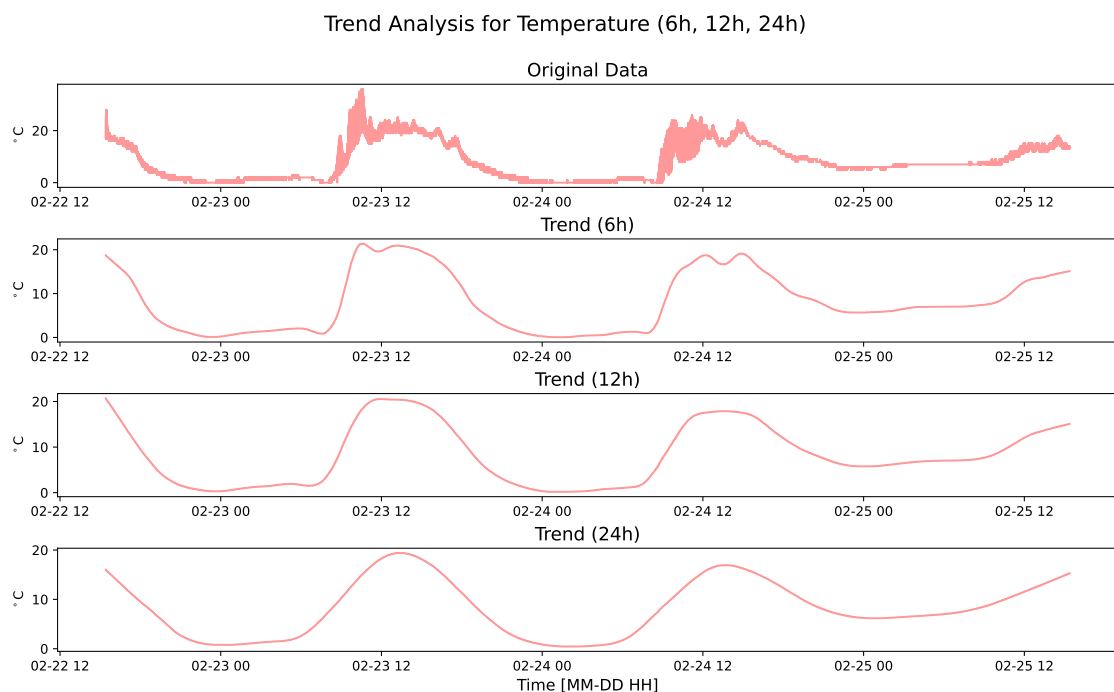


Figure 5.19: Temperature trends in Dataset 1 over 6-hour, 12-hour, and 24-hour.

Similar analyses were conducted for the variables Humidity and **IR** sensors (IR1-IR5). These analyses, which use the same **STL** method and focus on 6-hour, 12-hour, and 24-hour intervals, provide a comprehensive understanding of the trends and seasonal behaviors of the remaining variables in Dataset 1. Given the detailed nature and volume of the resulting plots, the continuation of this analysis is presented in Appendix D - Section D.1.

The **ADF** test was performed on each sensor variable in Dataset 1 to assess stationarity. Next, it is listed the **ADF** evaluation for each sensor:

- **Temperature:** The **ADF** statistic is -1.91 and a p value of 0.328 suggest that the temperature readings are non-stationary. This result indicates that temperature may experience trends or

seasonal variations across time, rather than fluctuating around a stable mean.

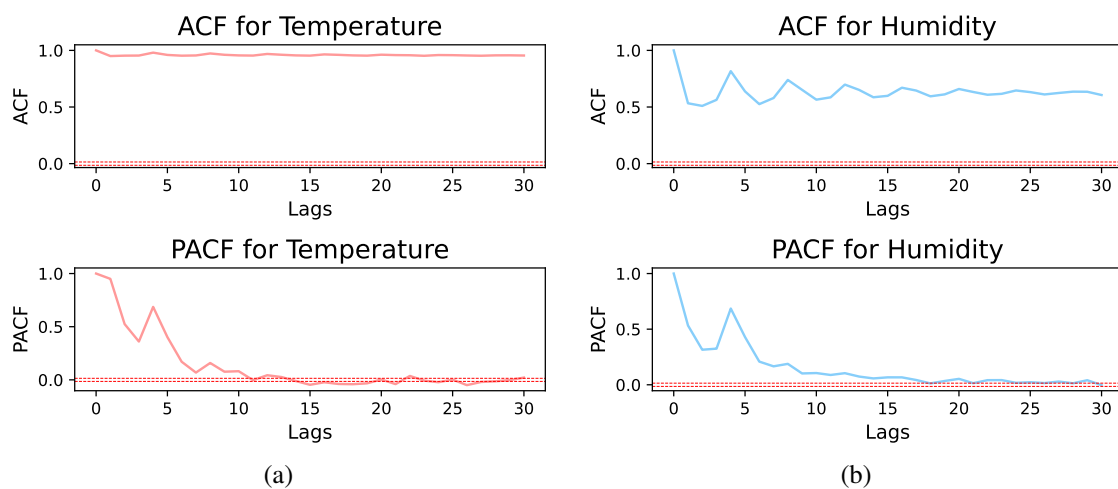
- **Humidity:** With an **ADF** statistic of -1.84 and a p value of 0.363 , the humidity data is also non-stationary. Humidity values likely show trends or cyclic patterns across days, which would be expected due to natural environmental cycles.
- **IR1:** **ADF** statistic of -1.67 and p value of 0.444 , indicating non-stationarity.
- **IR2:** **ADF** statistic of -2.23 and p value of 0.196 , also non-stationary.
- **IR3:** **ADF** statistic of -1.76 and p value of 0.401 , confirming non-stationarity.
- **IR4:** **ADF** statistic of -1.76 and p value of 0.401 , suggesting non-stationarity.
- **IR5:** **ADF** statistic of -1.67 and p value of 0.448 , indicating non-stationarity.

To explore the temporal dependencies and structure in Dataset 1, the **ACF** and **PACF** were analysed for each variable. Figure 5.20 contains the **ACF** and **PACF** graphs from each variable, indicating the values' behavior according to the applied lags.

Temperature **ACF** plot shows a steady decline within the first 10 lags, with no significant peaks afterward, suggesting a short-term dependency. The **PACF** plot exhibits a sharp peak at lags from 1 to 10, followed by smaller significant values at subsequent lags. This indicates an **AR** structure with considerable influence at the first lag and some diminishing influence beyond that.

Humidity **ACF** displays a rapid decline within the first few lags and then oscillates within the significance limits. The **PACF** reveals a clear peak at lags 4 and significant peaks at subsequent lags, suggesting an autoregressive pattern.

For the **IR** sensors (IR1 through IR5), the **ACF** plots display significant correlations at lags 1 and 2, with a gradual decline across subsequent lags, indicating a moderate temporal dependency. In the **PACF** plots, each sensor exhibits a notable peak at lag 4, with the remaining lags showing diminished values.



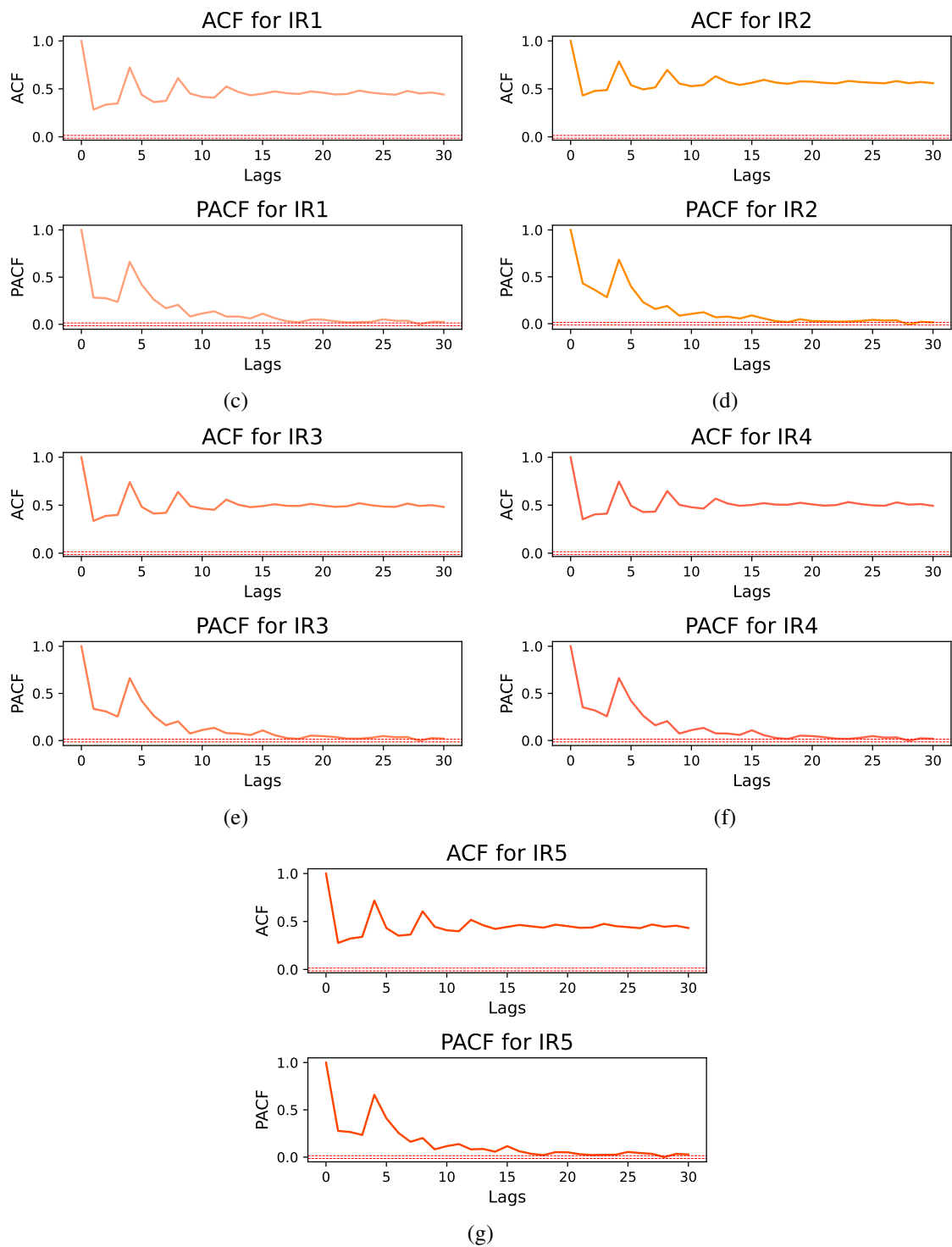
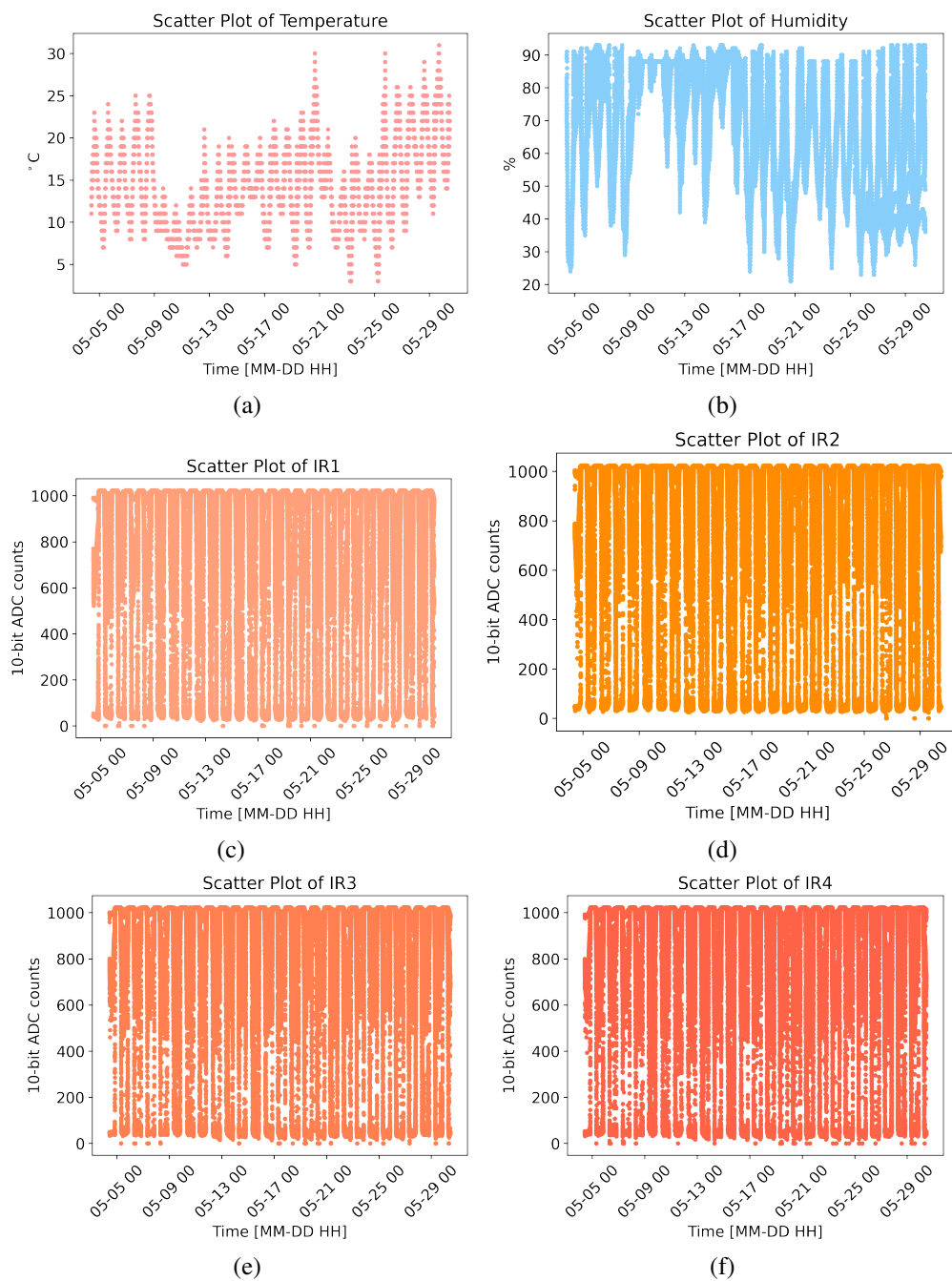


Figure 5.20: ACF and PACF plots for each variable (Temperature (a), Humidity (b), and IR sensors 1-5 (c-g)) in Dataset 1, with red dashed lines indicating significance thresholds for identifying statistically significant lags.

5.4.2 Dataset 2 - Description and Analysis

The second dataset, Dataset 2, was collected over multiple days in a scenario with higher **TCD** than in the previous dataset. This test, conducted from May 4, 2021, to May 30, 2021, aimed to evaluate sensor performance and data integrity under dense canopy conditions. The high **TCD** introduces unique environmental challenges, such as increased shading and potential obstructions, which impact sensor readings and data reliability. Figure 5.21 displays the raw data collected from all modules in Dataset 2, plotting the monitored variables individually.



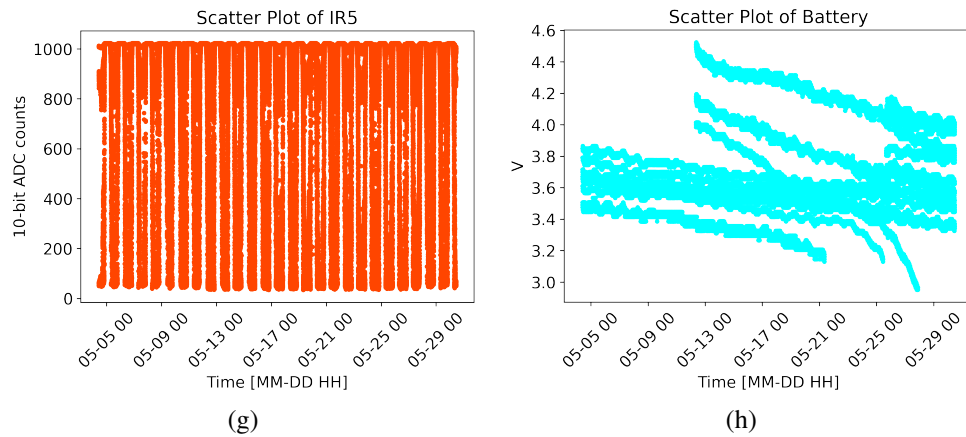


Figure 5.21: Temperature (a), Humidity (b), IR sensors 1-5 (c-g), and Battery (h) in Dataset 2 over the observation period.

The dataset has 319594 entries across 12 columns, each representing a specific sensor reading, timestamp, module identification, or metadata. Each row corresponds to a unique reading from one of the 13 modules collected in real-time intervals. The acquisition was set at a 60-second interval, with each module configured to transmit its payload in 60-second cycles.

The columns include `Battery`, `Humidity`, `Temperature`, and readings from five `IR` sensors (`IR1` through `IR5`). Additionally, `FS` column indicates the presence or absence of a fire event, while `Node` and `Node Version` provide metadata on the specific module and its configuration. The `time` column records the precise timestamp of each reading in UTC, ensuring temporal continuity across all modules. The absence of null entries in any column confirms the dataset's integrity, enabling accurate temporal and sensor-based analysis. The total count of entries for each module, segmented by variable, is displayed in Figure 5.22.

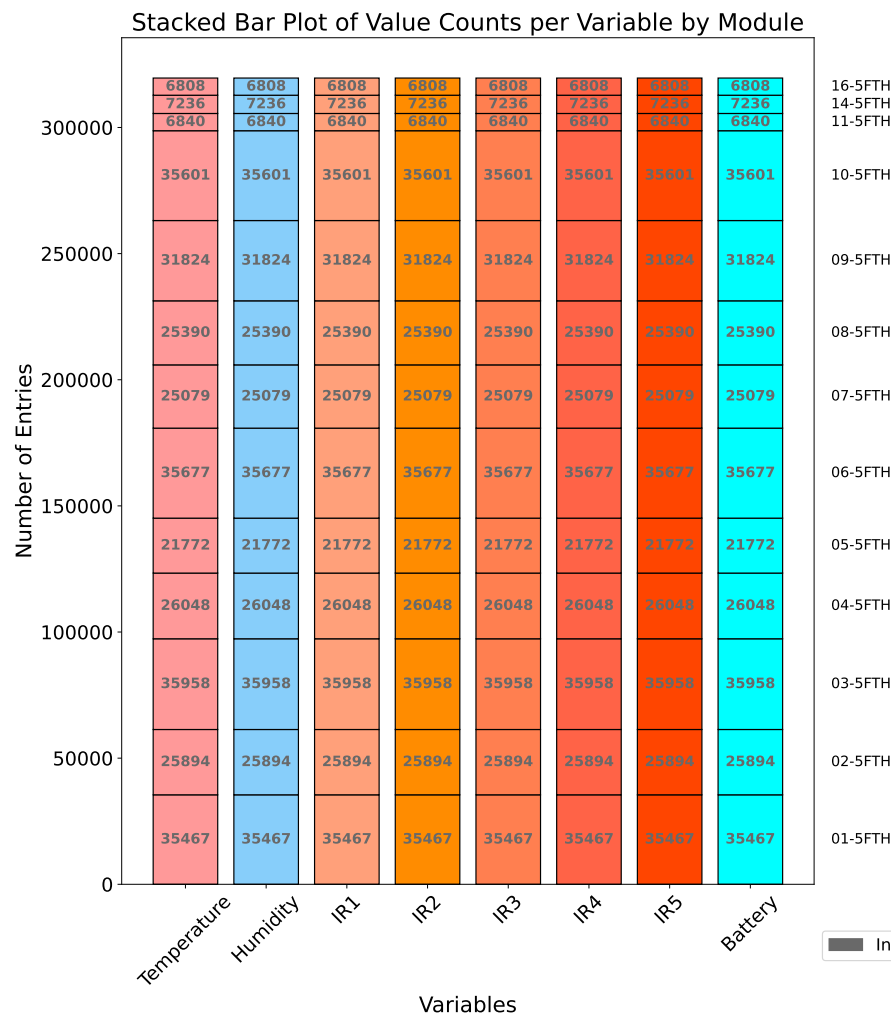


Figure 5.22: Total number of entries for Temperature, Humidity, IR sensors (IR1-IR5), and Battery recorded for each module in the Dataset 2.

For this dataset, 13 modules were deployed in a **WSN** around a central biomass fuel source. As previously mentioned, four separate **WSEs** were conducted during the testing period at different times, including one at night, to evaluate the performance of the **IR** sensors under different lighting conditions. The specific timing and duration of each **WSE** are: **WSE1**: May 12, from 15h00 to 15h45, with a total duration of 45 min; **WSE2**: May 19, from 23h20 to 23h40 (night-time test), lasting 20 min; **WSE3**: May 20, from 15h50 to 16h10, with a duration of 20 min; **WSE4**: May 25, from 18h25 to 18h50, running for 25 min. Given these four distinct **WSE** events, a detailed dataset analysis is required to fully understand the sensor behavior under high **TCD** conditions.

The daily counts of entries for Dataset 2 provide an overview of the volume of sensor readings recorded each day. Initial day, 4th, show moderate counts of around 5000 entries, increasing from May 5th onwards, with peak values observed on May 26th and 27th, reaching approximately 17000 daily. The fluctuations in daily counts are primarily attributed to the different numbers of active sensors (from Table 5.3). This data is visually represented in Figure 5.23, illustrating the daily

count pattern across the entire dataset collection period.

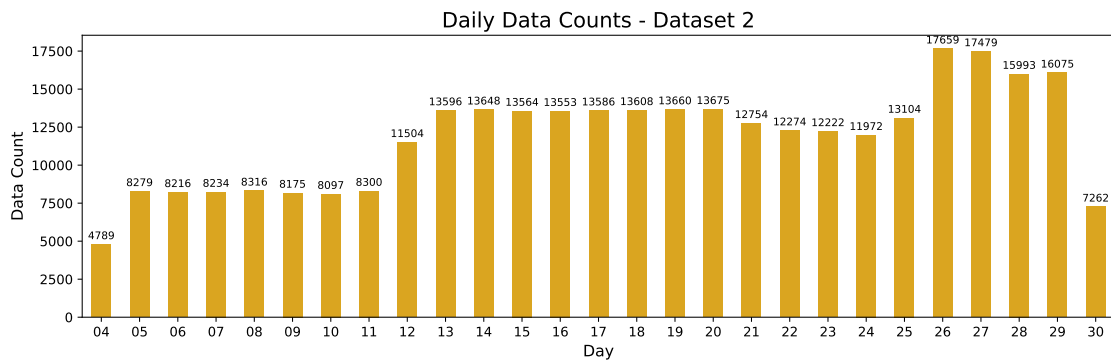


Figure 5.23: Daily sensor data entry counts for each day, illustrating the acquisition volume throughout the Dataset 2 collection period.

The columns include nine float type variables (Battery, Humidity, Temperature, five IR sensor readings, labeled IR1 through IR5, and Node Version), one boolean variable (FS), and two object-type variables (time and Node). The total memory usage of this dataset is approximately 27.1 MB.

The descriptive statistics for each variable in Dataset 2 indicates the dataset's distribution central tendency, dispersion, and shape. In this context, the Battery readings present an average value of 3.67 V and a standard deviation of 0.25 V. The minimum observed battery level is 2.95 V, and the maximum is 4.52 V. The first quartile (25%) is 3.50 V, the median (50%) is 3.61 V, and the third quartile (75%) is 3.82 V.

The mean is 66.01% and the standard deviation is 19.34%, for the Humidity values. These values range from a minimum of 21% to a maximum of 93%, with the first quartile at 49%, the median at 68%, and the third quartile at 87%. On the other hand, the average in the Temperature column is 14.24 °C, and the standard deviation is 5 °C. Temperatures range from 3 °C to 31 °C, with quartiles at 10 °C (25%), 14 °C (50%), and 18 °C (75%). The descriptive statistic analyses of the IR sensors are listed below:

- **IR1:** This sensor recorded an average reading of 734.23, with a standard deviation of 406.36. The values range from 0 to a maximum of 1023, with quartiles at 355 (25%), 997 (50%), and 1023 (75%).
- **IR2:** the readings present a mean of 737.96 and a standard deviation of 405.73. The minimum value is 0, and the maximum is 1023, with quartiles at 379, 994, and 1023 for the 25th, 50th, and 75th percentiles, respectively.
- **IR3:** the average reading is 733.10, with a standard deviation of 410.47. The values range from 0 to 1023, with 223, 997, and 1023 as the values for the 25th, 50th, and 75th percentiles, respectively.

- **IR4**: this sensor has a mean value of 730.85 and a standard deviation of 411.69. The minimum value is 0, and the maximum is 1023, with the quartiles at 185, 1000, and 1023.
- **IR5**: the mean is 791.04, with a standard deviation of 370.92. Values range from a minimum of 34 to a maximum of 1023, with quartiles at 7590 (25%), 1008 (50%), and 1023 (75%).

The unique value counts for each column in Dataset 2 reveal significant variability across sensor readings. The `time` column shows 319594 unique timestamps, indicating unique time entries for each reading across modules. The `Battery` column has 106 distinct values, reflecting a range in battery level readings. `Humidity` column presents 73 unique values, capturing diverse humidity measurements throughout the dataset. And the `Temperature` column has 29 unique values.

The **IR** sensors display significant variability, with unique value counts of 1015 for `IR1`, 1001 for `IR2`, 1014 for `IR3`, 1016 for `IR4`, and 990 for `IR5`. This high diversity suggests that each **IR** sensor captures a detailed range of intensity variations, essential for detecting environmental changes. The `FS` column has 2 unique values, which denote the presence or absence of a fire event. The `Node` column includes 13 distinct values, identifying each module uniquely, while `Node Version` holds 2 unique entries, corresponding to the different hardware configurations used in this scenario.

The subsequent analyses will focus on only the sensors' reading. These selected columns represent the essential measurements captured by the sensor modules, specifically designed to monitor environmental and **IR** variations around fire events. As the other columns serve to identify module specifications and fire event occurrences, while useful for context, they are not central to the detailed analysis of sensor behavior and environmental trends.

The Skewness method applied for each variable in Dataset 2 indicates asymmetries in the data distribution. A right-skewed is observed in `Temperature` (0.354), indicating a longer tail toward higher values. In contrast, `Humidity` has a slight left-skewed of -0.264 , suggesting a distribution that slightly leans toward higher humidity levels. The **IR** sensors (`IR1` through `IR5`) display more substantial left-skewed values, with `IR1` at -0.974 , `IR2` at -0.998 , `IR3` at -0.973 , `IR4` at -0.9560 , and `IR5` showing the most considerable skew at -1.285 . This indicates that the **IR** readings predominantly cluster around higher values, with fewer occurrences of low readings, a pattern that could be related to the sensors' sensitivity to consistent high **IR** levels in a high **TCD** environment.

The histogram plots alongside the best-fit distribution functions identified using the Fitter library can describe the dataset distribution in a graph mode. In this context, Figure 5.24 contains the plots with the histogram and best distribution density for each sensor reading. `Temperature` and `Humidity`, the Johnson SB (Bounded) distribution was the best fit, effectively modeling the slight asymmetry and capturing the natural variation observed in these environmental variables. The Levy L distribution provides the most suitable fit for `IR1`, `IR2`, `IR3`, and `IR4`. This distribution captures the frequent high and less common lower values. On the other hand, `IR5` aligns well with the Skew Cauchy distribution. All **IR** sensor distributions represent data concentration at higher values and its tail toward lower readings.

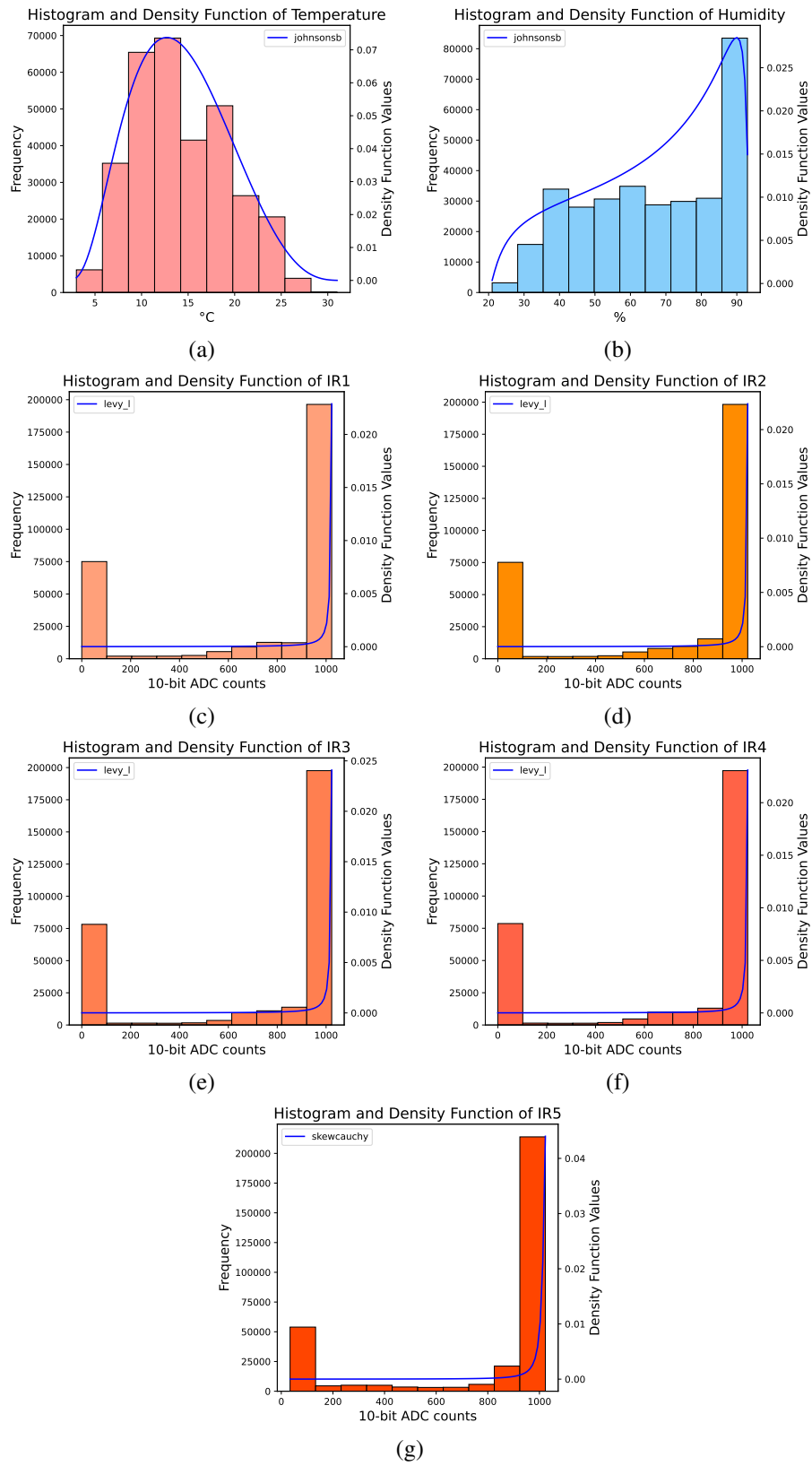
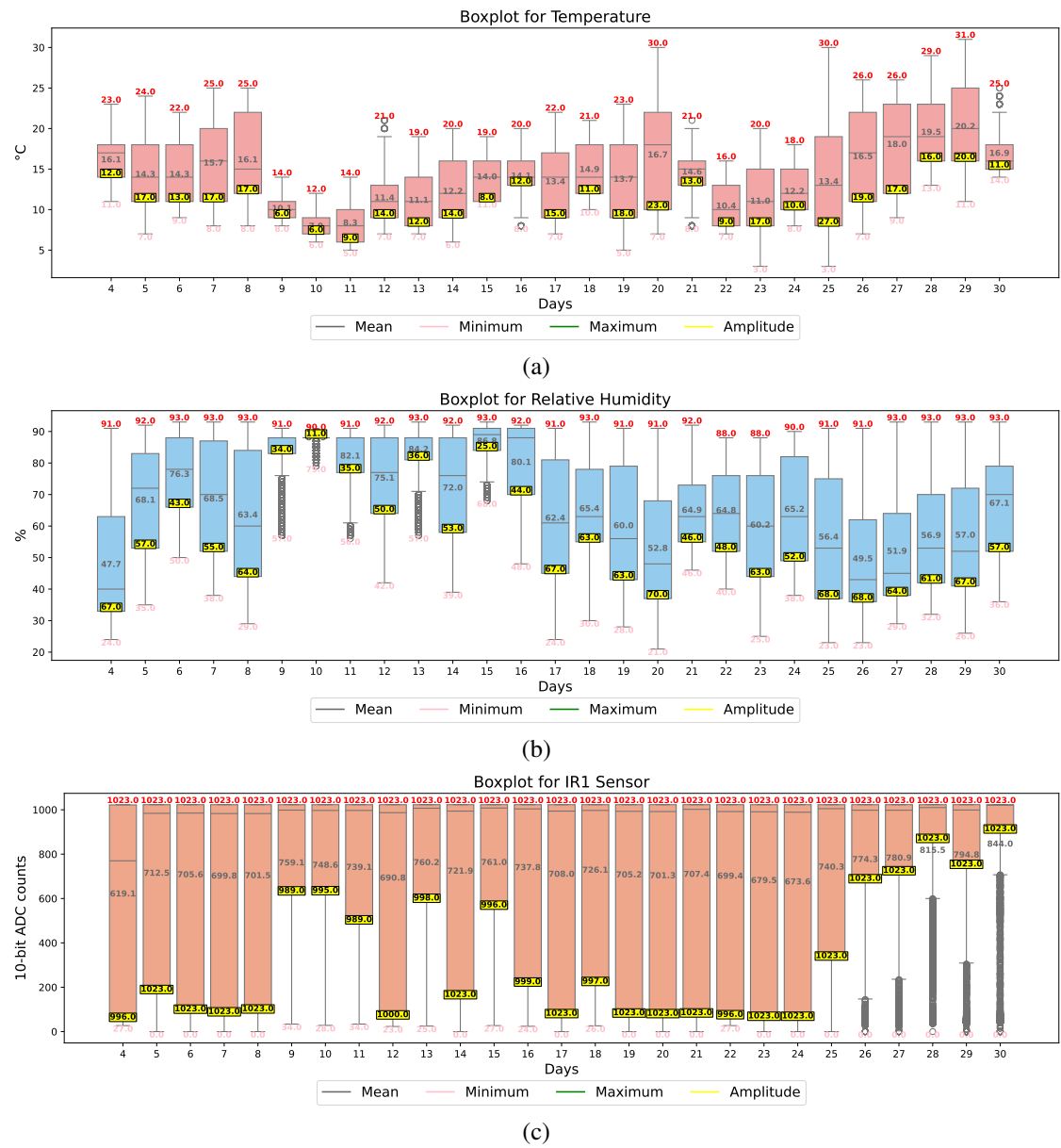


Figure 5.24: Histogram of Temperature (a), Humidity (b), and IR sensors 1-5 (c-g) in Dataset 2, which illustrate the data distribution and frequency over the observation period.

The boxplots provide a comprehensive view of the daily distribution patterns for each variable in Dataset 2. Throughout the observation period, Figure 5.25 exhibits the minimum, mean, maximum, and amplitude for Temperature, Humidity, and IR sensors. These plots highlight the central tendencies and the variability within each sensor’s data, revealing any outliers or shifts over time. By visualizing these metrics, the boxplots enable a clear comparison of daily fluctuations and help identify patterns or anomalies in sensor behavior throughout the dataset’s collection period.



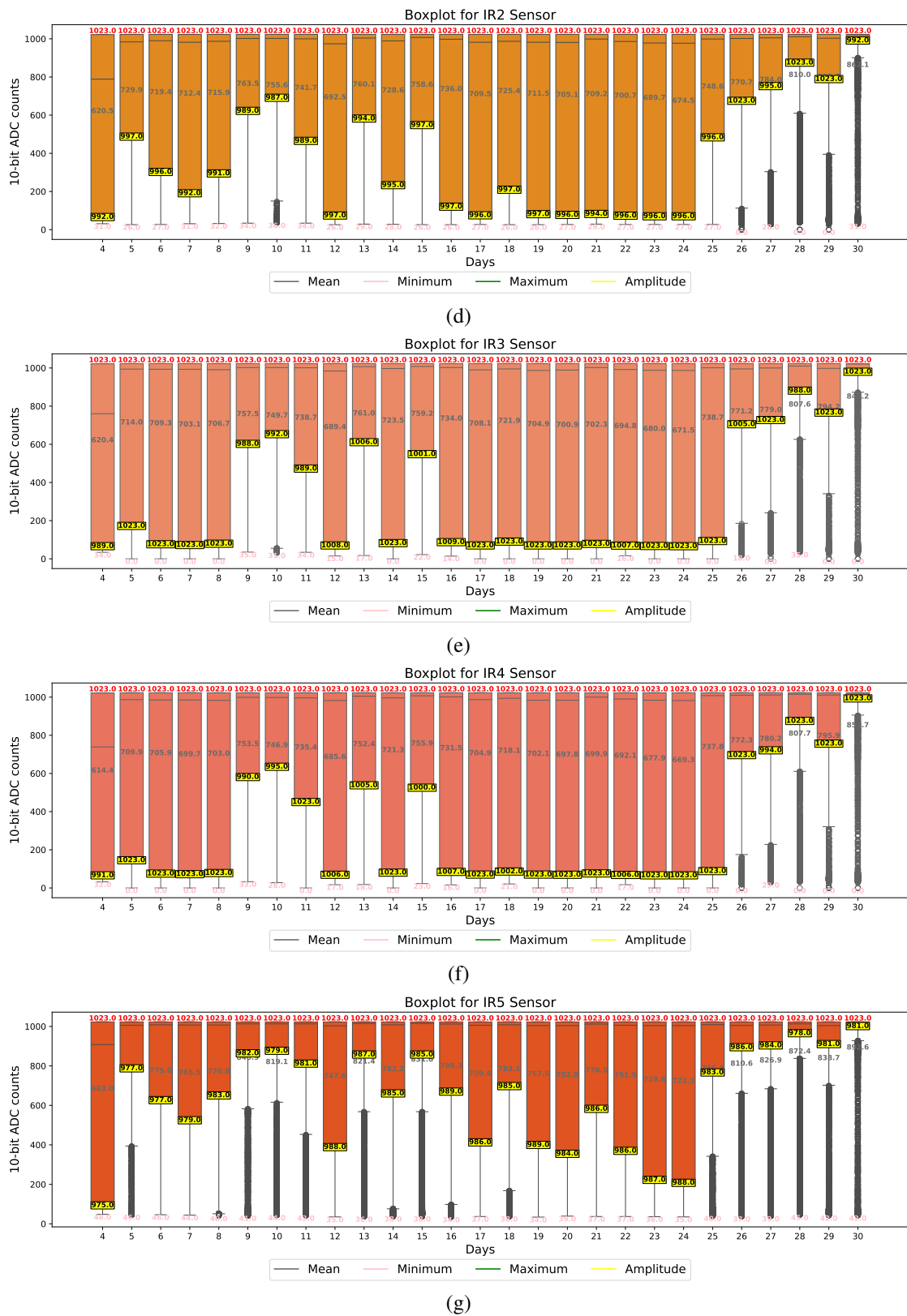


Figure 5.25: Boxplots illustrating the minimum, mean, maximum, and range for each variable (Temperature (a), Humidity (b), and IR sensors 1-5 (c-g)) in Dataset 2

The boxplot illustrates the Temperature distribution for Dataset 2 across multiple days (May 4-30, 2021). Temperatures range from 3 °C to 31 °C, with noticeable variations during fire events detailed in Table 5.4. On May 12 (WSE1), temperatures changed between 11 °C and 23 °C, showing moderate amplitude (12 °C). On May 19 (WSE2), the range expanded to 7 °C-24 °C, with higher variability (17 °C). May 20 (WSE3) presented the highest amplitude (23 °C), with temperatures ranging from 7 °C to 30 °C, suggesting a significant impact caused by WSE. On May 25 (WSE4), temperatures ranged from 14 °C to 31 °C with an amplitude of 17 °C. Non-fire days displayed smaller amplitudes and more stable distributions, highlighting the influence of fire events on temperature variability.

In the Relative Humidity's boxplot distribution, Humidity values range from 21% to 93%, with variations observed during fire events. On May 12 (WSE1), humidity oscillated between 24% and 91%, with an amplitude of 67%. On May 19 (WSE2), the range narrowed to 35%-92%, with a smaller amplitude of 57%. May 20 (WSE3) displayed the widest amplitude of 64%, ranging from 29% to 93%. On May 25 (WSE4), values ranged from 57% to 91%, with an amplitude of 34%. Non-fire days exhibit smaller variations, with a relatively stable humidity range, highlighting the influence of fire events on the recorded values.

The boxplot distributions of IR sensors reading (IR1 to IR5) across multiple days, ranging consistently from 0 to 1023, indicating the full scale of sensor measurements. Variability is observed across all sensors during fire events. On May 12 (WSE1), the amplitudes for IR sensors range between 989 and 996. On May 19 (WSE2), amplitudes remain high, around 997. May 20 (WSE3) shows slightly reduced amplitudes for some sensors, such as IR5 at 985, but maintains wide ranges across all. Finally, amplitudes stayed consistently large on May 25 (WSE4), close to 996 for most sensors. The non-fire days display similar ranges and amplitudes, suggesting the IR sensors may not distinctly capture fire events when a boxplot is used, as their distributions remain broad and relatively stable throughout the dataset.

The correlation analysis of Dataset 2, in Figure 5.26, reveals some distinct relationships among the variables. As expected, there is a negative correlation between Humidity and Temperature, with a coefficient of approximately -0.71 , indicating that humidity tends to decrease as temperature rises – a typical environmental pattern. Among the IR sensors (IR1 through IR5), correlations are consistently high, with coefficients above 0.95, suggesting that they exhibit similar responses to changes in environmental conditions or fire events.

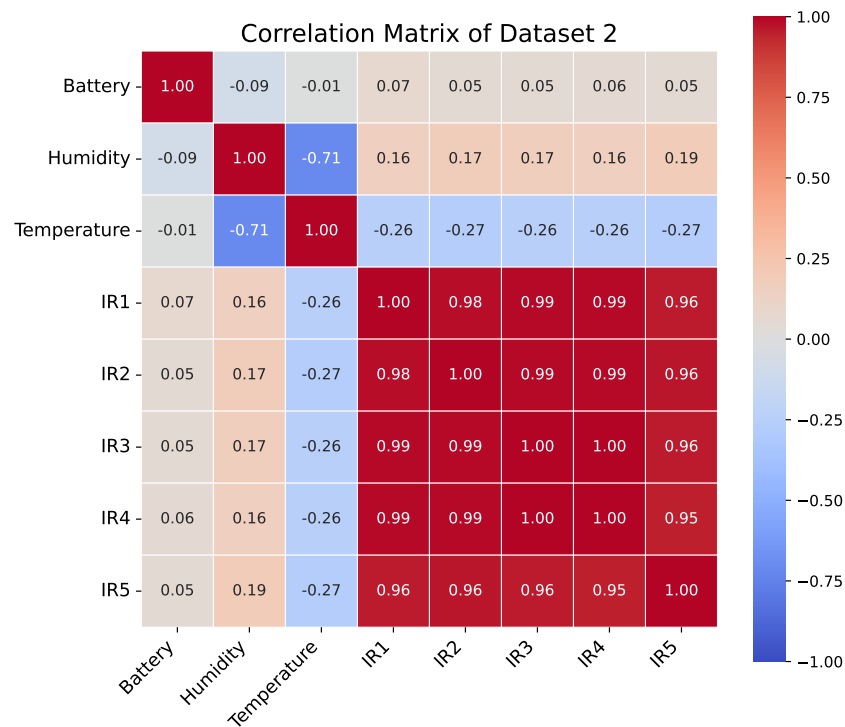


Figure 5.26: Correlation matrix of Dataset 2, illustrating the relationships among Temperature, Humidity, and IR sensor values (IR1-IR5).

The trend analysis, performed using **STL** method, the Temperature variable in Dataset 2, shown in Figure 5.27, provides information about the variations at different temporal intervals – 6h, 12h, and 24h. The original data in the top panel exhibits fluctuating temperature patterns across the month, with periodic spikes likely corresponding to changes in environmental conditions and fire events.

The 6-hour trend captures short-term fluctuations, offering a detailed view of temperature changes throughout the day. This interval is beneficial in identifying rapid temperature shifts potentially linked to daytime heating cycles or immediate responses to fire events. The 12-hour trend smooths these fluctuations, revealing more stable temperature transitions and allowing for a more apparent distinction between day and night cycles. Finally, the 24-hour trend provides a broad view of daily temperature patterns, capturing the diurnal cycle without the influence of shorter fluctuations. This long-term trend is essential for understanding overall temperature behavior, providing a baseline for identifying sustained deviations throughout the month.

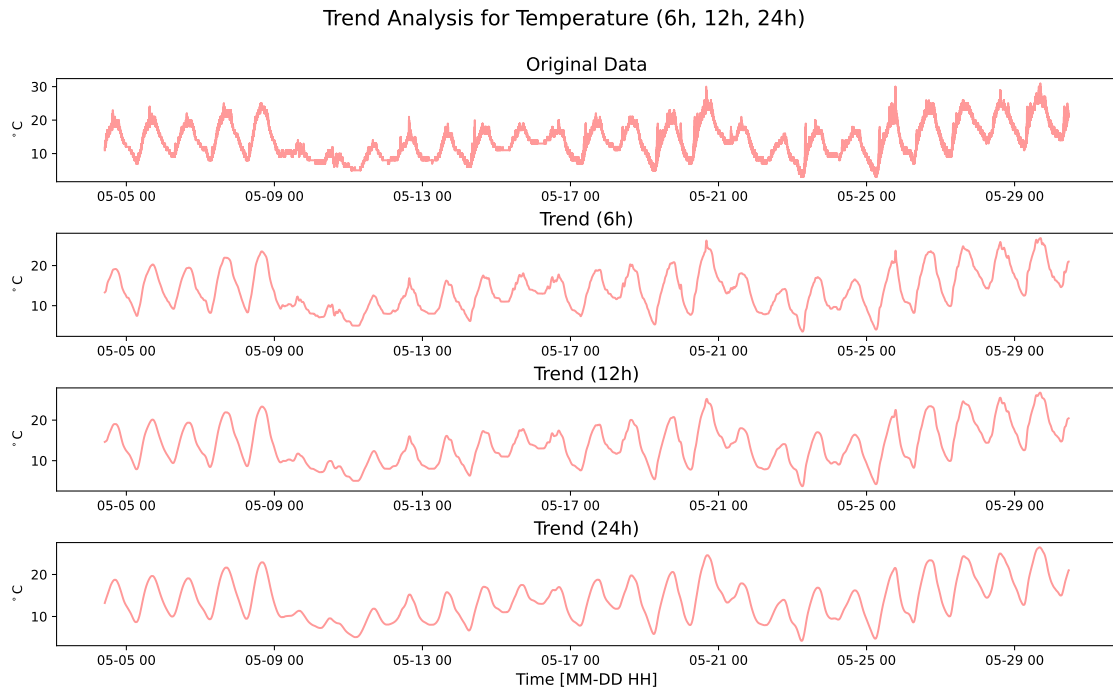


Figure 5.27: Temperature trends over different time intervals (6-hour, 12-hour, and 24-hour) in Dataset 2.

A similar approach was applied to analyse the variables Humidity and the **IR** sensors (IR1-IR5) in Dataset 2. Using the **STL** method, the trends were examined over 6-hour, 12-hour, and 24-hour intervals to uncover seasonal patterns and behavioral dynamics of these variables. Due to the detailed nature and extensive number of plots generated during this analysis, the results are presented in Appendix D - Section D.2, providing supplementary **EDA** into the dataset's temporal characteristics.

The **ADF** test for Dataset 2 indicates different levels of stationarity across the sensor readings. The **ADF** for Temperature outputs an **ADF** statistic of -2.2965 , with a p value of 0.1730 , suggesting non-stationarity as it does not adequate the significance threshold of 5%. Similarly, Humidity shows an **ADF** statistic of -2.3001 and a p value of 0.1719 , indicating that it is also non-stationary. These outputs suggest that temperature and humidity may exhibit trends or cycles over time rather than fluctuating around a constant mean, consistent with environmental patterns influenced by daily and seasonal cycles.

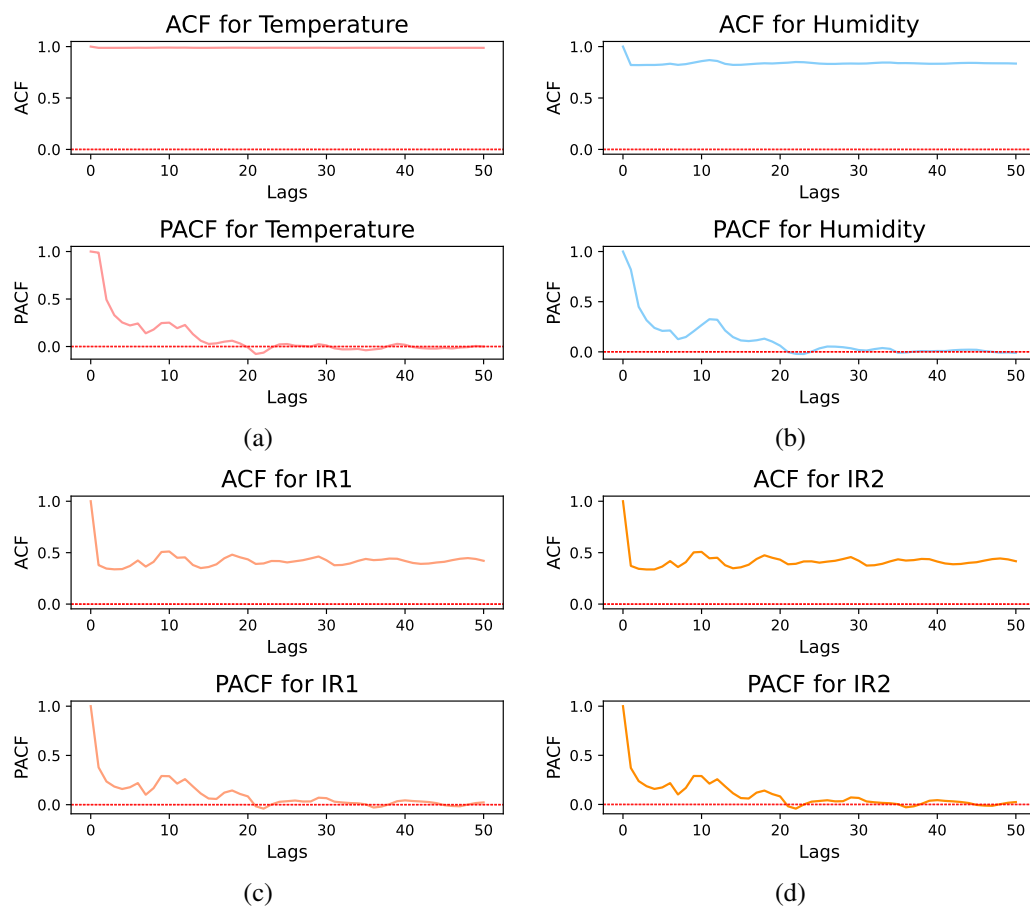
In contrast, all **IR** sensors (IR1 to IR5) are stationarity. IR1, IR2, IR3, IR4, and IR5 have **ADF** statistics of -5.051 , -4.934 , -5.072 , -5.021 , and -5.908 , respectively, with p values well below 0.05 , thereby rejecting the null hypothesis of non-stationarity. This stationarity suggests that the **IR** sensor readings fluctuate around a mean, which could reflect stable **IR** intensity patterns unaffected by cumulative trends.

To further examine temporal dependencies and patterns in Dataset 2, the **ACF** and **PACF** analyses were applied to each sensor variable. For Temperature, the **ACF** plot shows a decline in correlation across the first few lags, leveling off as the lags increase, suggesting short-term

dependency but without a solid seasonal component. The **PACF** plot displays significant values primarily at the initial lags, especially at lags 1, followed by diminished values.

The **ACF** plot for Humidity displays a decline in autocorrelation, with significant correlations at early lags, particularly noticeable up to lags 10. This pattern suggests that Humidity readings have a short-term dependency, influenced by recent values, but with diminishing correlations as the lag increases. The **PACF** plot also shows a peak at lags 1 and some reduced values beyond this, reinforcing the indication of a short-term dependency in humidity variations.

The **ACF** and **PACF** analyses for the **IR** sensors – IR1 through IR5 – present a consistent temporal pattern across these sensors. The **ACF** plots for each sensor exhibit significant correlations within the first two lags, followed by a decay. This indicates that recent past values influence each sensor's readings, with diminishing dependencies beyond the initial lags. Similarly, the **PACF** plots show peaks at lags 1, indicating that an **AR** structure where the first lag has the most substantial impact would be appropriate for these variables. Figure 5.28 presents the **ACF** and **PACF** plots for each variable in this dataset, detailing these dependency patterns.



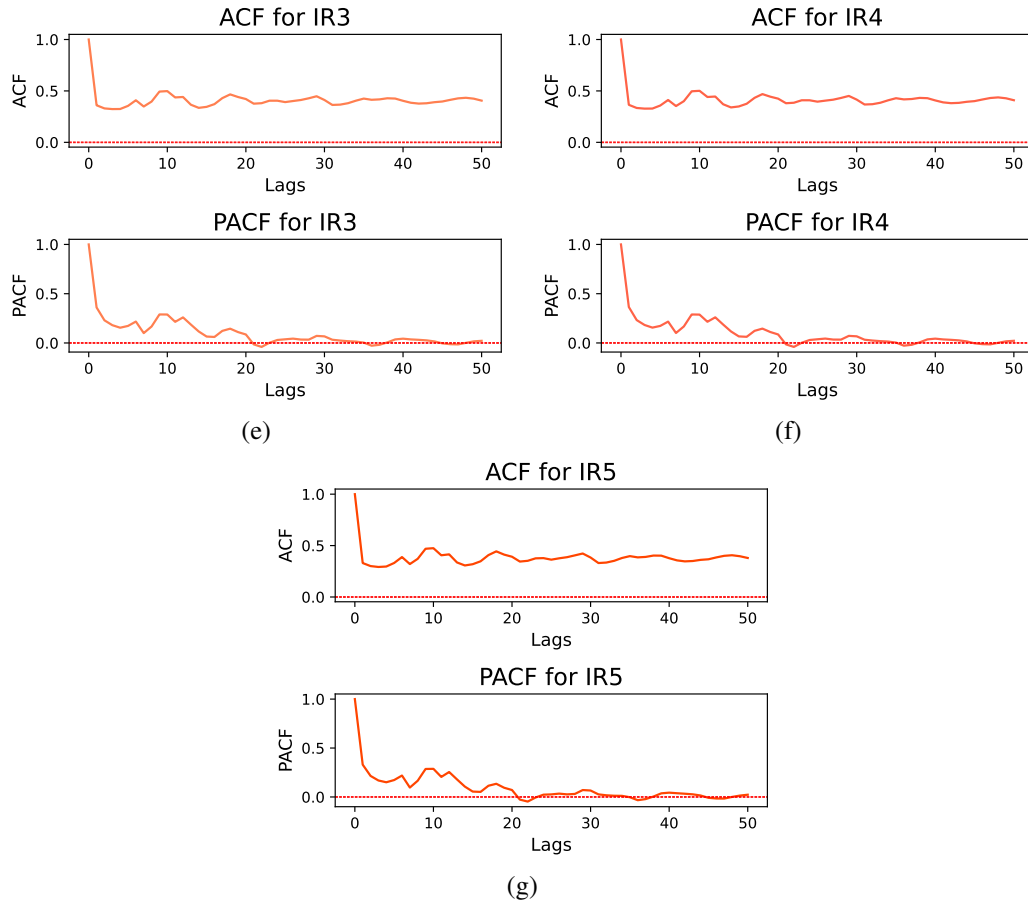


Figure 5.28: ACF and PACF plots for each variable (Temperature (a), Humidity (b), and IR sensors 1-5 (c-g)) in Dataset 2.

5.4.3 Dataset 3 - Description and Analysis

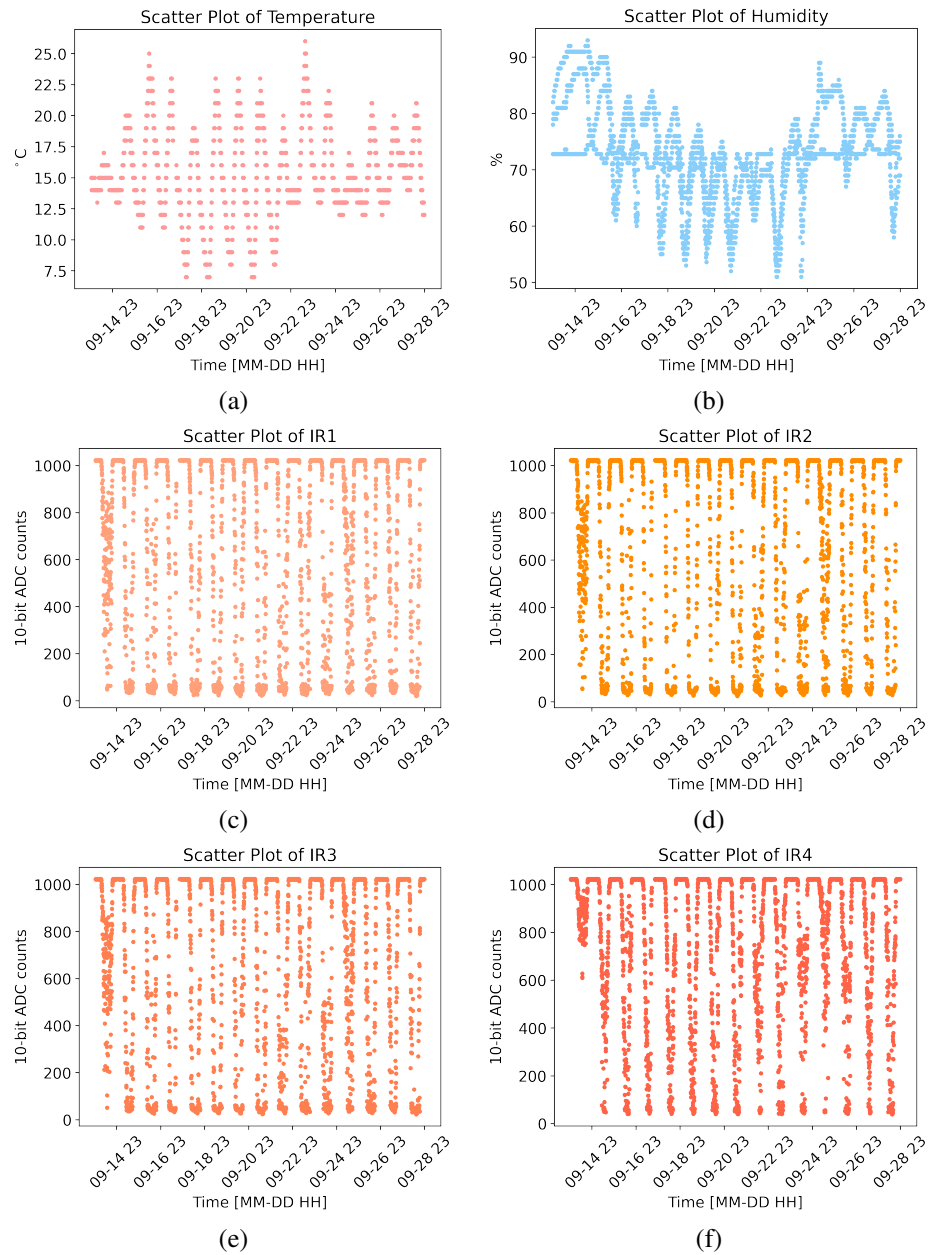
Dataset 3 was collected over a period of 15 days, beginning on September 14, 2021, at 0h00, and concluding on September 28, 2021, at 23h59. This dataset represents a high TCD scenario, where four modules, labeled as 12-5FTH, 14-5FTH, 17-5FTH, and 19-5FTH, were deployed at a height of 1.40m above ground. All modules used the 5FTH 2.0 model and were strategically positioned to monitor environmental conditions in a period (September) known for high temperatures and an elevated risk of fire.

Upon initial analysis of the data collected from the WSN in Dataset 3, it was observed that the sensor readings from module 14-5FTH deviated significantly from the expected patterns exhibited by the other modules. These deviations were evident across multiple variables and persisted consistently throughout the dataset. Given the controlled and uniform experimental setup, such discrepancies suggest potential issues related to the sensor calibration, hardware malfunction, or unexpected environmental factors uniquely influencing node 14-5FTH.

To ensure the reliability and consistency of the subsequent analysis, module 14-5FTH data was excluded. This decision was made to prevent potential biases or inaccuracies that could arise from

including outlier data that does not align with the intended conditions of the experiment. In this sense, the remaining data from modules 12-5FTH, 17-5FTH, and 19-5FTH will be used to conduct the intended evaluations.

The **DROWSIER** filter was implemented during the data collection process. This adaptive method ensures that data transmission occurs at least every 15 min or whenever the **DROWSIER** filter detects anomalous events. This approach optimizes network efficiency by prioritizing the collection of relevant data while reducing redundant transmissions. To provide an initial overview, Figure 5.29 presents scatter plots for each variable captured by the **WSN** in this scenario.



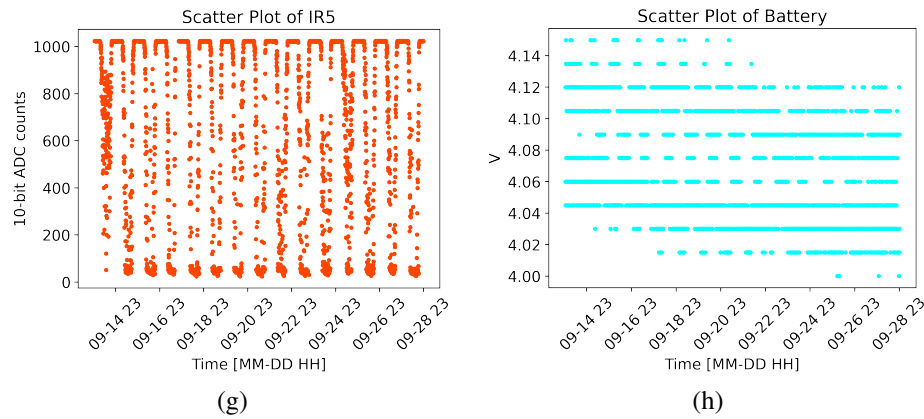


Figure 5.29: Temperature (a), Humidity (b), IR sensors 1-5 (c-g), and Battery (h) in Dataset 3 over the observation period.

The scenario resembles the conditions of Dataset 2, the **WSN** was positioned around a central firepit. Consequently, a **WSE** was conducted on September 21, 2021, lasting approximately 40 min, starting at 11h20 and concluding at 12h00. This configuration and controlled fire event were designed to replicate realistic environmental conditions, providing a robust testing ground for evaluating the effectiveness of the **ML** models and selected strategies. To illustrate the distribution of data entries across the variables and nodes, Figure 5.30 presents a stacked bar plot summarizing the value counts for each variable grouped by module **ID**.

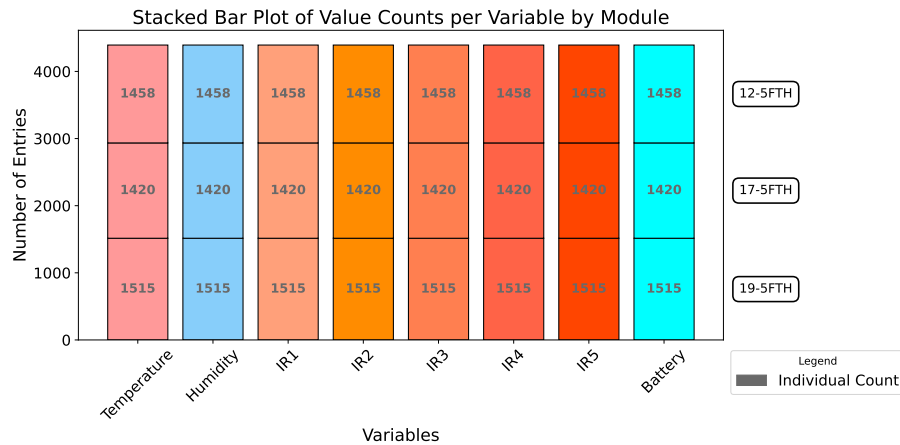


Figure 5.30: Total number of entries for each module in the Dataset 3.

With a total of 4393 entries, the Dataset 3 includes the following variables: Timestamps, Battery voltage, Humidity, Temperature, IR1 through IR5 sensor readings, a binary **FS** indicator, Node **ID**, and Node Version (in total 12 columns). The **Battery** column has 11 unique values, while **Humidity** and **Temperature** exhibit 66 and 20 unique values, respectively, indicating a range of environmental conditions during the test. The **IR1-IR5** display between 703 and 795 unique values, and three distinct **Node IDs** and a single **Node Version** confirm the uniformity of module types and configurations. Additionally, **FS** column marks fire-related periods, with **True** indicating the

presence of fire and `False` indicating its absence. The data is stored as `float64` for numerical values, `bool` for the `FS` column, and object for categorical fields (`Node` and `Node Version`). The dataset uses approximately 381.8 kB of memory.

The plot indicates an even distribution of entries across variables for all modules, consistent with the **DROWSIER** filtering approach applied during data collection. Notably, module 19-5FTH has the highest count with 1515 entries, while 17-5FTH has the lowest with 1420 entries. This difference could indicate that the **DROWSIER** filter was more active in some modules than others, potentially detecting more anomalous events and resulting in higher transmission frequencies. To further analyse the data distribution, Figure 5.31 presents the daily entry count for each module.

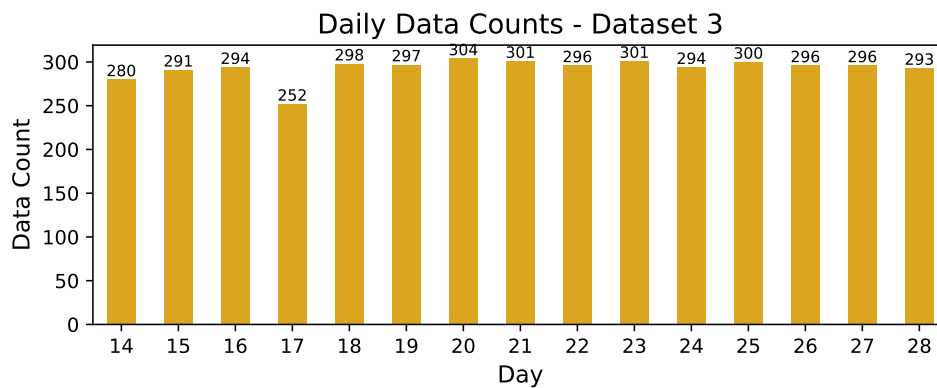


Figure 5.31: Daily sensor data entry counts for each day, illustrating the acquisition volume throughout the Dataset 3 collection period.

The average Battery voltage across all modules is approximately 4.06 V, with a standard deviation of 0.031 V. The minimum observed value is 4.0 V, and the maximum is 4.15 V, indicating stable battery performance during the test. Humidity readings range between 51 % and 93 %, with an average value of 73.18 % and a standard deviation of 8.22 %. The **IQR** is between 69 % and 79 %, showing typical variations for the high **TCD** region during September.

Temperature values range from 7 °C to 26 °C, with a mean of 15.2 °C and a standard deviation of 3.6 °C. The **IQR** have values between 13 °C and 18 °C, consistent with the seasonal high-temperature conditions described earlier. The **IR** sensors (IR1 to IR5) presented diverse responses, with mean values of approximately 668.26, 660.379, 673.18, 799.70, and 685.06, respectively. The sensor values ranged from 20 to 1023 for IR1, 24 to 1023 for IR2, 23 to 1023 for IR3, 39 to 1023 for IR4, and 24 to 1023 for IR5, with standard deviations ranging from 318.42 to 428.89. The high maximum values suggest frequent saturation during activity, while the variability highlights differences in sensor responses.

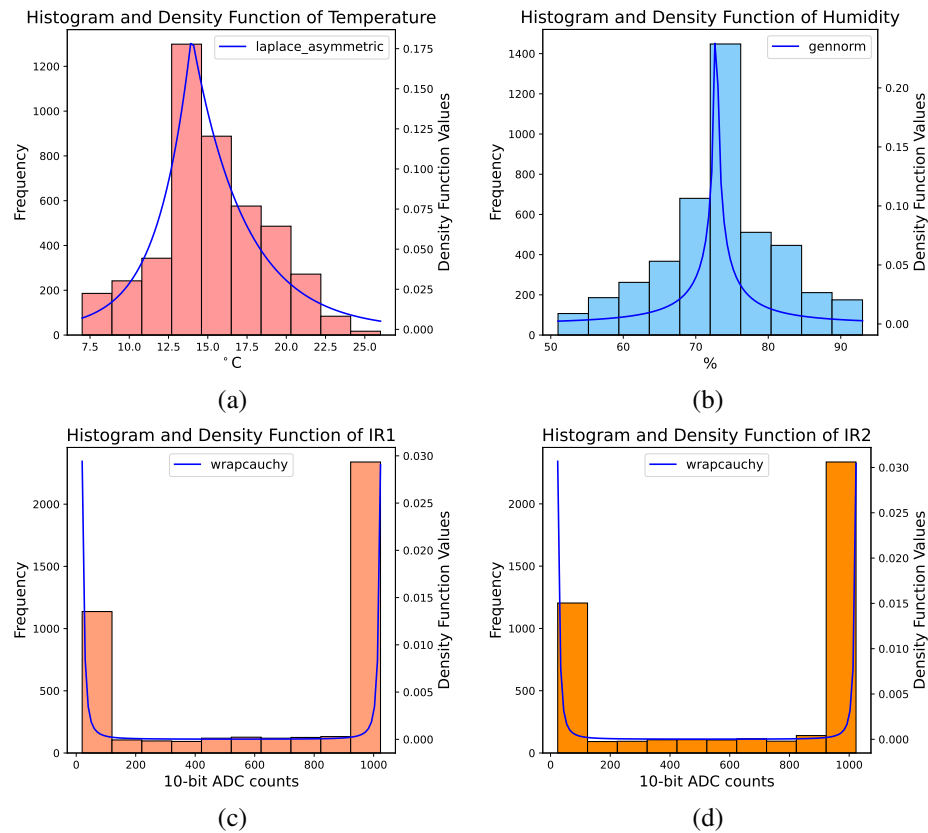
The Battery variable, despite its importance in monitoring the operational status of the modules, is not directly relevant to environmental or fire-related analysis. Similar to the approach taken in Datasets 1 and Dataset 2, the Battery values will be excluded from subsequent analyses.

The skewness analysis provides the symmetry of the distributions for the numeric variables in Dataset 3. A skewness value close to zero indicates a symmetric distribution, while negative

or positive values denote left or right skew, respectively. Humidity and Temperature also show slight skewness, with -0.10 and 0.10 , respectively, indicating that humidity has a left skew while temperature has a right skew. The IR sensor readings (IR1 to IR5) exhibit notable negative skewness, indicating distributions skewed to the left, with lower values occurring more frequently than higher ones. Specifically, IR1 has a skewness of -0.58 , IR2 -0.54 , IR3 -0.60 , IR4 -1.27 , and IR5 -0.66 . This consistent pattern among the IR sensors suggests that while most readings are concentrated at lower ADC values, higher readings are less frequent but present.

The histograms for the variables in Dataset 3 were generated to explore their distributions, accompanied by the best-fitting probability density functions identified using the Fitter library. The Temperature variable was best described by a Laplace Symmetric distribution, reflecting its central tendency and symmetrical variability. For the Humidity variable, the Generalized Normal distribution provided the best fit, effectively capturing the unique patterns in its variability. Among the IR sensors, IR1-IR3 exhibited a Wrap Cauchy distribution, emphasizing its cyclic and concentrated characteristics. Finally, IR4 and IR5 were most accurately described by the Skew Cauchy distribution, indicative of its asymmetrical data patterns.

These fitted distributions provide information about the data patterns and are a foundation for subsequent anomaly detection and forecasting analyses. The fitted histograms and their respective probability density functions are illustrated in Figure 5.32.



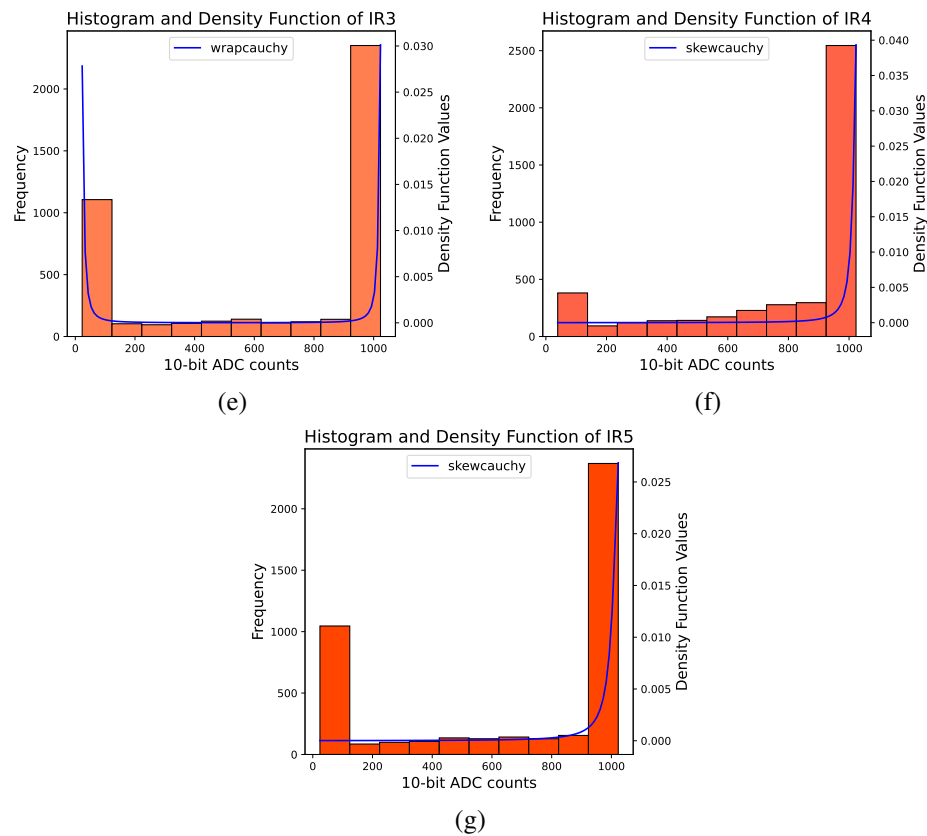
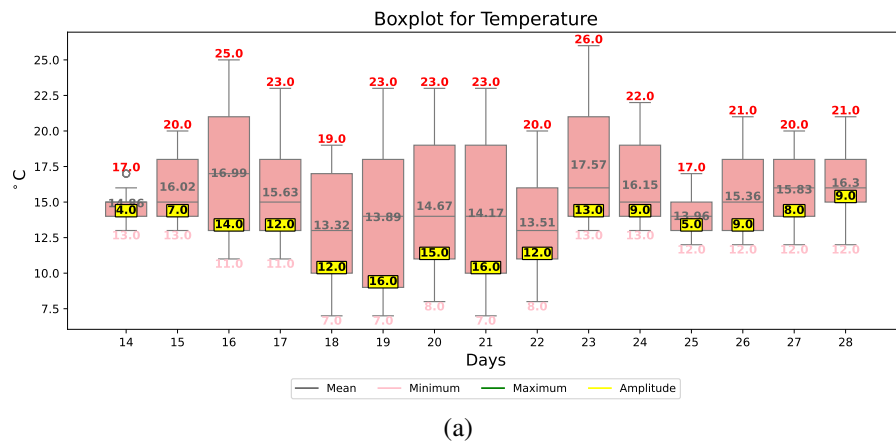
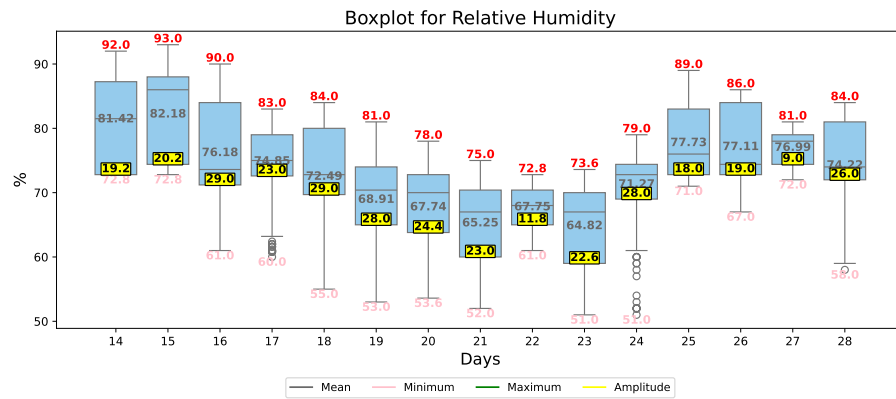


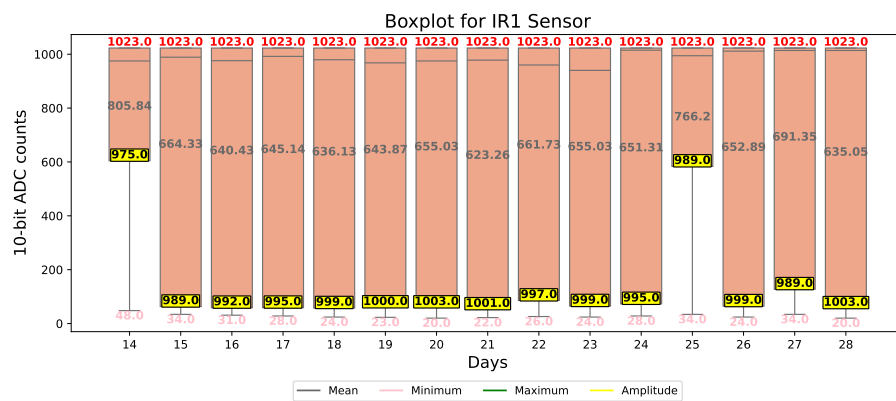
Figure 5.32: Histogram of Temperature (a), Humidity (b), and IR sensors 1-5 (c-g) in Dataset 3, which illustrate the data distribution and frequency over the observation period.

The variability and potential outliers in Dataset 3 can be seen in boxplots generated for all sensor variables (Figure 5.33 represents the boxplots for Temperature, Humidity, and IR1-IR5). Boxplots provide a visual summary of the data, highlighting the median, interquartile range, and extreme values. By analysing these plots, it is possible to assess whether the data exhibit consistent patterns or if specific periods or sensors deviate significantly, potentially signaling anomalous events.

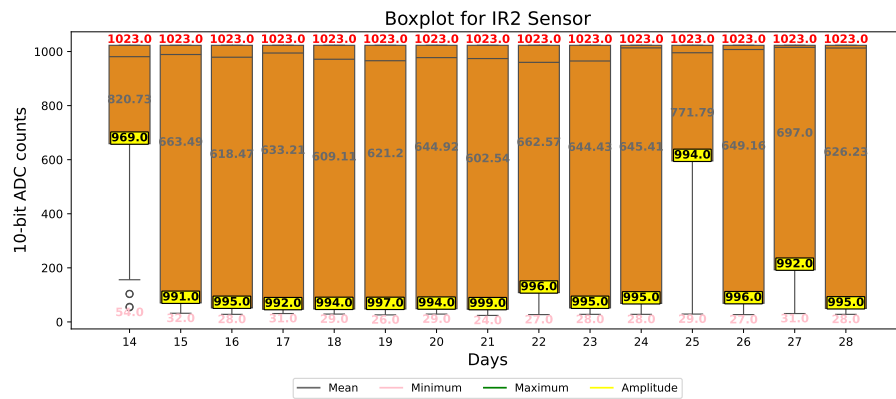




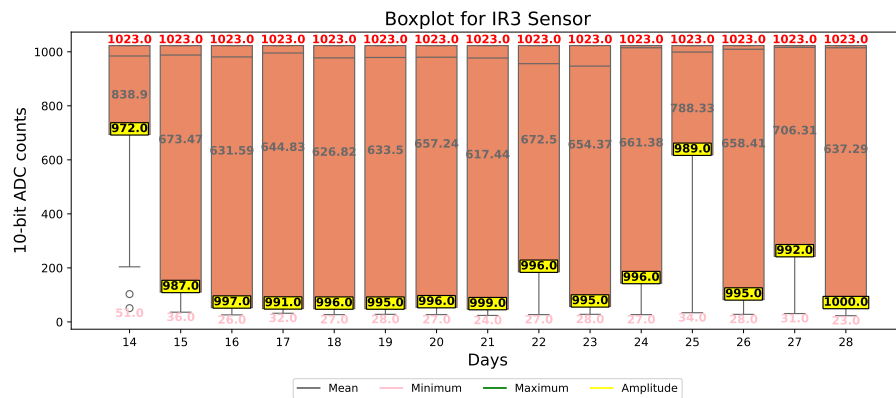
(b)



(c)



(d)



(e)

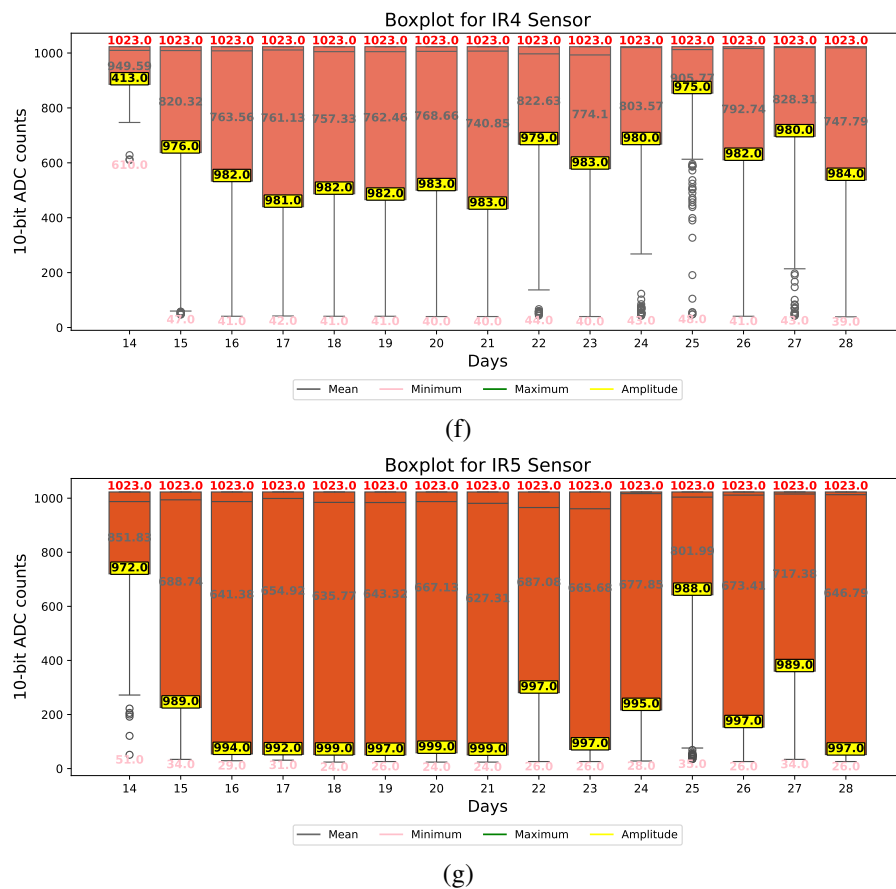


Figure 5.33: Boxplots illustrating the minimum, mean, maximum, and range for each variable (Temperature (a), Humidity (b), and IR sensors 1-5 (c-g)) in Dataset 3

The boxplot illustrates the Temperature distribution for Dataset 3 across multiple days (September 14-28, 2021). Temperatures range from 7°C to 26°C, with notable variations on September 21, the day of the fire event. Other days display smaller amplitudes, such as 4°C on September 14 and 7°C on September 15. These patterns suggest that the fire event on September 21 influenced the temperature readings, creating a wider range and higher values than non-fire days. Overall, the dataset shows moderate variability in temperature, with the fire event standing out for its more significant impact, and no outliers were identified.

The Humidity distribution for Dataset 3 across multiple days ranges from 51% to 93%, with variations observed on September 21, the day of the fire event. On this day, humidity ranged between 52% and 91%, with an amplitude of 39%. Despite the fire event, the humidity patterns on September 21 do not exhibit the most extreme deviations; instead, September 24 and September 23 represent the first and second days with the most significant amplitudes, respectively. Overall, the dataset shows moderate variability in humidity, with amplitudes ranging from 9% to 29% across all days. IR sensor readings (IR1 to IR5) for Dataset 3 across multiple days in September show a consistent range across the boxplots. On September 21, the day of the fire event, the sensors exhibited amplitudes ranging from 413 to 1003, with mean values between 602.54 (IR2) and 949.69

(IR4).

A correlation matrix was generated (Figure 5.34) to examine the relationships among all variables, including Temperature, Humidity, and IR1 through IR5. This correlation analysis helps identify potential interdependencies and interactions between the variables.

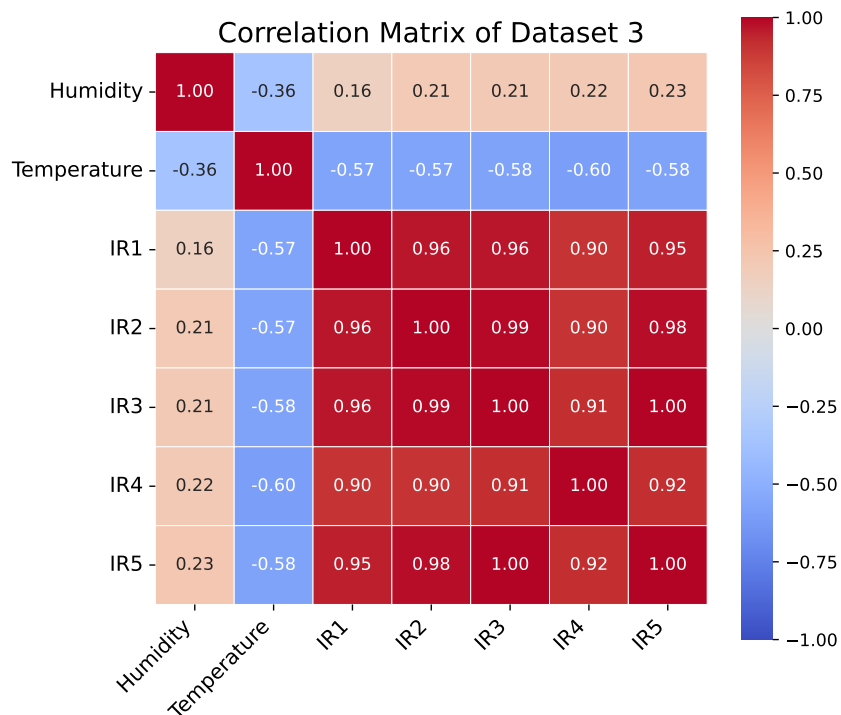


Figure 5.34: Correlation matrix of Dataset 3, illustrating the relationships among Temperature, Humidity, and IR sensor values (IR1-IR5).

A negative correlation of -0.34 is observed between Temperature and Humidity. This value is lower than in the previous dataset. This is probably due to the seasonal characteristics of September when temperature increases, but humidity does not follow a proportional inverse relationship. This divergence might also reflect the impact of fire dynamics, which locally distort expected environmental relationships.

The IR sensors (IR1 through IR5) show positive correlations among themselves, ranging from 0.9 to 1.0. This is consistent with their function and proximity in monitoring the same firepit environment and for sunlight noising. For example, IR3 and IR5 correlate perfectly (with a max of 1.0) due to their spatial arrangement and shared exposure to the fire source (both are fixed at the same modules' surfaces). Slightly lower correlations might be explained by differences in their direct alignment with the heat source or noises from sunlight, such as 0.96 between IR1 and IR2 (arranged in opposite directions), and all correlations of IR4 range from 0.90 to 0.92 (IR4 is directed to down).

The ADF test results indicate that all variables in Dataset 3 are likely stationary. Temperature, Humidity, and IR sensor readings (IR1-IR5) exhibit highly negative ADF statistics, with p values

well below the 0.05 threshold, confirming stationarity. These findings validate the dataset's suitability for time series modeling techniques that assume stationarity, such as **ARIMA** or **RNN**-based approaches.

The **ADF** test for Dataset 3 revealed stationarity across all sensor variables. The **ADF** statistic and corresponding p values for each variable are detailed below:

- **Temperature:** The **ADF** statistic of -7.90 and p value lower than 0.05, indicating stationarity. The test rejects the null hypothesis, suggesting that temperature fluctuates around a stable mean.
- **Humidity:** An **ADF** statistic of -4.24 and a p value of 0.00056 suggest that humidity readings are stationary. This implies stability in the readings over time without significant trends or cycles.
- **IR1:** With an **ADF** statistic of -6.94 and a p value lower than 0.05, IR1 is confirmed to be stationary.
- **IR2:** An **ADF** statistic of -6.72 and a p value lower than 0.05, suggest stationarity for IR2.
- **IR3:** The **ADF** statistic for IR3 is -6.87 , with a p value lower than 0.05, confirming stationarity.
- **IR4:** The **ADF** statistic of -7.22 and a p value lower than 0.05, confirm stationarity for IR4.
- **IR5:** An **ADF** statistic of -7.12 and a p value lower than 0.05, suggest stationarity for IR5.

These **ADF**'s outputs indicate that all variables in Dataset 3 exhibit stationarity, implying that their values oscillate around a consistent mean without significant trends or non-stationary behaviors.

5.4.4 Dataset 4 - Description and Analysis

Dataset 4 was collected in a scenario with controlled and supervised fires to analyse **WSN** data in a low **TCD** environment. The data collection began on January 26, 2022, at 11 h40 and concluded on January 28, 2022, at 11 h40. This dataset incorporates the implementation of the **DROWSIER** filtering strategy, ensuring that data is collected at a minimum interval of 15 min or whenever an anomalous event is detected by the filter. Figure 5.35 illustrates the scatter plot of the variables, providing an overview of the data distribution and relationships across the dataset.

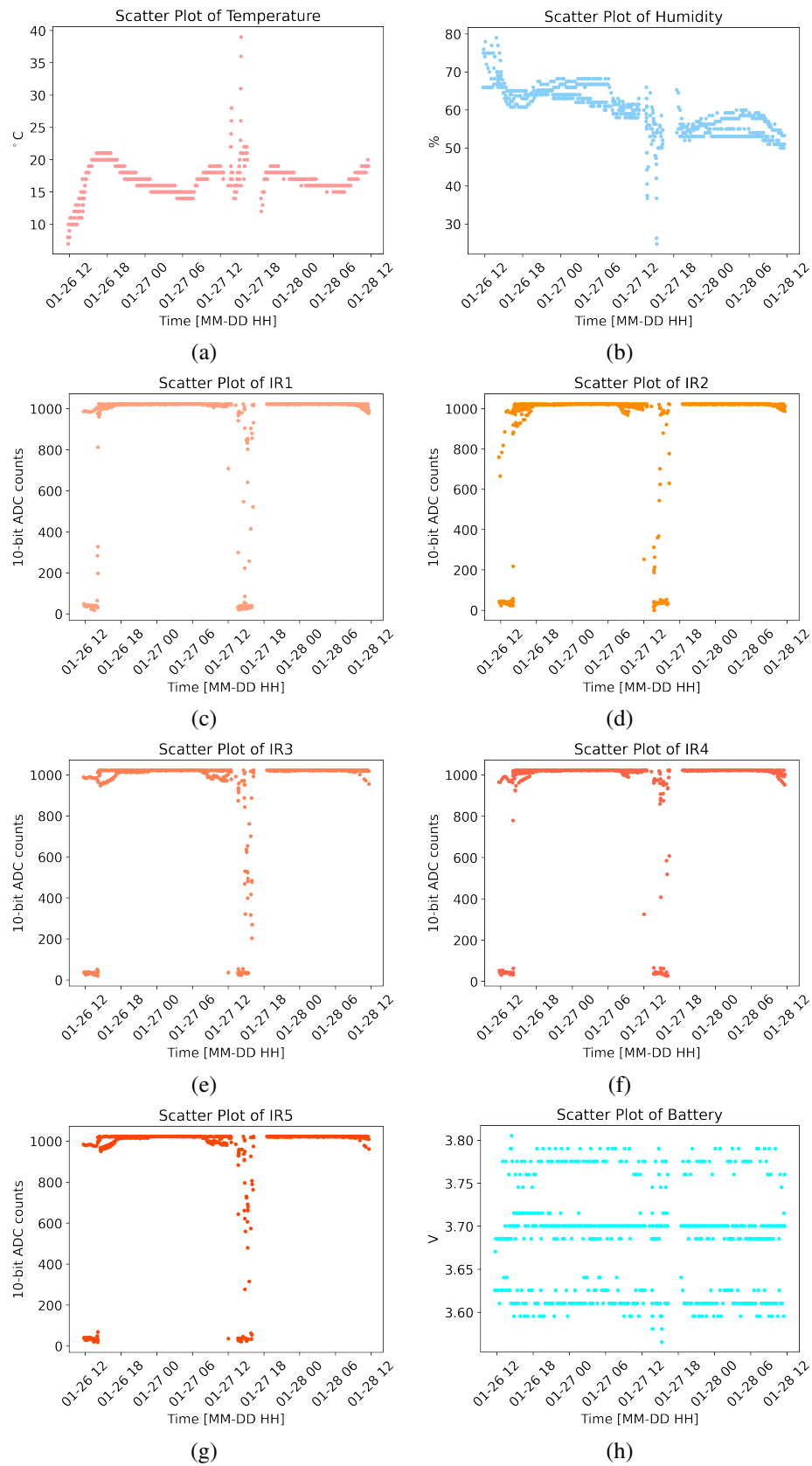


Figure 5.35: Temperature (a), Humidity (b), IR sensors 1-5 (c-g), and Battery (h) in Dataset 4 over the observation period.

The **WSE** conducted on January 27, 2022, lasted approximately 3 h10, starting at 13h10 and ending at 16h20. On January 26, 181 entries were recorded, increasing significantly to 311 on January 27. On January 28, the count decreased to 151 entries. During the dataset's timeframe, a total of 643 entries were collected, distributed across 12 variables: Battery, Humidity, Temperature, IR1 through IR5 sensor readings, a binary **FS** indicator, Node **ID**, Node Version, and time. Notably, the **FS** variable marks the presence or absence of fire, with **True** indicating fire-related periods.

The Battery variable presents 15 unique values, reflecting gradual changes in battery voltage over time. Humidity displays 59 unique values, indicating diverse environmental conditions captured in the dataset. The Temperature variable has 23 unique values. The **IR** Sensors (IR1-IR5) have 93 to 111 unique values. It comprises numerical variables (Battery, Humidity, Temperature, and IR1-IR5), a boolean column (**FS**), and categorical columns (Node **ID** and Node Version). All columns are complete, without null values (as indicated in Figure 5.36), and the total memory usage is approximately 55.9kB.

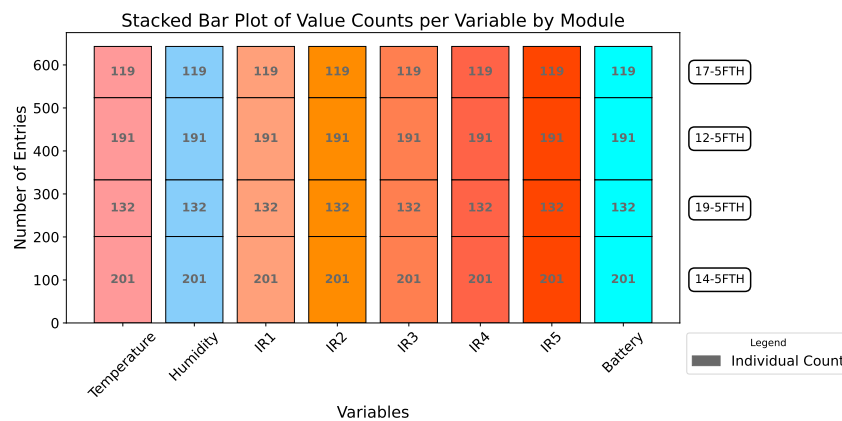


Figure 5.36: Total number of entries for each module in the Dataset 4.

A statistical summary of the Dataset 4 highlights the recorded values' range and mean. Battery voltage varies from 3.57 V to 3.81 V, with a mean of 3.68 V. Humidity ranges from 24.75% to 79.0%, with a mean of 60.01%. Temperature extends from 7.0°C to 39.0°C, with a mean of 17.1°C. The **IR** sensors (IR1-IR5) exhibit values ranging from 18 to 1023 (10-bit ADC count). Their mean values are as follows: 926.89 (IR1), 911.90 (IR2), 926.79 (IR3), 936.16 (IR4), and 927.88 (IR5). Subsequent analyses exclude the battery variable, as its primary purpose is to ensure the operational integrity of the modules.

The skewness values of the numeric variables in Dataset 4 highlight their distribution characteristics. Temperature has a skewness value of 1.42, suggesting a right-skewed distribution, while Humidity presents a skewness of -0.55 , reflecting a slight left skewness. The **IR** sensors (IR1-IR5) exhibit high negative skewness from -2.54 to -3.05 , showing that most values were on the left skewness. In Figure 5.37, the histograms and fitted probability density functions for each variable visually illustrate the data distribution.

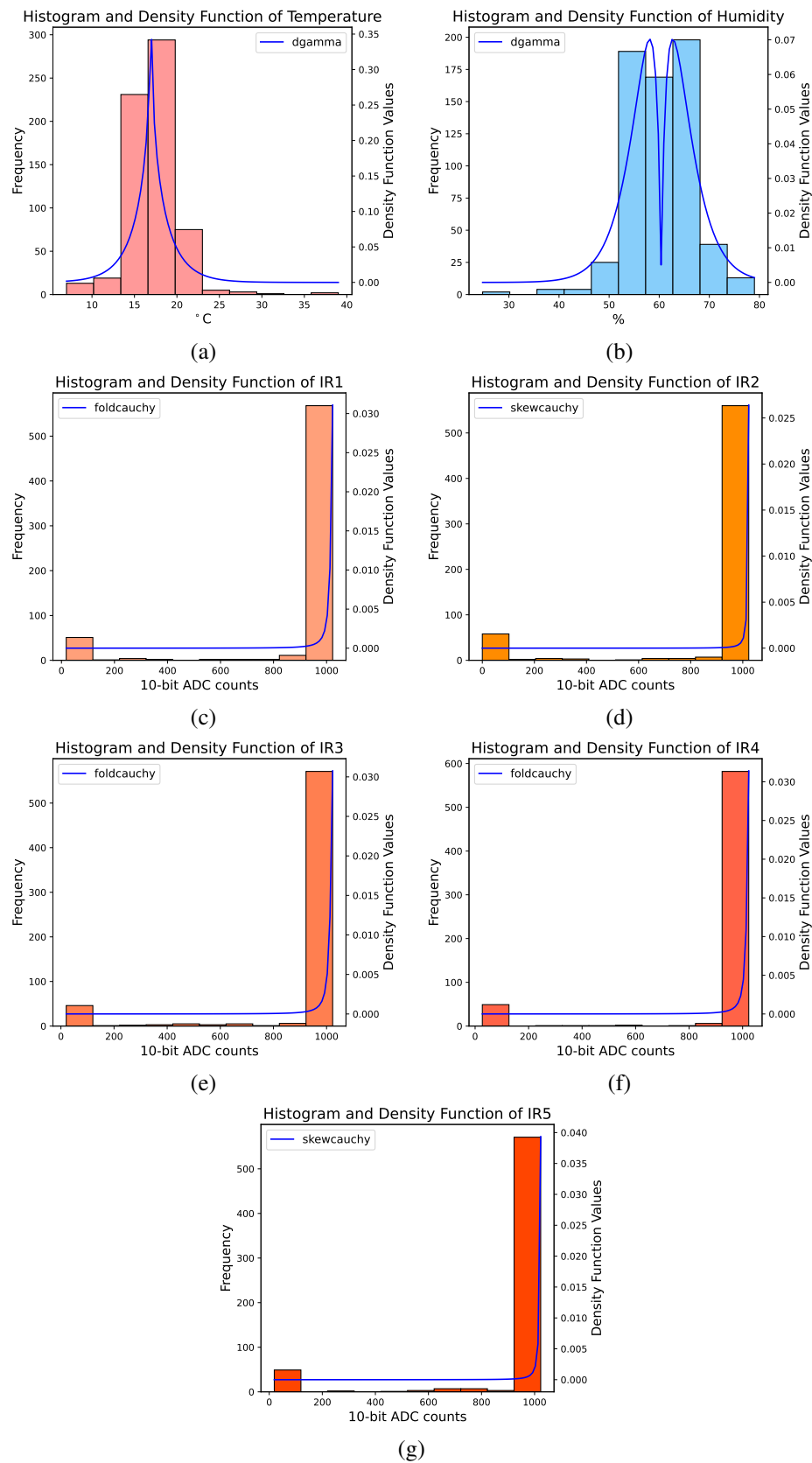
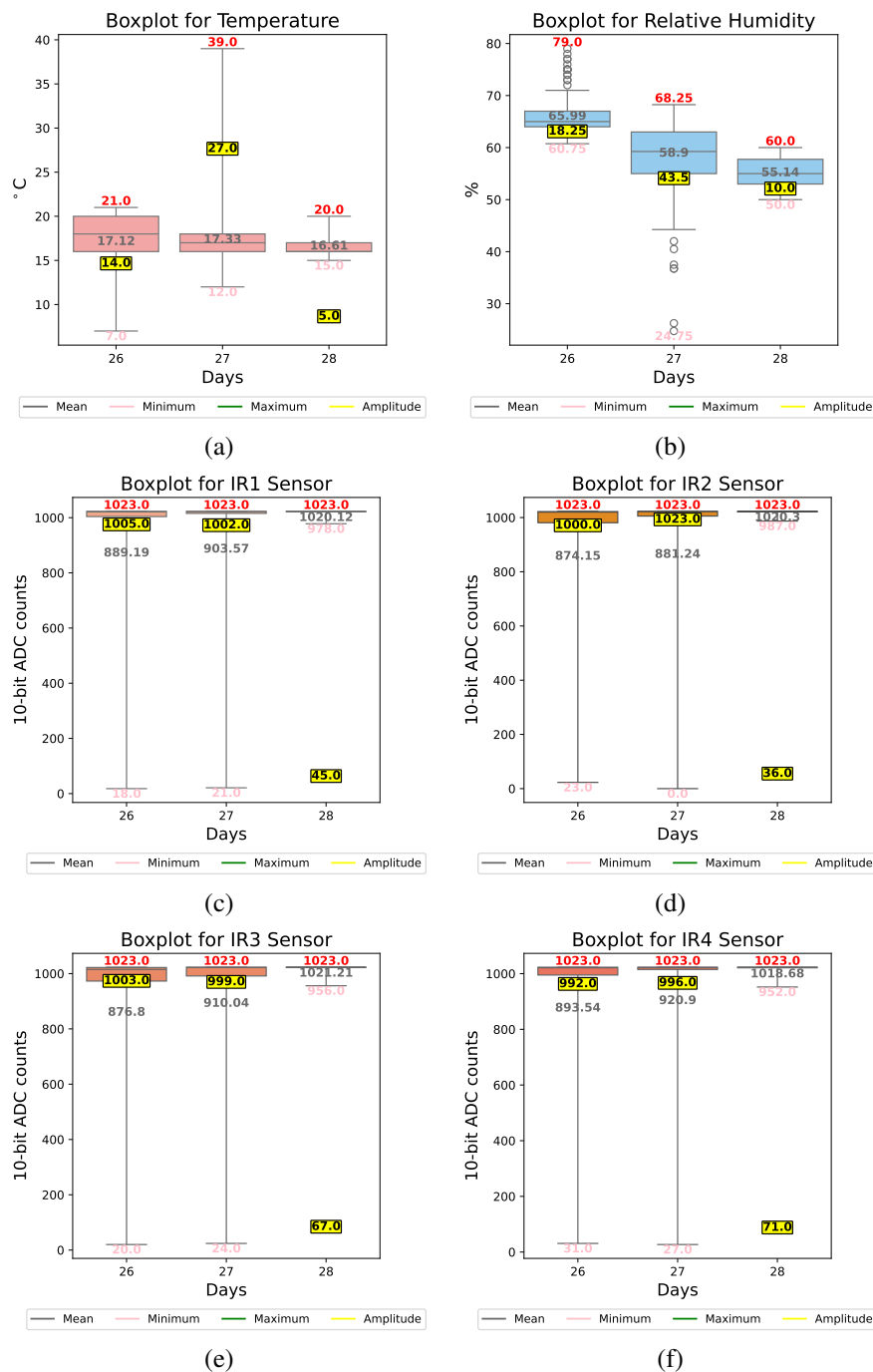


Figure 5.37: Histogram of Temperature (a), Humidity (b), and IR sensors 1-5 (c-g) in Dataset 4.

These distributions emphasize the distinct behaviors of the variables. The Gamma distribution best fits the Temperature variable, effectively capturing its skewness and moderate variability with a shape parameter that controls the tail behavior. Similarly, the same distribution most effectively represents the variability of the Humidity variable, particularly its asymmetric and bounded nature. The Folded Cauchy Distribution best fits IR1, IR3, and IR4, capturing its heavy-tailed nature and skewness. Finally, the Skew Cauchy Distribution better fits IR2 and IR5, modeling their pronounced skewness and extended tails, highlighting irregular environmental variations. Boxplots were generated for all sensor variables, shown in Figure 5.38.



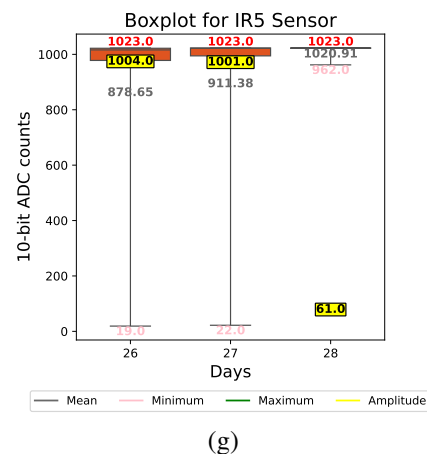


Figure 5.38: Boxplots illustrating the minimum, mean, maximum, and range for each variable (Temperature (a), Humidity (b), and IR sensors 1-5 (c-g)) in Dataset 4.

The boxplot for Temperature variable reveals a significant variability on January 27, coinciding with the **WSE** period. The Humidity variable presents a similar significance, with the most considerable amplitude occurring on January 27, suggesting a notable environmental response during the fire event. Across the **IR** sensors, the box size was influenced during the **WSE** on January 27; along all the plots, the boxes are smaller than the last day. The correlation matrix for Dataset 4 evidences the relationships between the variables. Therefore, Figure 5.39 indicates these relationships, presenting the magnitude of correlations.

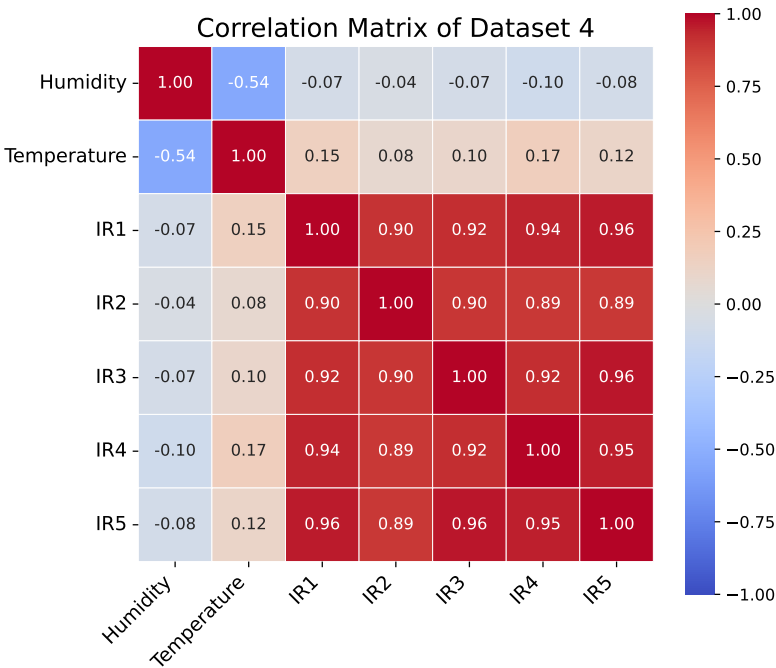


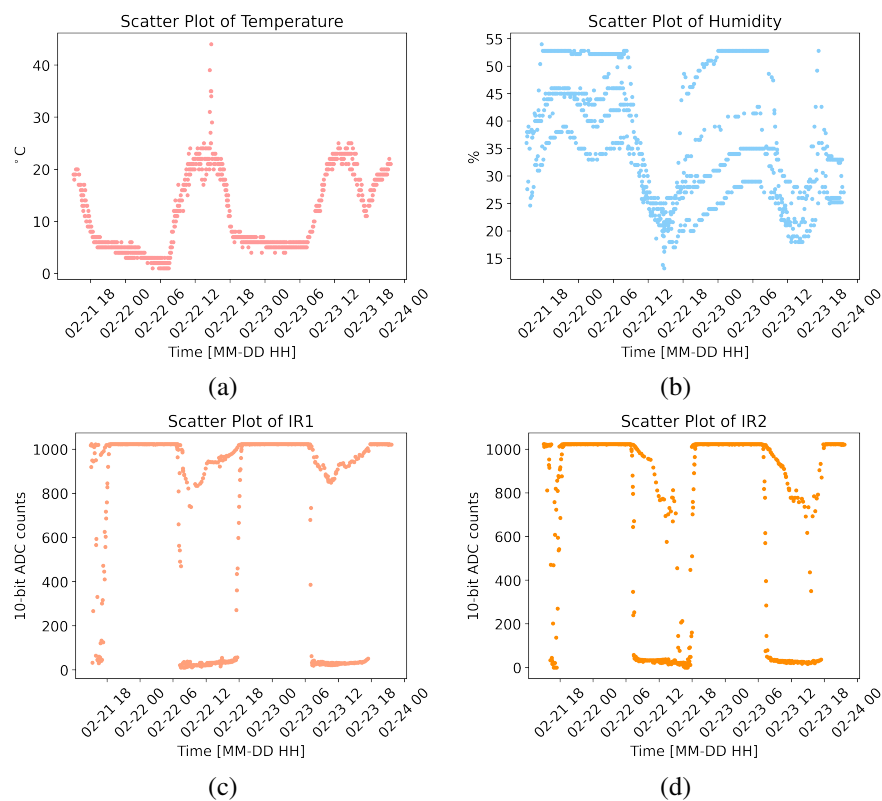
Figure 5.39: Correlation matrix of Dataset 4, illustrating the relationships among Temperature, Humidity, and IR sensor values (IR1-IR5).

The **IR** sensors (IR1 through IR5) are correlated with each other with coefficients ranging from 0.89 to 0.96. This correlation suggests that these sensors consistently capture similar trends in the data acquisition period. Despite Temperature and Humidity exhibiting a positive correlation with the **IR** sensors, the values are close to zero. On the other hand, the correlation between Temperature and Humidity is negative with a value of -0.54 , indicating an inverse behavior in their data patterns.

The **ADF** test can determine the stationarity of the variables in Dataset 4, a crucial property for time series analysis. The test confirms that Temperature, IR1, IR3, IR4, and IR5 are stationary, as their p values are below the significance threshold of 0.05. These variables maintain consistent statistical properties, making them suitable for modeling and forecasting without additional transformations. In contrast, Humidity and IR2 are non-stationary, as their p values exceed the threshold (0.31 and 0.06, respectively).

5.4.5 Dataset 5 - Description and Analysis

Dataset 5 was collected during a controlled and supervised fire test to analyse sensor behavior under real fire conditions. The test was carried out in an area with **TCD**, ranging from 42 to 88 and averaging 68. Data acquisition began on February 21, 2022, at 15h and concluded on February 23, 2022, at 23h59. The **WSN** used in this test consisted of five modules: 05-5FTH, 06-5FTH, 08-5FTH, 10-5FTH, and 14-5FTH. The data collection has the **DROWSIER** filter implementation, which ensures efficient communication by transmitting data every 15 min under regular conditions or upon detecting anomalous events. Figure 5.40 presents the scatter plots showing the recorded variables, allowing for an initial assessment of the dataset's structure.



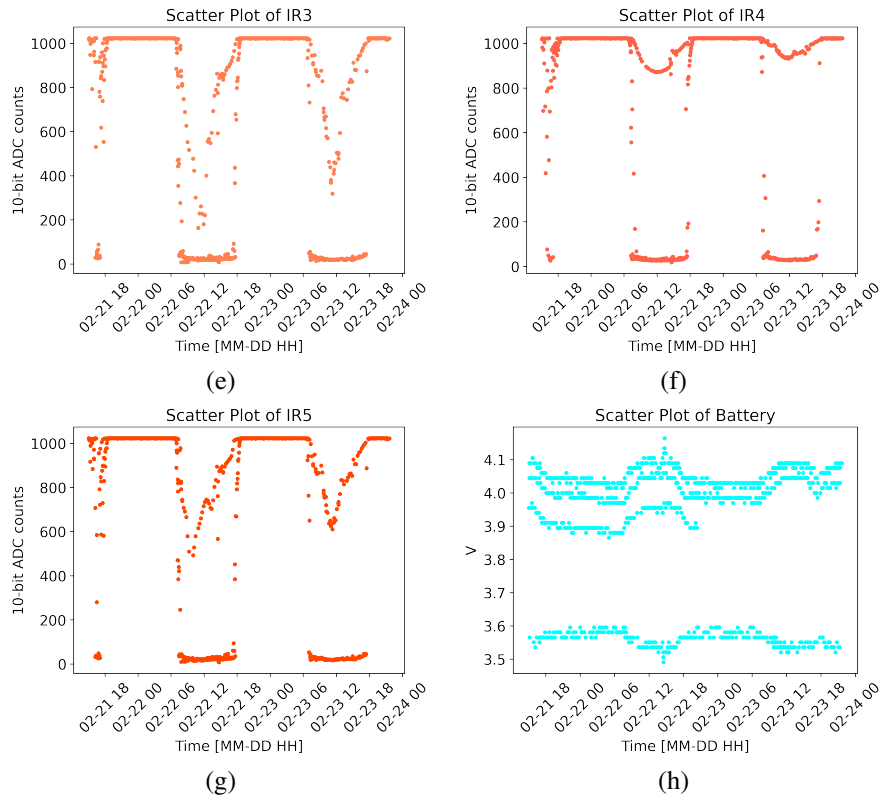


Figure 5.40: Temperature (a), Humidity (b), IR sensors 1-5 (c-g), and Battery (h) in Dataset 5 over the observation period.

The **WSE** for Dataset 5 was on February 22, between 14h05 and 15h55, lasting approximately 1h50. Over the dataset's duration, 864 entries were recorded, covering environmental conditions before, during, and after the fire. On February 21, 150 entries were registered, representing the initial data acquisition phase. This number increased to 406 entries on February 22, and on February 23, the dataset concluded with 308 entries. Dataset 5 consists of the following variables: Time, Battery, Humidity, Temperature, IR1 through IR5, **FS**, Node **ID**, and Node Version.

The Battery variable exhibits 28 unique values, while Humidity and Temperature show 81 and 32 unique values, respectively. The **IR** sensor readings also present diversity, with IR1 having 187 unique values, IR2 with 184, IR3 with 201, IR4 with 153, and IR5 with 177 unique values. Moreover, the dataset is composed of numerical variables (Battery, Humidity, Temperature, and IR1-IR5), a boolean column (**FS**), and categorical columns (Node **ID** and Node Version). No one columns have null values, indicated by Figure 5.41, and the total memory usage is approximately 75.1 kB.

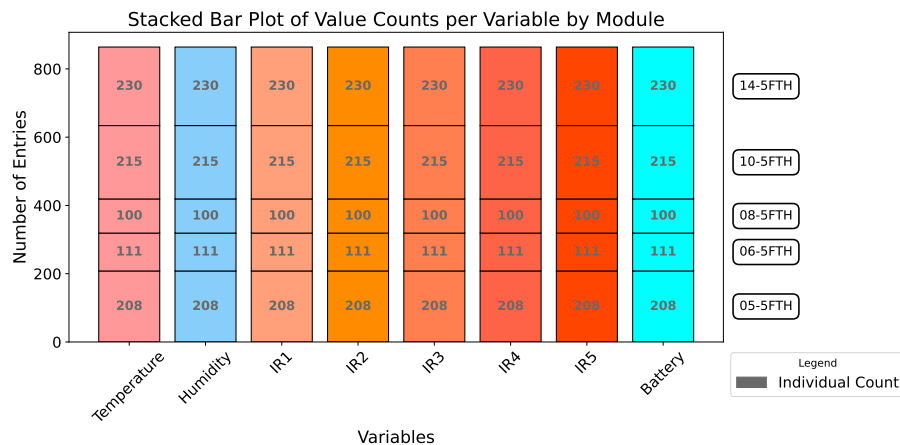
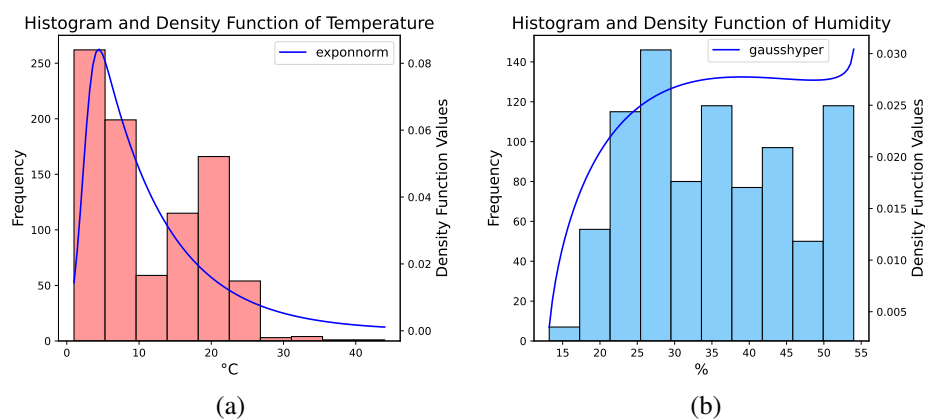


Figure 5.41: Total number of entries for each module in the Dataset 5.

The Battery variable ranges between 3.49 V and 4.16 V, with a mean of 3.89 V. Humidity values extend from 13.2% to 54%, with a mean of 35.23%. Temperature exhibits a range from 1 °C to 44 °C and a mean of 11.48 °C. IR1-IR5 show variability consistent, with values between 0 and 1023 (10-bit ADC count).

Given the consistent nature and lower variability of the voltage values across the dataset, the Battery variable will be excluded from subsequent analyses. This decision aligns with the approach taken in previous datasets, ensuring a focus on variables directly related to environmental conditions and fire detection.

The skewness analysis of Dataset 5 reveals distinct patterns in the distributions of the numerical variables. Temperature and Humidity exhibit positive skewness (0.57 and 0.23, respectively), indicating right skewness distributions, where lower values are more common and higher values are less frequent. On the other hand, the IR sensor variables (IR1 to IR5) presented negative skewness ranging from -0.74 to -0.93 , reflecting left skewness distributions. The histograms in Figure 5.42 from Dataset 5 provide an overview of the distributions for each variable, paired with their best-fit probability density functions.



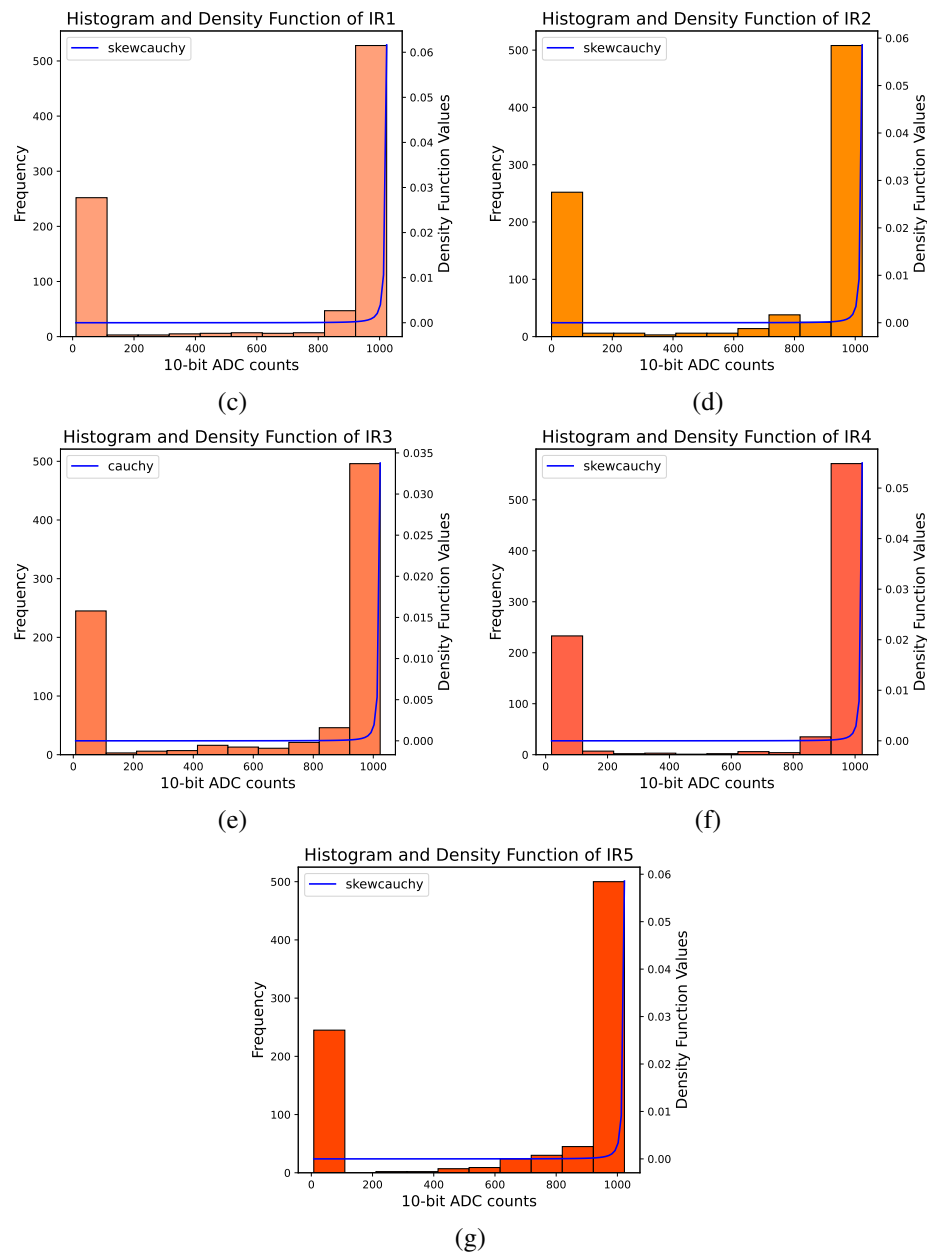
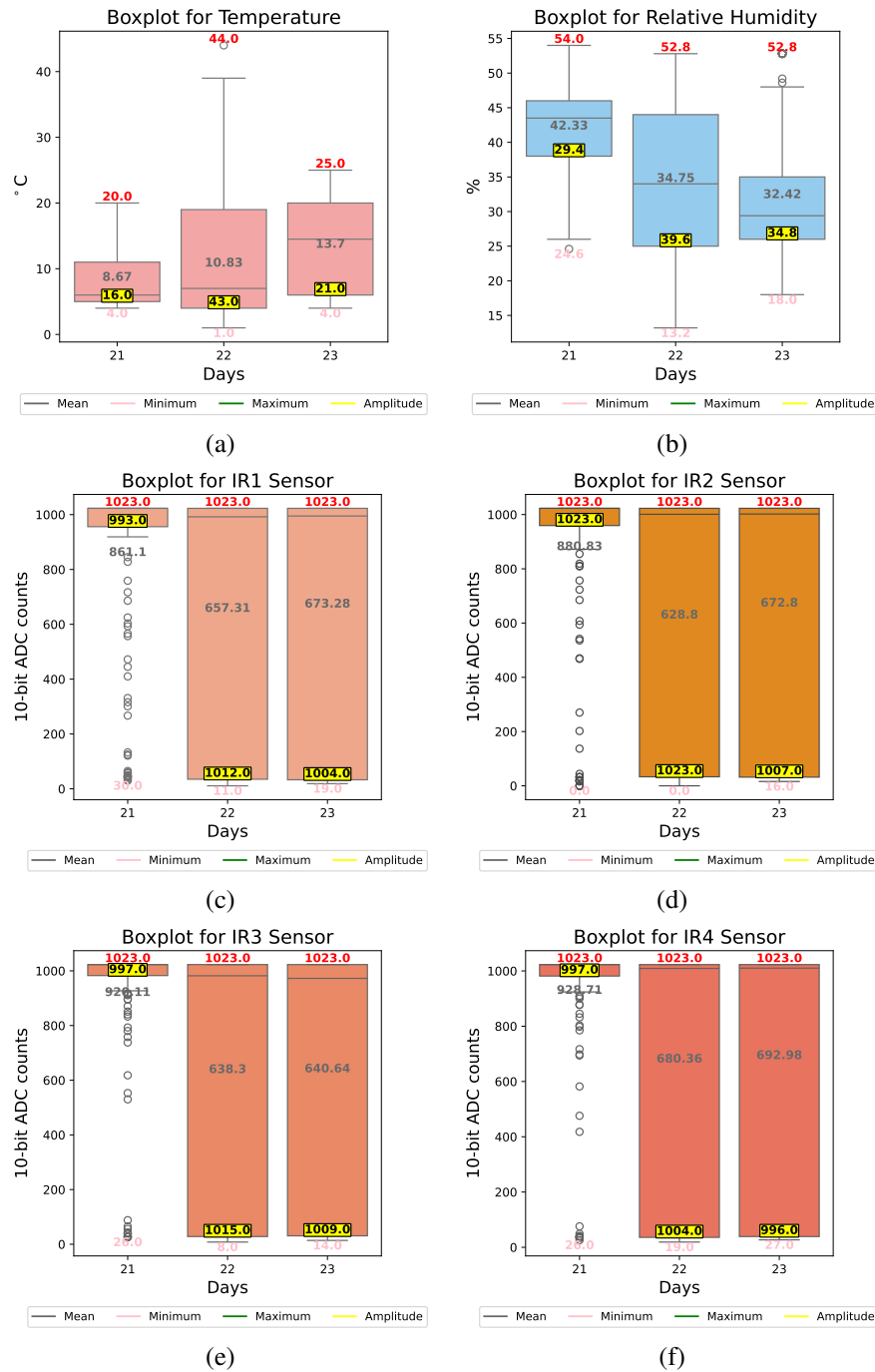


Figure 5.42: Histogram of Temperature (a), Humidity (b), and IR sensors 1-5 (c-g) in Dataset 5, which illustrate the data distribution and frequency over the observation period.

The Temperature variable follows the Exponential Normal distribution, characterized by positive skewness, moderate variability, and a tendency to concentrate around the mean while exhibiting a longer right tail. The Humidity variable aligns well with the Generalized Gaussian Hypergeometric distribution, which effectively captures its bounded range and potential asymmetry. This distribution's flexibility enables it to model the variable's behavior, including skewness and moderate tails observed in the data. The IR sensors (IR1-IR5) show heavy-tailed and skewed behavior, with four of them following the Skew Cauchy distribution and one fitting the Cauchy distribution.

Boxplots were generated for all numerical variables in Dataset 5, including Temperature,

Humidity, and IR sensor readings (IR1-IR5). These visual representations (Figure 5.43) highlight the data distribution, central tendencies, and potential outliers across different data collection days. By examining the **IQR**, median, and amplitude for each variable, variations and potential anomalies can be identified, particularly during the fire event on February 22.



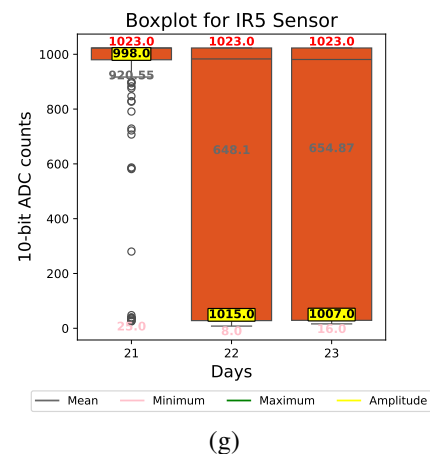


Figure 5.43: Boxplots illustrating the minimum, mean, maximum, and range for each variable (Temperature (a), Humidity (b), and IR sensors 1-5 (c-g)) in Dataset 5

The boxplot analysis highlights significant changes on February 22 due to the **WSE**. A wide range of values is observed for Temperature, with a maximum reaching 44 °C and a notable increase in the median compared to other days, reflecting the fire's influence. Humidity remains relatively stable across the dataset, but February 22 shows a slight decrease in the median and a narrower **IQR**. The analysis of the **IR** sensors (IR1-IR5) reveals a larger amplitude of the readings, indicating a broader range of sensor responses influenced by the **WSE**. The correlation matrix in Figure 5.44 presents the correlation values of the variables in the dataset.

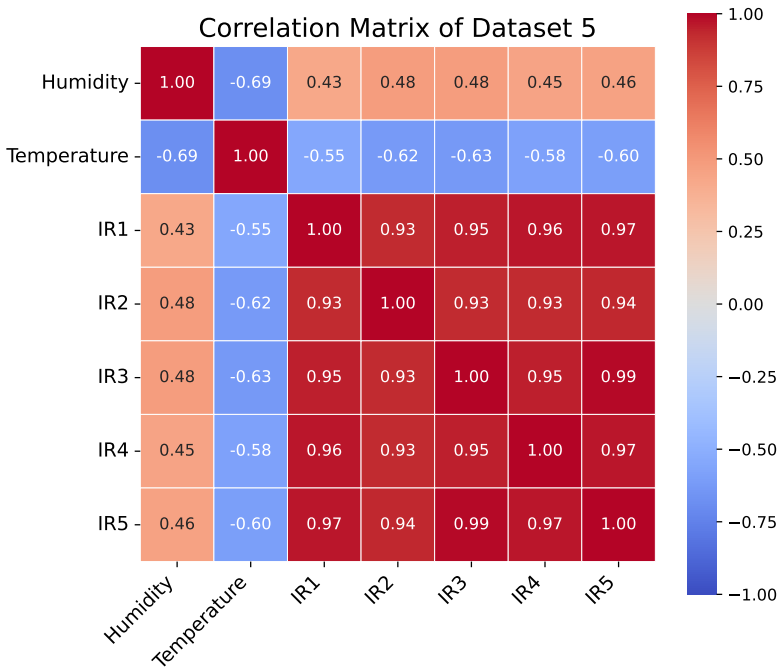


Figure 5.44: Correlation matrix of Dataset 5, illustrating the relationships among Temperature, Humidity, and IR sensor values (IR1-IR5).

Humidity shows a negative correlation with Temperature, with a -0.69 value, which is consistent with typical atmospheric dynamics where temperature and humidity often exhibit an inverse relationship. Conversely, the **IR** sensors (IR1-IR5) present positive correlations among themselves, ranging from 0.93 to 0.99, indicating that their readings are closely aligned, potentially due to the sensors' uniform response to fire and other shared environmental factors.

The results of the **ADF** test indicate that all analysed variables in Dataset 5 are likely non-stationary. The reason is that the p values do not align with the criteria, which is the threshold of 0.05. The p values for each variable are as follows: Temperature is 0.43, Humidity is 0.63, IR1 is 0.34, IR2 is 0.13, IR3 is 0.37, IR4 is 0.26, and IR5 is 0.16.

5.5 Data Processing

The data processing phase is a critical step in preparing the datasets for analysis and modeling, ensuring data quality, consistency, and structure. This section outlines the systematic approach employed to process the datasets, addressing issues such as data inconsistencies, transmission irregularities, and the high dimensionality of the data. The data processing procedures were designed to maintain data integrity while enabling effective analysis, particularly for identifying environmental anomalies and fire events.

The first step focuses on the Comparative Analysis of Module Versions, which evaluates the performance of 5FTH module versions 1.0 and 2.0 under identical environmental conditions. By creating a subset of Dataset 2 (named as Set 1-2), the statistical characteristics of the sensor variables were compared across module versions. Standardization techniques were applied to balance the number of entries between groups, ensuring fair comparisons. Correlation analyses were conducted for each group and during specific events, such as the **WSE**, to assess consistency in sensor responses.

The **DROWSIER** Filter was introduced as a novel data-filtering algorithm to reduce network bandwidth usage and conserve battery life in the **WSN**. Four algorithms: Exponential Smoothing, Simple Moving Average, Double Moving Average, and Least Squares Strategy were developed and integrated by focusing on significant sensor data variations. These algorithms were optimized using Dataset 2 to detect critical environmental changes while minimizing redundant data transmissions. Combining these algorithms in the **DROWSIER** Filter provided a robust solution for real-time wildfire monitoring, balancing responsiveness, and energy efficiency avoiding redundant transmissions.

Finally, Time Series Resampling was conducted to address irregular data acquisition intervals and ensure consistent time-series data. Resampling methods were employed for both datasets to standardize data intervals, reducing noise while preserving trends. Before and after resampling, stationarity checks revealed improvements for variables, particularly for Humidity and **IR** sensors, which became more suitable for time-series modeling.

The data processing steps in this section provide a foundation for subsequent analyses, including the development and application of **ML** Models described in the next section. By addressing data

inconsistencies, optimizing feature extraction, and enhancing temporal stability, these procedures ensure that the datasets are prepared for reliable and meaningful understanding.

5.5.1 Comparative Analysis of Module Versions

In this section, the data collected by 5FTH 1.0 and 5FTH 2.0 of the 5FTH modules, deployed simultaneously within the network, are analysed to assess performance consistency and potential improvements offered by the updated version. Each sensor captured continuous readings with and without WSE (WSE4) during Dataset 2 acquisition. Therefore, based on the data, it is possible to determine which version can be used on a large scale and if they are equivalent. To achieve this, a short EDA is planned to compare the data behaviors regarding statistics and equality.

To enable direct comparison between the two module versions under identical data collection conditions, a subset of Dataset 2 was created and named Set 1-2. Therefore, Set 1-2 includes the intersection data (represented by Figure 5.45) recorded when both module versions were installed and operational within the same environment. In this sense, from Table 5.3, the period under analysis comprises from May 25, 2021, at 16h15 to May 30, 2021, at 10h05, during which both versions were installed and operational in the same environment, ensuring comparable conditions across all modules. Then, a descriptive analysis of Set 1-2 provides an overview of the primary statistics for each sensor variable and battery level.

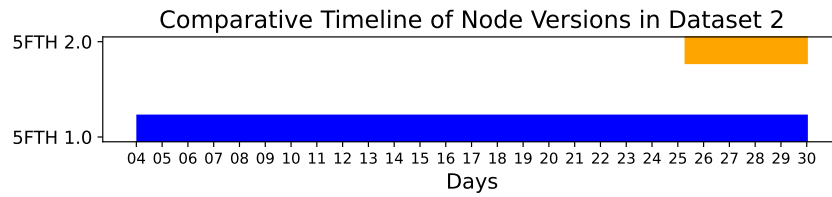


Figure 5.45: Timeline of overlapping data collection for both module versions in Dataset 2.

The Set 1-2 records 44086 entries, with an average battery voltage of approximately 3.70 V and a standard deviation of 0.24 V. The minimum battery voltage observed is 2.95 V, and the maximum is 4.07 V. The first quartile ($Q1$) is 3.49 V, the median ($Q2$) is 3.66 V, and the third quartile ($Q3$) is 3.96 V. The average humidity level is 57%, with a standard deviation of 17.93%. The minimum recorded humidity is 26%, while the maximum is 93%. The quartile values are $Q1$ at 41%, $Q2$ at 53%, and $Q3$ at 71%. Temperature values range between 11 °C and 31 °C, with an average of approximately 19.51 °C and a standard deviation of 4.03 °C. The first quartile is 16 °C, the median is 19 °C, and the third quartile is 23 °C.

For IR1 and IR2, the mean values are approximately 821.09 and 823.24, respectively, with standard deviations around 352.51 and 354.14. Both sensors have a minimum reading of 0 and a maximum of 1023, with median values close to 1011 and upper quartiles at 1023. IR3 and IR4 also exhibit similar behavior, with means around 817.62 and 818.59 and standard deviations of 361.41 and 363.77. IR3 and IR4 have a maximum reading of 1023 and median values around 1011. IR5

has a mean of 870.53 with a standard deviation of 305.36. The values for IR5 range from 42 to 1023, with quartiles at $Q1 = 951$, $Q2 = 1015$, and $Q3 = 1023$.

The Set 1-2 was divided into two groups based on the `Node Version` column, which labels the module version with “1.0” or “2.0”. The first group, Group 1, contains modules classified with “1.0”, for which ten modules were selected, namely, from 01-5FTH to 10-5FTH. The Group 2 consists of three modules labeled as “2.0”, and they are identified as 11-5FTH, 14-5FTH, and 16-5FTH. The descriptive statistics for each group reveal variations between the two versions across the sensor readings:

- Group 1 has 32106 rows, the average Battery level is 3.65 V, with a standard deviation of 0.25 V, and readings range between 2.95 V and 4.05 V. Humidity ranging from 26% to 93%, displays an average of 63.13% and the standard deviation is 17.26%, The mean Temperature value is approximately 19.58°C, with values from 12°C to 31°C, and the standard deviation is 3.98 V. Among the IR sensors, IR1, IR2, and IR3 averaged between 744 and 753, and IR5 showed a slightly higher mean of 818.85. The IR’s standard deviations are 390.66, 392.41, 400.61, 401.75, and 343.56, respectively, from IR1 to IR5.
- The amount of entries is 11980 in Group 2. Battery level has a mean of 3.83 V, with a standard deviation of 0.15 V and values ranging from 3.61 V to 4.07 V. Humidity in this group presents an average of 40.81%, with a range from 32% to 54%, and a standard deviation of 4.71%. Temperature averages around 19.34°C with a standard deviation of 4.13°C, and this parameter ranges from 11°C to 28°C. The IR sensors reached mean values of 1006.19 (IR1), 1010.31 (IR2), 1008.67 (IR3), 1017.57 (IR4), 1009.04 (IR5), and the standard deviations from IR1 to IR5 are 35.14, 27.83, 20.53, 6.86, 21.21, respectively.

Notably, the number of entries between Group 1 and Group 2 differs, mainly due to their different density in terms of modules. Moreover, variations in transmission frequencies may introduce inconsistencies that could bias the comparison analysis. In this sense, a standardization is necessary to balance the group’s entries. To address this, a uniform resampling approach was applied to average readings within 120-second intervals in each group. After resampling, any null values arising from missing data points were filled with the previous available value. This standardization resulted in a uniform row count of 3414 entries for both groups.

A correlation analysis between the variables of each group was conducted to assess the interrelationships within Group 1 and Group 2. The correlation values for each group are presented in Figure 5.46. The variable Battery was excluded from the analysis due to inconsistencies arising from the activation times of the modules in each group.

Humidity and Temperature variables exhibit a positive correlation, indicating that both groups respond similarly to environmental changes despite minor discrepancies. For IR sensors (IR1 through IR5), positive correlations are observed also but show variability across groups. This discrepancy could be due to the geographical placements of modules, which means that each group may be exposed to different sunlight angles, shadows, or physical obstructions, affecting the IR sensors’ readings.

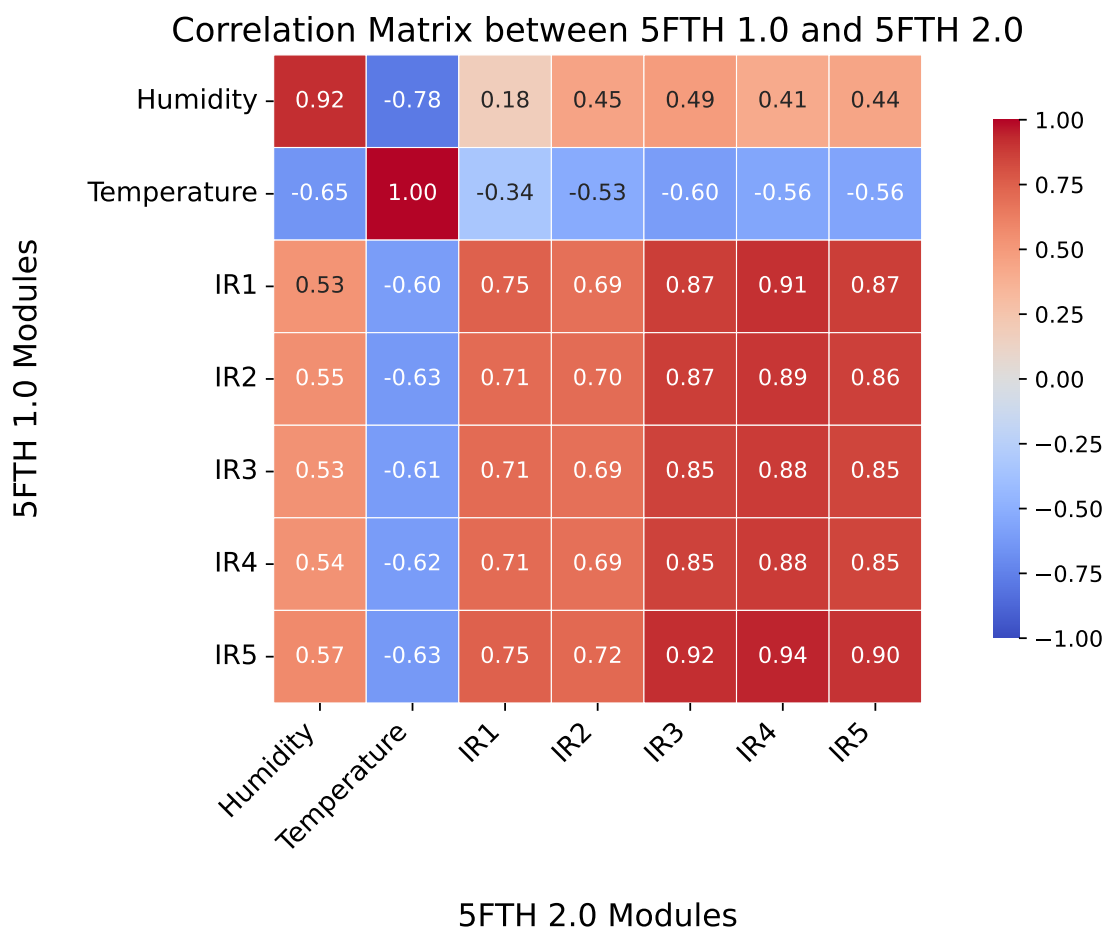


Figure 5.46: Correlation matrix for Group 1 and Group 2, displaying the relationships between Battery, Humidity, Temperature, and the five IR sensor readings.

The 25th May is a crucial day in the Set 1-2 because it includes a **WSE** with both module groups activated. A correlation analysis was performed on the data from this day to check the relationship between the two groups. Therefore, this approach provides information about the modules from each version during a **WSE**. Figure 5.47 illustrates the resulting correlation matrix, reflecting the interaction and alignment of sensors' readings between modules' versions on this day.

On 25th May, the correlation analysis revealed relationships between the two groups' Temperature, Humidity, and **IR** readings. Temperature and Humidity data show higher values between the groups on this day, due to both groups experiencing the same environmental conditions during the **WSE**. This intense correlation indicates that, under identical conditions, both module versions capture these environmental parameters consistently.

For **IR** sensor readings, the correlation also intensifies on 25th May compared to the overall dataset. This suggests that the **IR** sensors across both groups responded similarly during the **WSE**, despite their geographic positioning in the entire dataset. This day-specific analysis highlights how synchronized and event-driven conditions can unify group sensor responses.

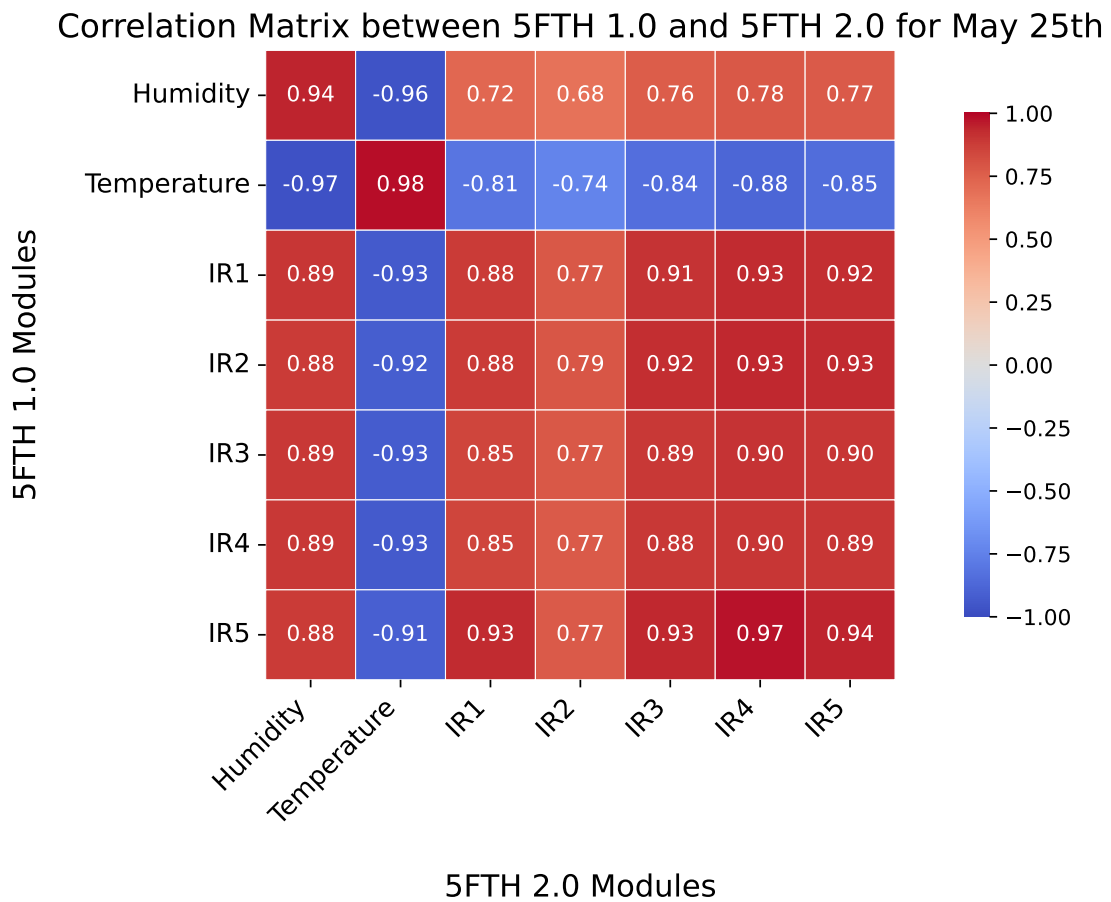


Figure 5.47: Correlation matrix for Group 1 and Group 2, displaying the relationships between all variables during WSE.

In addition to correlation analysis, the **Two One-Sided Tests (TOST)** procedure was employed to rigorously assess the equivalence of mean values between the two groups for each variable over the entire data period. While correlation examines relationships and similarities in behavior, it does not confirm statistical equivalence in the distributions of the two groups. The **TOST** can determine if the means of the groups are statistically equivalent within a specified bound, providing a more precise measure of comparability across the modules.

In the **TOST** procedure, a margin (bound) was set equal to one standard deviation of the values in the primary group for each variable. The test involves calculating two t-statistics: one for the upper bound and one for the lower bound, assessing whether both fall within the critical range, thereby rejecting the null hypothesis of non-equivalence. For each variable, if both t-statistics have the criteria, the groups are considered statistically equivalent within the chosen margin.

For the analysis on 25th May, when both versions were operational during **WSE**, the **TOST** was applied to determine mean equivalence across the two groups for each variable. The **TOST** results for this specific day are as follows:

- **Humidity:** With a bound of approximately $\pm 14.02\%$, the test successfully rejected both null

hypotheses, confirming mean equivalence for humidity readings. This outcome suggests that both module versions captured similar humidity levels during the fire event, providing consistent measurements under comparable environmental conditions.

- **Temperature:** With a bound of $\pm 4.21^\circ\text{C}$, the test also rejected both null hypotheses, indicating equivalence in the temperature measurements across the two groups. This finding reflects that the modules responded similarly to temperature changes during the fire event.
- **IR Sensors (IR1 through IR5):** For all **IR** sensors, with bounds ranging from ± 251.31 to ± 316.78 , both null hypotheses were rejected. This shows mean equivalence across the **IR** readings between the two groups, indicating that the sensors recorded similar **IR** intensities during **WSE**, despite potential differences in their geographical placements.

The comparative analysis between 5FTH 1.0 and 5FTH 2.0 provides an understanding of their performance and suitability for large-scale deployment. The correlation matrix for the entire Set 1-2 reveals that correlations in Humidity and Temperature suggest that both versions of the modules consistently respond to environmental changes.

The **WSE**-specific analysis performed on 25th May offers further clarity. During this day, the correlation between sensor readings (specifically Humidity, Temperature, and **IR** sensors) between the module versions was increased, suggesting that both versions respond similarly under identical environmental conditions. This consistency, particularly in **IR** sensor readings, indicates that the modules' response during fire events is robust and largely unaffected by potential environmental obstructions or geographic placement. The heightened alignment of readings across groups during 25th May points to a unified response from the sensors when exposed to synchronized conditions, such as a fire events.

Furthermore, the **TOST** procedure used to assess statistical equivalence between the module versions confirms that most variables show mean equivalence. Specifically, statistical equivalence was observed in Humidity, Temperature, and all **IR** sensors. This finding supports the idea that both module versions can reliably capture these parameters, making them interchangeable for environmental monitoring.

Considering the large-scale deployment, both module versions capture critical environmental parameters effectively and indicated alignment in their response to fire events. This consistency, especially during the **WSE**, suggests that either version could be deployed under similar environmental conditions with confidence in their performance. However, with 5FTH 2.0 offering a more robust enclosure that enhances durability against environmental factors such as humidity and temperature fluctuations, it holds a distinct advantage over 5FTH 1.0. Consequently, 5FTH 2.0 is the preferred choice for future large-scale installations and will also be used in the final testing of the **ML** models described in the following subsections.

5.5.2 DROWSIER Filter

The **WSN** used to make the Dataset 2 generates a substantial volume of data due to modules' time interval 60s. Additionally, the data was collected over a period of 27 days, resulting in a comprehensive dataset for analysis. While beneficial for real-time monitoring, this high-frequency data transmission introduces significant challenges, particularly in energy consumption. With each transmission, there is a peak in battery usage that directly affects the longevity of the 5FTH modules. Since these modules are often planned to be placed in difficult-to-access locations, frequent maintenance – especially replacing or recharging batteries – becomes costly and time-consuming. In this sense, this subsection is focused on summarizing the development of the **DROWSIER** filter, which is fully described in [30].

In addition to the energy implications, a high-resolution time interval for data transmission extensively uses the network's bandwidth. This impact is especially relevant for networks operating on **LoRaWAN** technology, which, although designed for low-power operations, must manage data efficiently to prevent bandwidth congestion. Without an effective data filtering mechanism, the constant flow of messages may lead to redundant transmissions that do not contribute valuable information to the monitoring system, thereby wasting bandwidth and battery life. Consequently, this situation requires a pre-filtering algorithm that can intelligently reduce the volume of data transmitted by the **WSN**, focusing on only the most critical information required to detect and alert for fire ignitions and other significant anomalies.

A set of algorithms was combined to develop a filter message before transmission to address the challenges of high-frequency data transmission and reduce unnecessary network traffic. These algorithms aim to detect only significant changes in sensor readings, thus conserving bandwidth and battery power by preventing redundant data from being sent. Therefore, based on the Dataset 2, four main approaches were implemented, each with a unique method of filtering data and identifying meaningful variations. Each of these algorithms is lightweight and computationally efficient, designed to run on the constrained hardware of the sensor modules

5.5.2.1 Exponential Smoothing Algorithm (A1)

This algorithm applies exponential smoothing to prioritize recent data, giving them more weight and allowing for faster detection of sudden changes. Exponential smoothing works effectively to smooth out random fluctuations while capturing abrupt trends, probably indicating critical events such as fire ignitions. By adjusting the smoothing factor, the algorithm filters out non-significant variations, ensuring that only essential data triggers message transmission. Equation (5.1) represents the Exponential Smoothing Algorithm (A1) applied to message filtering.

$$Y_t = \omega X_t + (1 - \omega)X_{t-1}, t \geq 0 \quad (5.1)$$

where Y_t is the output after applying an exponential smoothing filter, t is the time index, ω is a weight variable with $0 < \omega < 1$, X_t represents the actual data at time t , and X_{t-1} is previous data given a t .

In this sense, A1 compares the difference between the previously read values and the actual values. The difference between successive timestamps can be given by Equation (5.2):

$$\Delta_{A1} = |Y_t - X_t| \quad (5.2)$$

The difference between these data is calibrated through adjustments in the weight variable ω . By adjusting ω , each module microcontroller can compare the previously stored data with the actual without uploading them from the server. Then, each module can trigger a message to the server with the sensor values only when an anomaly is detected.

5.5.2.2 Simple Moving Average (A2)

This approach uses a **MA** calculation to smooth the data over a short window of previous values. By averaging a given data window, A2 filters out minor fluctuations and focuses on identifying trends over a defined period. When current readings significantly deviate from this **MA**, the algorithm flags the data as anomalous, prompting a message transmission. This technique effectively balances the need for timely data reporting with energy efficiency. Given a set of n samples X_t , that represents the value collected by the sensor at the time t ; the current **MA** is expressed by Equation (5.3):

$$SMA_t = \frac{1}{n} \sum_{i=1}^n X_{t-i} \quad (5.3)$$

In this case, a range of n samples is considered to evaluate a windows function that finds fire ignitions using a **MA**. The following equation (Equation (5.4)) defines the associated error:

$$\Delta_{A2} = |SMA_t - X_t| \quad (5.4)$$

hence, the difference between actual data with a simple **MA** is considered to evaluate the presence of fire in the monitored region, as presented by the following condition. If $\Delta_{A2} > D$ (with $D > 0$), it could indicate the possibility of fire ignitions near a module. Thereupon, the data should be immediately transmitted to the cloud system. On the other hand, if the $\Delta_{A2} \leq D$, the module will not send the message and consequently, it will help to reduce the volume of messages and save energy.

5.5.2.3 Double Moving Average (A3)

A variation of the simple **MA**, this algorithm compares two **MA**s calculated over different time windows. By analysing the difference between these two averages, A3 can more precisely identify rapid shifts in sensor values, as it captures both short-term and longer-term trends. This layered approach enhances anomaly detection accuracy and reduces the likelihood of false alerts caused by minor fluctuations.

In this sense, to capture both short-term and long-term trends, the algorithm calculates two **MA** over different time windows. The first **MA**, $SMA_t^{(n_1)}$, considers the last n_1 sensor readings,

excluding the current reading (X_0). The second **MA**, $SMA_t^{(n_2)}$, uses the last n_2 readings, also excluding X_0 . By comparing these two **MA**s, as shown in Equation (5.5), the algorithm identifies rapid shifts in the sensor values, indicating potential anomalies.

$$\Delta_{A3} = \left| SMA_t^{(n_1)} - SMA_t^{(n_2)} \right| \quad (5.5)$$

To ensure meaningful results, it is necessary that $n_1 \neq n_2$ and that both n_1 and n_2 are positive integers ($n_1, n_2 \in \mathbb{N}^*$). If $n_1 = n_2$, the difference Δ_{A3} would always be zero, rendering the algorithm ineffective for anomaly detection. Non-integer or non-positive values for n_1 and n_2 are not valid, as they do not correspond to a realistic window size.

5.5.2.4 Least Squares Strategy (A4)

The least squares algorithm is designed to predict sensor readings based on a simple **LR** model. Using the slope and intercept calculated from recent data points, A4 projects an expected value for the next reading. If the actual sensor data deviates significantly from this predicted value, it triggers an alert. This predictive approach is practical for recognizing trends and detecting anomalies that follow a gradual pattern.

$$\begin{aligned} \alpha &= \bar{Y} - \beta \bar{X}, \\ \beta &= \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sum_{i=1}^n (X_i - \bar{X})^2}, \\ Y_{t+1} &= \alpha + \beta X_{t+1}. \end{aligned} \quad (5.6)$$

It's possible to indicate anomalies by considering three previous samples to define a linear equation that minimizes the sum of the squares between the samples and the linear function values. In this way, the linear function represented by Equation (5.6) is used to predict the next sample (in this case, the 4th sample). In addition, this value is compared with the actual value (X_{t+1}) collected at the time $t + 1$.

When the error between the predicted value Y_{t+1} and the actual value collected by the sensor, X_{t+1} , is bigger than D , as presented in Equation (5.7), the data should be immediately transmitted to the cloud, due to fire suspicion.

$$\Delta_{A4} = |Y_{t+1} - X_{t+1}| \quad (5.7)$$

The threshold D can be defined based on historical data or sensor sensitivity to minimize false positives while ensuring fire detection.

5.5.2.5 Combining Algorithms in the DROWSIER Filter

The **DROWSIER** filter can be made by fine-tuning each algorithm's parameters and combining their outputs. This combination approach leverages the strengths of each algorithm since the combination

process uses a set of optimized parameters for each algorithm. After testing a variety of values using the Dataset 2, the following variables were identified as optimal for each approach:

For Algorithm A1, the best parameter for ω is 0.3, and for D is 0.1. With these settings, the algorithm accurately identified potential fire ignitions in controlled testing environments. Next, the best parameter for A2 is $n = 6$ and $D = 0.0$. Setting the window size to 6 samples allows A2 to smooth the data effectively, reducing minor noise while remaining responsive to shifts indicative of fire. A3 performs best when the parameters $n_1 = 5$ and $n_2 = 6$ with a threshold of $D = 0.0$. The last algorithm, A4, indicates anomalies when D is 0.0 because this setting captures gradual but impactful trends that can precede sudden environmental changes, such as those caused by fire ignitions. Moreover, the best parameters mentioned were found when the interval was defined as 15 min.

The **DROWSIER** filter combines these algorithms in their best configurations to maximize the detection accuracy of fire-related anomalies while keeping false alerts to a minimum. Therefore, each algorithm contributes a distinct layer of analysis: A1 provides quick responses to immediate changes; A2 and A3 add stability by smoothing over short and long-term fluctuations; A4 aids in detecting subtle but meaningful trends over time.

Computational tests were conducted to evaluate the combinations of outputs from each algorithm with each variable, and the best combination of the described approaches indicates that **DROWSIER** should be implemented with A1 applied to **IR** sensor values, A2 to temperature values, and A3 to **IR** sensor values. Therefore, a message is queued when these three algorithms simultaneously detect abrupt changes in their respective variable parameters. Specifically, when **DROWSIER** identifies an abrupt variation using the A1, A2, and A3 approaches, a message containing all sensor values is sent immediately. This verification occurs minute by minute. Additionally, if no abrupt variations are detected by **DROWSIER** in the past 15 min, the module will still send a message to the cloud with the most recent sensor readings to ensure the system remains informed.

Applying the **DROWSIER** filter in each module balances responsiveness and energy efficiency, reducing unnecessary data transmissions and preserving module battery life. Additionally, it overcomes the issues of high data volume and bandwidth usage during the real-time wildfire monitoring in a **WSN** based on **LoRaWAN** network.

5.5.3 Temporal Data Analysis

The verification of time intervals between each entry in each dataset is a crucial step in the temporal data analysis process. This examination ensures that the data collected is continuous and consistent across the observation period. By assessing the time difference between consecutive entries, it is possible to identify any irregularities, such as missing data points or interruptions in data transmission, which could impact subsequent analyses.

5.5.3.1 Dataset 1 - Verification of time intervals

In Dataset 1, each module was configured to transmit data in fixed cycles, expected to be 60-second intervals. Therefore, the dataset must have at least one entry between each minute. Calculating the time delta between successive entries can highlight any deviations from this expected interval.

The calculated time differences between consecutive entries in Dataset 1 present a mean interval of approximately 14.32 s and a standard deviation of 11.32 s. It indicates that most entries occur within a relatively short timeframe. However, the minimum time interval recorded is close to zero, and the maximum time difference reaches 109.67 s. This means that the entries were recorded in some periods without the expected time intervals.

Additionally, the 25th percentile is 5.14 s, the 50th percentile is 11.60 s, and the 75th percentile shows intervals of 21.16 s. These larger intervals may indicate occasional delays or transmission gaps, potentially due to factors such as network interference or sensor communication lags.

To enable consistent temporal analysis, the strategy chosen is to standardize the dataset with uniform intervals across the entire timeframe. The chosen method involves calculating the average of all sensors and battery readings within each 60 s to consolidate data points effectively. In cases where no entries are recorded for a particular 60 s interval, the last observed average value is carried forward, maintaining data continuity while preserving the dataset's original trends.

After resampling Dataset 1 to 60-second intervals, notable changes were observed in the distribution metrics for each variable. The entry count decreased slightly from 18096 to 4319, reflecting data consolidation into uniform time intervals. For Battery, the mean remained close to the original, with a slight reduction from 3.78 V to 3.77 V. The standard deviation also decreased from 0.17 V to 0.05 V, indicating a reduction in extreme fluctuations post-resampling. The minimum and maximum values stayed consistent (min: 3.55 V, max: 3.99 V), preserving the original range.

Humidity slightly reduced the mean from 77.04% to 77.01%, while the standard deviation decreased from 18.55% to 14.86%. This suggests overall continuity in the humidity measurements, with interquartile ranges remaining consistent. Temperature's mean increased marginally from 8.01 °C to 8.19 °C after resampling, with the standard deviation holding steady around 7.06 °C to 6.91 °C. Minimum and maximum values remained stable (min: 0 °C, max: 28 °C), indicating that the resampling preserved the original temperature range.

The IR sensors displayed minor variations in means and standard deviations, with a slight decrease in the average values, indicating more uniform readings post-resampling. For instance, IR1's mean dropped from 786.51 to 781.67, and its standard deviation decreased from 399.61 to 275.54. Similar trends were observed across the other IR sensors, with quartiles showing minor adjustments and maximum values remaining at 1023.

The stationarity check using the ADF outputs greater stability after resampling, especially for the Humidity and IR variables. Initially, before standardization, all sensors presented non-stationary behavior, with ADF statistics and p values that did not attend the critical levels required to reject the null hypothesis of non-stationarity. However, after standardization, Humidity (ADF: -3.30, p value: 0.014) and all IR variables (IR1 to IR5) exhibited values significantly lower than the critical level

of 5%, indicating that these variables became stationary after resampling the dataset. In contrast, Temperature remains non-stationary, with **ADF** and p values of -2.08 and 0.25 , respectively. The Battery variable, which was close to stationary before standardization, now appears strongly non-stationary (**ADF**: -1.27 , p value < 0.64).

Overall, the variable distributions remained consistent, preserving data integrity and ensuring that the new sampling strategy did not impact extreme values. The slight reduction in variability, as indicated by lower standard deviations across several variables, suggests a smoothing effect that removes noise without compromising essential temporal information. On the other hand, this approach reduces potential impacts from missing data and provides a stable foundation for subsequent analysis, such as detecting temporal patterns or implementing time-series models.

5.5.3.2 Dataset 2 - Verification of time intervals

Similar to Dataset 1, the data acquisition system for Dataset 2 was configured with a 60-second interval for data transmission from each module. Consequently, this configuration is expected to obtain at least one data entry per minute. The time intervals between successive entries were computed for the entire dataset to evaluate the consistency of data transmission.

The statistical summary of the time differences is as follows: the dataset contains an average interval of approximately 7.04 s and a standard deviation of 7.24 s. The minimum recorded interval is 0 s, while the maximum interval extends to 886.12 s. Additionally, the 25th percentile is 1.81 s, the 50th percentile (median) is 4.73 s, and the 75th percentile is 9.93 s, indicating variability in data transmission frequency.

The consistency across the dataset was achieved by a resampling method with 60-second intervals. This approach aligns with the previously observed variability in the recorded intervals and aims to standardize the data structure. For each resampled, average values of sensor readings within the 60-second window were calculated. When no data entries were available for a given timestamp, the most recent available value was carried forward to fill the gap. This method preserves the continuity of the data and maintains the overall trends.

Before resampling, the original data had 319594 entries, while the resampled dataset reduced this to 37504 entries. The effect of resampling can be observed through the following statistical changes in the variables. The mean Battery value of Battery slightly decreased from 3.67 V to 3.60 V after resampling. The standard deviation also reduced from 0.255 V to 0.06 V, indicating less variability in the resampled data. The range remained similar, with the minimum and maximum values unchanged. The mean Humidity value increased from 65.99% to 67.80% , and the standard deviation slightly decreased from 19.35% to 17.66% . The interquartile range narrowed, reflecting a more consistent distribution in the resampled data.

The mean Temperature decreased marginally from 14.25°C to 13.85°C , while the standard deviation dropped from 5.0°C to 4.87°C . The resampling process slightly compressed the spread of temperature values, leading to a narrower distribution. All **IR** sensor readings (IR1 to IR5) reduced their standard deviations, indicating less spread in the data after resampling. The mean values of IR1, IR2, IR3, and IR4 generally showed minor fluctuations. IR5 obtained a more pronounced

decrease in mean value, from 791.19 to 800.91. The minimum and maximum values for the **IR** sensors remained consistent.

Analysing and comparing the stationarity results before and after resampling for each variable in Dataset 2 can determine if this method changes any significant data characteristics. In this sense, Temperature has the **ADF** statistic of -2.29 and a p value of 0.17 , indicating non-stationarity in its original data. After resampling, the **ADF** statistic improved to -4.27 with a lower p value of 0.0005 . Therefore, the Temperature variable turns stationary.

Before Resampling, Humidity's **ADF** statistic was -2.30 , and a p value of 0.17 with non-stationarity. Then, after this process, the **ADF** statistic increased to -5.03 , and the p value is lower than 0.0001 , indicating that the variable is approaching stationarity. All **IR** sensors were identified as stationary before resampling, with **ADF** statistics ranging from -5.90 to -4.93 and very low p values. After, the stationarity of **IR** sensors improved even further, with **ADF** statistics becoming more negative (e.g., -6.57 for **IR1** and -6.54 for **IR3**) and p values decreasing further, reinforcing their stationary nature. On the other hand, the Battery's **ADF** statistic was -5.04 , and the p value was lower than 5% , and then, the **ADF** after resampling became -2.04 with a p value of 0.26 .

For the most part, the 60s resampling process showed some improvement toward stationarity. Temperature and Humidity became stationary, and the **IR** sensors are more stable after resampling. In contrast, Battery variables were stationary and became non-stationary. Therefore, the resampling process positively affects significant variables to detect wildfires, enhancing their temporal structure.

5.5.3.3 Dataset 3 - Verification of time intervals

The data in Dataset 3 was collected with the **DROWSIER** filtering implemented. As a result, it is expected that the minimum interval between consecutive messages from each module should be 15 min. This filtering ensures efficient data transmission by reducing redundant information while allowing for earlier transmissions when anomalies are detected.

The time differences between consecutive messages for the three modules in Dataset 3 were analysed, and the descriptive statistics of these intervals are presented. The time interval differences have a mean interval of approximately 4.90 min, with a standard deviation of 4.70 min. The minimum recorded interval is 0.00 min, while the maximum interval remains just below 15 min. These results confirm that the **DROWSIER** filter generated intervals smaller than 15 min, as expected, especially in cases where anomalous events were detected.

Given these irregular intervals, standardization of the dataset was necessary to ensure consistency in temporal analysis. A resampling process with 60s intervals was applied, similar to the approach used in Datasets 1 and 2. However, in contrast to the previous datasets, where mean values were used during resampling, for Dataset 3, a normalization strategy was employed. The last recorded value was carried forward for each resampled interval until a new value was obtained. This approach preserves the temporal dynamics of the data while adapting it to a uniform time scale, facilitating analysis and comparison.

The resampling process applied to Dataset 3 increased the total number of entries, from 4393 to 21527. Before resampling, the dataset displayed variability across its variables, with mean values

such as 15.2°C for Temperature, 73.18% for Humidity, and ranges for the **IR** sensors (IR1-IR5) from 20 to 1023 **ADC** counts. After resampling, the statistics for most variables remained consistent with the pre-resampled dataset. For instance, the mean Temperature slightly increased to 15.26°C, while Humidity's mean value showed minimal change at 73.19%. Similarly, the ranges for the **IR** sensors (IR1-IR5) remained intact, from 20 to 1023 **ADC** counts.

The data in Dataset 4 and Dataset 5 were also collected with the **DROWSIER** filtering implemented. Consequently, the same procedure was applied, as these datasets share the same filtering that ensures a minimum interval of 15 min between consecutive messages unless anomalies are detected. In this sense, in both mentioned datasets, the time differences between successive messages were similarly analysed, confirming the expected behavior of the **DROWSIER** filter in generating intervals smaller than 15 min during anomalous events. To ensure consistency in temporal analysis, both datasets were resampled to 60s intervals, following the methodology described before. Post-resampling, the statistical properties of the variables in Datasets 4 and 5 remained consistent with their pre-resampled versions, ensuring the integrity of the data for subsequent analysis and comparison.

5.6 Machine Learning Models

This section is based on the insights derived from the **EDA** in Section 5.4, which is intended to apply **ML** techniques to identify patterns and predict values based on the historical data collected by the **WSN** in different scenarios. Therefore, the main objective is to develop strategies focused on detecting anomalies that can indicate the presence of abrupt events such as fires. Four models were implemented, namely: **RWR**, **ARIMA**, **RNN**, and **TimesFM**.

They were chosen based on the following considerations. While practical for linear relationships, the **RWR** model usually has difficulty during the non-linear dynamics often present in environmental data [230] (which may indicate sudden data changing clearly). Similarly, **ARIMA** performed well for stationary series but sometimes faced challenges with sudden shifts in data [265] (especially during fire events). **RNN** models, with their capability to capture temporal dependencies, can indicate forecast data based on the data patterns, and they may show promising results in handling abrupt changes [221]. Finally, **TimesFM**, with its ability to generalize across datasets without requiring additional training, offers a practical solution for large-scale environmental monitoring systems [270].

All models were trained and evaluated using a sliding window approach [278]. The training phase uses a specified data interval, determined through iterative experimentation and performance evaluation, to predict the next 15 min. The testing was performed exclusively on the forecast period, ensuring that training data for a given window was not reused during test phase. This approach tries to replicate a real-time forecasting scenario, aligning with the requirements for early detection of WSNs' variables anomalies.

The datasets analysed in this section were resampled to a fixed interval of 60s, as described in Section 5.5.3. This ensures uniformity and comparability across all variables and scenarios. **ML**

models were evaluated over two distinct periods: with and without **WSE**. Then, in the graphical results, periods corresponding to supervised fire tests are shaded in gray and labeled as **FS**. The start of each **FS** period was marked at the ignition of biomass, while the end was defined as the point where flames were no longer visible for at least five minutes. All fire tests were conducted under the firefighters' supervision, authorization, and control.

Battery data was excluded from the **ML** analysis for two primary reasons. First, battery levels exhibit minimal variability within the timeframes analysed, with no rapid decay patterns that would necessitate predictive modeling. Second, battery readings are not directly influenced by environmental changes caused by fire events. Thus, including battery levels in anomaly detection would neither improve the detection accuracy nor provide meaningful information.

Anomalies, defined as deviations between predicted and actual sensor readings, are flagged when observed values fall outside the expected range. These anomalies serve as potential indicators of fire events, as they typically correspond to abrupt changes in environmental conditions. In this perspective, anomalies were detected using the **IForest** method [279], with a threshold set at the 95th percentile to balance sensitivity and specificity [280, 281, 282]. This threshold was chosen to minimize false positives while ensuring that significant deviations – such as those caused by fire events – are reliably detected. A lower threshold could lead to excessive false alarms triggered by natural environmental variations, while a higher threshold might overlook critical anomalies associated with fire events. This approach assumes that fire events induce abrupt changes in sensor readings, which are unlikely to align with historical patterns. For instance, sudden spikes in Temperature or **IR** intensity often accompany fire events, making these variables particularly sensitive to anomaly detection.

Each model was evaluated both during the hours preceding and throughout the entirety of the **WSE** periods for each dataset, indicating their ability to predict values and flag anomalies under both normal and fire-induced conditions. The results provide the ability of these models to predict values and flag anomalies under normal and extreme environmental conditions. The effectiveness of each model in detecting fire-induced anomalies was assessed by comparing the predictions to actual sensor readings.

5.6.1 Rolling Window Regression

The **RWR** method was implemented to extend the prior **LR** analysis, in Appendix D - Section D.3, adding a dynamic element to evaluate model performance over time. This approach allows for the continuous updating of the model by using a rolling window for both training and test. Specifically, a step size of 15 min was chosen, meaning that the training and testing windows move forward incrementally by this amount throughout the datasets.

The setup for the **RWR** follows a methodology similar to the previously mentioned regression analysis. Therefore, the selected lag values for each variable were obtained from **ACF** and **PACF** methods (performed in Section 5.4). Furthermore, to evaluate the **RWR** under different environmental conditions, its performance was assessed using datasets with contrasting **TCD** levels.

In this sense, the datasets used were Dataset 1 and Dataset 2, which were uniformly resampled and standardized in time, as described in the preceding subsection.

5.6.1.1 RWR models for WSN in Low TCD scenario

The analysis began by splitting Dataset 1 into distinct train and test periods. The initial training period was from February 22, 15h25 to February 23 at 7h30, in total 16h05 of data. Next, the testing period started at 7h45 on February 23. The rolling window approach was implemented up to 11h00, incrementing the window every 15 min.

For every run of the rolling window, a **LR** model was trained individually for each variable using its respective lagged values. The selected lags were as follows: 5 for Temperature, 3 for Humidity, 1 for IR1-IR4, and 8 for IR5. Each **LR** model was then validated using the actual data corresponding to the subsequent 15 min, which matched the prediction interval. This iterative process enabled continuous model training and testing, allowing for a dynamic assessment of the model's performance as new data became available.

Predictions for every window were stored, while the corresponding actual values were retained for comparison purposes. Overall, 14 rolling windows were used to evaluate the model's ability to forecast subsequent samples accurately. The performance metrics of these models for each variable are summarized in Table 5.8, which indicates the minimum and maximum results.

Table 5.8: Range Minimum and Maximum of the RWR Models Metrics for WSE period in Dataset 1.

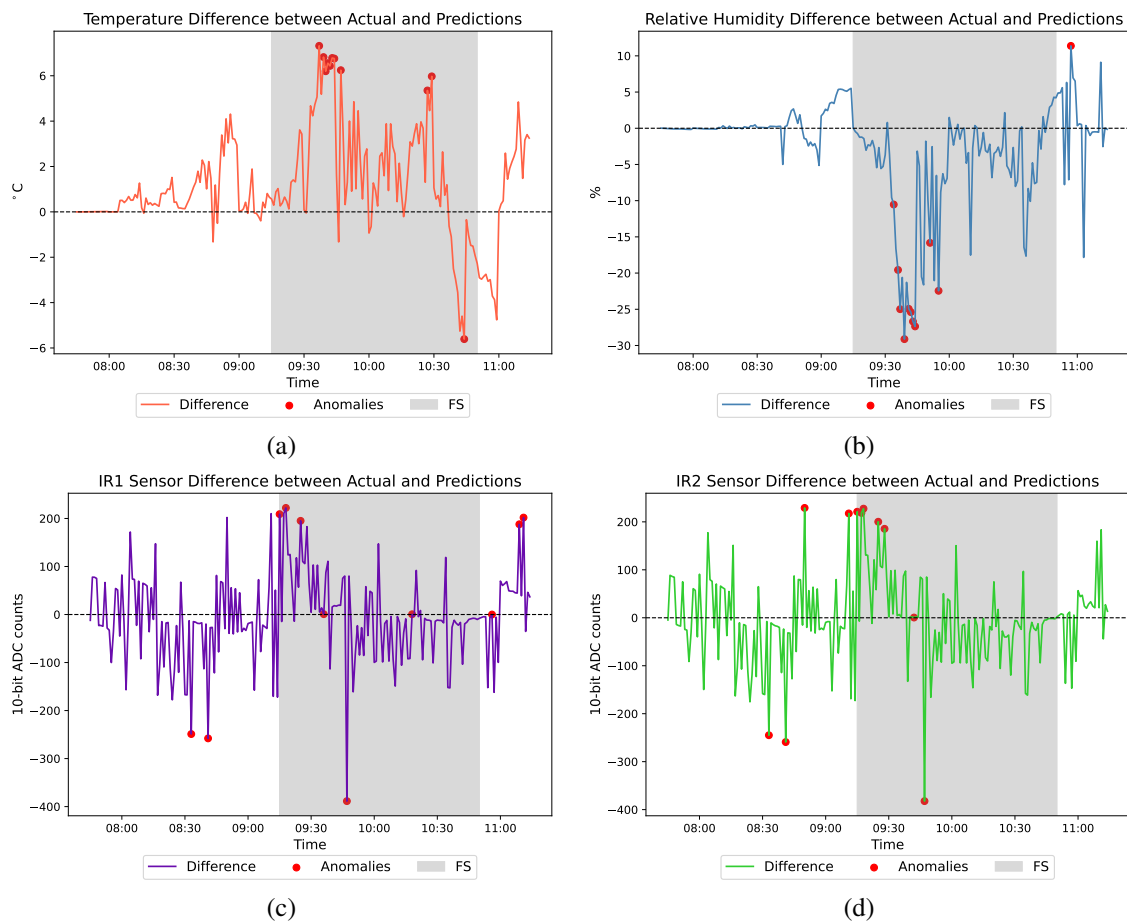
Variable	MAE	MedAE	Explained Variance	MSE	RMSE
Temperature	[0.007, 4.863]	[0.006, 5.187]	[-1.223, 0.391]	[0.00008, 28.900]	[0.009, 5.376]
Humidity	[0.088, 17.649]	[0.091, 20.606]	[-0.183, 0.329]	[0.010, 397.793]	[0.101, 19.945]
IR1	[32.964, 120.665]	[7.861, 118.081]	[-0.080, 0.023]	[2976.849, 19210.064]	[54.561, 138.600]
IR2	[28.835, 124.730]	[4.525, 120.850]	[-0.086, 0.020]	[3192.043, 20582.422]	[56.498, 143.466]
IR3	[31.342, 111.745]	[12.462, 116.177]	[-0.079, 0.066]	[2653.173, 16717.100]	[51.509, 129.295]
IR4	[32.411, 124.172]	[16.105, 135.412]	[-0.074, 0.069]	[2791.804, 18604.026]	[52.838, 136.397]
IR5	[37.928, 84.572]	[22.396, 85.340]	[-0.329, 0.131]	[2476.903, 13670.395]	[49.768, 116.920]

For Temperature and Humidity, the **RWR** method outputted moderate predictive accuracy, with minimum error values close to zero (e.g., **MAE** and **Median Absolute Error (MedAE)**) in some run, indicating its ability to capture patterns in stable environmental conditions. However, the high maximum errors and negative Explained Variance values observed in some windows highlight the method's limitations in handling transient or abrupt changes in these variables. The **IR** sensors (IR1-IR5) presented more significant challenges for the **RWR** models, as evidenced by consistently high error metrics (e.g., **MSE** and **RMSE**) and predominantly negative Explained Variance values. This was particularly notable for IR1 and IR2, where the models have difficulties in generalizing effectively across rolling windows, resulting in significant prediction errors. Among the **IR** sensors, IR5 have relatively better performance, with lower maximum errors and a maximum Explained Variance of 0.131, suggesting its data patterns were captured more robustly.

This behavior could be advantageous, as the model's difficulty adapting to abrupt variations during fire events may indicate irregular patterns caused by such incidents, enhancing its potential

utility for fire anomaly detection. Graphical plots in Appendix D - Figure D.16 provide a visual comparison of the predicted and actual sensor readings, facilitating a more intuitive assessment of the model's performance. To contextualize these results, the WSE periods were highlighted on the plots, as indicated by the FS column, marking the moments when fire events occurred and offering additional understanding of the model's behavior during these critical periods.

The differences between the actual and predicted values were computed to identify deviations that could indicate anomalies. Therefore, a plot was created to display the differences (Figure 5.48), with anomalies identified using an IForest method. The anomaly scores were performed, and a 95th percentile threshold was applied to detect significant deviations. The anomalous points were highlighted on the difference plots, emphasizing regions where the model's predictions differed substantially from the actual values, potentially indicating anomalies.



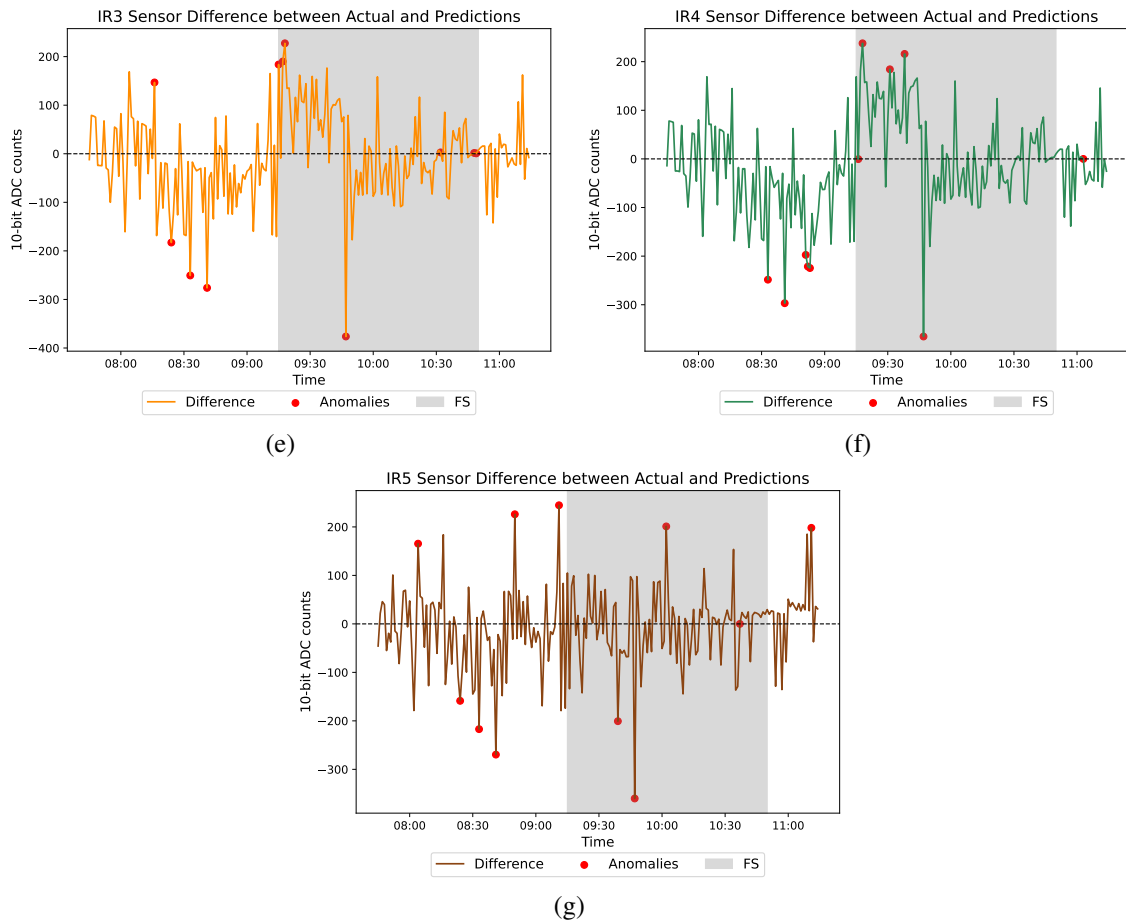


Figure 5.48: Graphical representation of the difference between actual and RWR Model's predicted variables' values in Dataset 1. The plots display the actual versus predicted values for Temperature (a), Humidity (b), and IR1 through IR5 (c)-(g).

The anomaly detection plot analysis shows that Temperature and Humidity variables exhibit anomalies primarily during the fire event period. Before 9h15, no anomalies were detected, indicating that the model's predictions aligned well with actual values under normal conditions. However, starting around 9h30, the Temperature variable shows recurring anomalies (11 anomalies), marking the onset of significant deviations coinciding with the fire event. Similarly, Humidity anomalies are detected shortly after, exhibited 10 anomalies, but there is a noticeable increase in anomaly frequency post-9h30. On the other hand, Humidity continues to show anomalies even after the **WSE** period. This persistent behavior can be attributed to the rolling window model being continuously trained with data influenced by the fire, implying that the model's updates were affected. Therefore, these anomalies should not be prematurely classified as false alarms.

In contrast, the **IR** sensors (IR1–IR5) showed a broader distribution of anomalies across the three periods. During the fire period, IR1 detected 6 anomalies, IR2 and IR3 detected 7 each, IR4 detected 5, and IR5 detected 4. Before the fire period, the anomaly counts were as follows: 3 for IR1, 5 for IR2 and IR3, 5 for IR4, and 6 for IR5. This suggests that external lighting conditions may be affecting the model's performance, as the **RWR** model had difficulty adapting to sudden

and significant fluctuations in illumination. This limitation highlights the need for additional considerations, such as preprocessing for light variability or using more robust models, to improve anomaly detection for **IR** sensors in low **TCD** environments.

5.6.1.2 RWR models for WSN in High TCD scenario

The analysis in the high **TCD** scenario began by selecting Dataset 2. This dataset contains additional days before the first **WSE**, allowing for a broader analysis of pre-event conditions. According to the **EDA** performed in the previous section, trends were observed in all variables within 24h. Therefore, as part of an initial analysis, intervals of 24h, 48h, and 72h were tested to determine the optimal window for capturing patterns and training the model. The metrics indicated that the windows of 24h before **WSE1** provided the best performance. For this reason, the following analysis is based on the method applied using data from the 24h preceding the moment of **WSE1**.

The training period started from May 11, 13h00 to May 12, 13h30. Next, the test period started May 12, 13h45. The rolling window approach was applied incrementally, continuing until 15h45, with the window advancing every 15 min. The same **LR** method used previously was employed, applying the optimal lag values determined for each variable through the **ACF** and **PACF** analysis. The lag values selected were 3 lags for Temperature and IR5, 17 for Humidity, 10 for IR1, 1 for IR2, and 3 for IR4.

This approach allowed the model to continuously train and test predictions, providing a dynamic assessment of its performance under the high **TCD** scenario. In total, 9 runs were conducted to evaluate the model's predictive performance over time. Table 5.9 presents the range of these metrics for the entire analysis period, showcasing the model's performance variations.

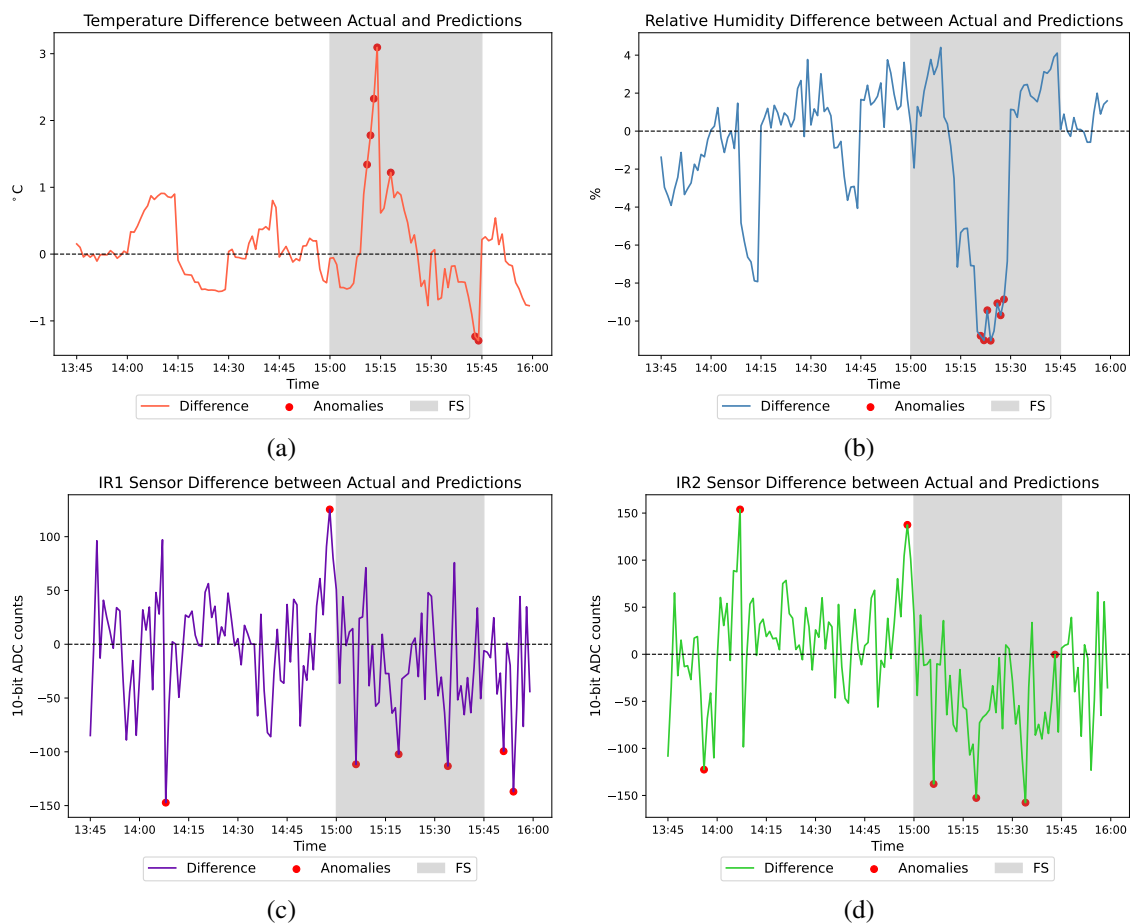
Table 5.9: Range Minimum and Maximum of the RWR Models Metrics for the first WSE period in Dataset 2.

Variable	MAE	MedAE	Explained Variance	MSE	RMSE
Temperature	[0.045, 0.816]	[0.044, 0.818]	[-0.626, 0.096]	[0.040, 1.467]	[0.062, 1.211]
Humidity	[0.682, 8.506]	[0.586, 9.063]	[-0.659, 0.015]	[0.822, 76.908]	[0.907, 8.770]
IR1	[21.952, 47.696]	[23.873, 47.894]	[-0.109, 0.150]	[813.092, 3255.513]	[28.515, 57.057]
IR2	[28.878, 68.137]	[18.848, 74.207]	[-0.051, 0.044]	[1280.011, 5993.290]	[35.777, 77.416]
IR3	[26.985, 57.320]	[20.784, 51.010]	[-0.071, 0.064]	[1305.722, 5054.989]	[36.135, 71.098]
IR4	[28.782, 45.547]	[15.792, 42.430]	[-0.134, 0.109]	[1315.873, 3070.731]	[36.275, 55.414]
IR5	[27.922, 87.713]	[21.411, 69.467]	[-0.180, 0.115]	[1194.797, 10699.528]	[34.566, 103.439]

These metrics indicate that the models have limited predictive power for **IR** variables, with some difficulty in accurately capturing the underlying patterns of the data. IR1 performed relatively better compared with IR2-IR5, with lower maximum errors (**RMSE** up to 57.057), while IR5 showed the highest variability (**RMSE** up to 103.439), reflecting its sensitivity to environmental factors. For Temperature, the model performed well overall, with low error values (**MAE** up to 0.816 and **RMSE** up to 1.211). Humidity displayed more significant variability, with higher errors (**MAE** up to 8.506). These results indicate the model's ability to capture general trends but highlight challenges in handling dynamic conditions, particularly for the **IR** sensors.

Mainly, the inconsistency of metrics is due to abrupt variations around the **WSE** period, which the **RWR** model could not handle well. Such abrupt changes can disrupt the model's data continuity and stationarity assumptions, affecting predictive performance. In addition to evaluating the performance metrics, graphical plots were generated to visually check the model's behavior during the forecast period. It is possible to compare the **RWR** model predictions with the actual sensor readings, providing the model's performance and ability to track the variations in each variable over time in Appendix D - Figure D.17.

Figure 5.49 displays the difference between the forecasted values and the actual readings for each variable, providing a clearer view of the model's deviations over time. In each plot, anomaly detection was performed using an **IForest** method to identify significant deviations. The anomaly scores were calculated, and a 95th percentile threshold was applied to flag values as anomalies. These anomalous points were then highlighted in the plot, emphasizing moments where the model's predictions significantly diverged from the actual data, which may indicate critical events or sensor irregularities.



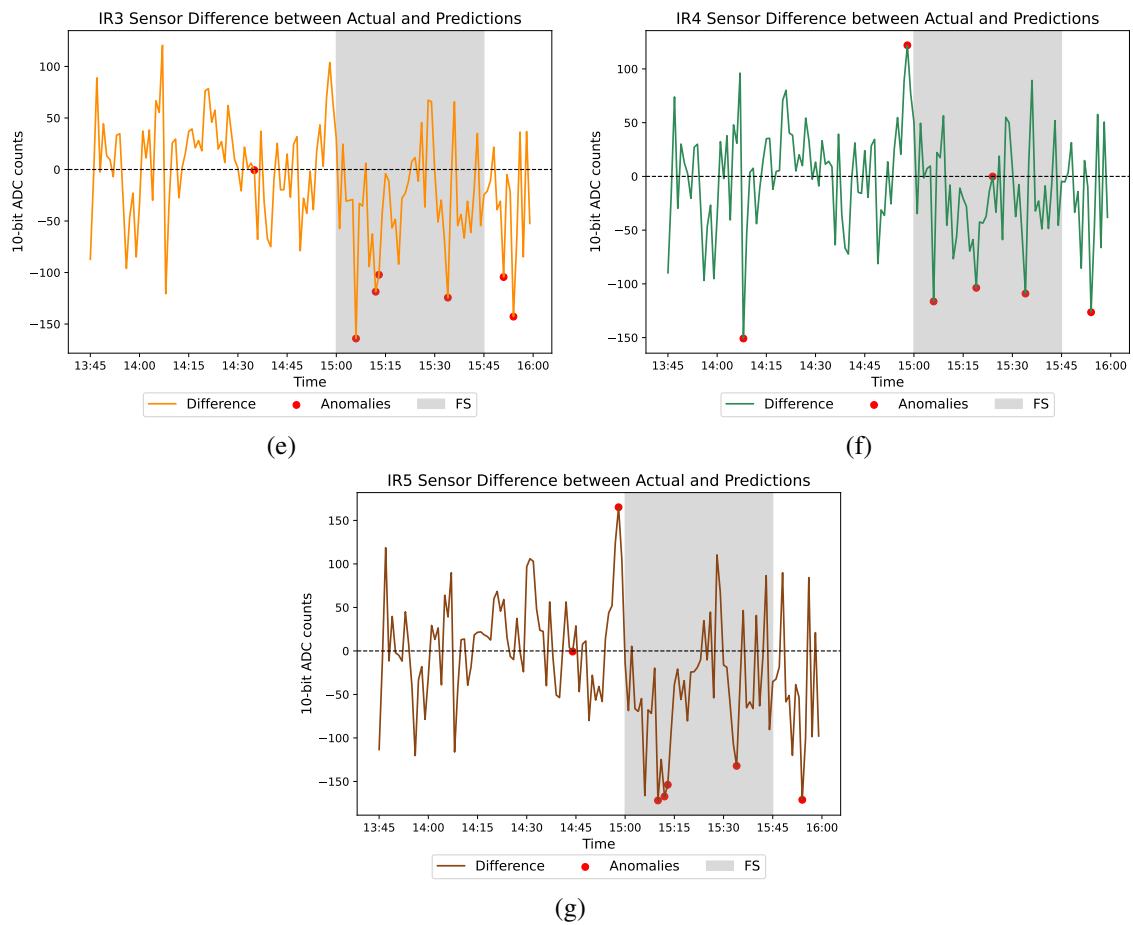


Figure 5.49: Graphical representation of the difference between actual and RWR Model's predicted variables' values in Dataset 2 during first WSE. The plots display the actual versus predicted values for Temperature (a), Humidity (b), and IR1 through IR5 (c)-(g).

The difference between the forecasted and actual Temperature values remains relatively small before 15h00. However, as the **WSE** period progresses, the difference grows more pronounced, indicating that the model forecast is not prepared for sudden changes. Anomalies are identified only during the fire event, with a total of 7 anomalies detected, suggesting the model's sensitivity to abrupt changes caused by environmental events.

The Humidity plot reveals more intense differences after 14h45. Similar to the Temperature, the anomaly detection method identifies spikes during the **WSE**, with 7 anomalies occurring during the fire period. The most frequent and intense anomalies are observed near the event's peak, highlighting that humidity variations associated with the fire significantly deviate from the model's predictions **WSE**.

Each **IR** sensor plot (IR1 through IR5) shows a similar behavior, with differences between actual and forecasted values that fluctuate significantly throughout the **WSE** period. IR1 detected 3 anomalies during the fire and 2 before and after the event; IR2 identified 4 anomalies during the fire and 3 before it; IR3, IR4, and IR5 each detected 4 anomalies during the fire period, with additional anomalies before and after: 1 for IR3, 2 for IR4, and 2 for IR5 after the fire.

Overall, the anomaly detection approach using the **IForest** method identifies significant deviations between the **RWR** forecasted data and actual data during the **WSE**. This analysis suggests that while the model is not perfectly reliable in differentiating all irregularities, it highlights critical periods where environmental changes occur.

5.6.2 ARIMA

The **ARIMA** model was employed for time series analysis across both Dataset 1 and Dataset 2 to forecast sensor readings and identify potential anomalies. **ARIMA** is a powerful and widely used method for time series analysis, especially suitable for data with trends or seasonal patterns. Therefore, the model's configuration was automated using the AutoARIMA algorithm, which optimizes the model's parameters by performing an extensive search over a specified range of **AR** and **MA** orders [283].

The AutoARIMA algorithm adjusts the parameters p , d , and q (where p is the number of lag observations, d is the degree of differencing, and q is the size of the moving average window) to minimize errors and improve the model's fit. Seasonal effects were also considered, and the seasonality parameter m was configured based on the identified periodicity in the data. Specifically, the AutoARIMA uses tests to calculate the degree of differencing needed to make the series stationary, during the all of the modelling was used the **ADF** test.

The analysis was conducted using rolling windows, similar to the approach used in previous sections. The datasets were split into initial train and test periods, and the model was iteratively updated every 15 min. The **ARIMA** model was trained on the most recent training data for each run and tested on the subsequent 15-minute window. This rolling forecast approach allowed the model to adapt dynamically, continuously assessing its performance in response to the evolving data patterns.

After analysing the **ACF** and **PACF** plots for the resampled datasets, it was calculated that none of the variables exhibited significant autocorrelation beyond lag 4. Thereby, the following AutoARIMA configuration was applied consistently across both datasets: $start_p = 0$, $start_q = 0$, $max_p = 2$, $max_q = 2$, $D = None$, $start_P = 0$, $start_Q = 0$, $max_P = 3$, and $max_Q = 3$. This setup ensured that the model accounted for potential short-term dependencies while minimizing complexity.

Additionally, the `stepwise` option was disabled to maintain consistency in the processing. By turning off `stepwise`, the model avoided making potential extrapolations or shortcuts that could compromise the uniformity of the analysis, ensuring a more systematic and stable parameter exploration. A combination of metrics, including **MAPE**, **MAE**, **MSE**, and **RMSE**, was employed to evaluate the **ARIMA** model's performance. These metrics are commonly used for **ARIMA** models. This is because **ARIMA** models are sensitive to both the magnitude and the distribution of forecast errors. Using a diverse set of metrics, it is possible to understand the aspects of the model's predictive performance, such as overall error magnitude, sensitivity to large deviations, and the model's ability to minimize absolute and squared errors.

5.6.2.1 ARIMA models for WSN in Low TCD scenario

The **ARIMA** model was applied to the **WSN** data collected in the low **TCD** scenario (Dataset 1). The analysis began by splitting the data into distinct train and test periods. The training data encompassed observations from February 22, 15h25 to February 23 at 7h30, while the testing period started February 23 at 7h30. This setup allowed the model to be trained and tested on early data and continuously tested in 15-minute increments, until February 23 at 11h00.

The AutoARIMA algorithm was employed for each run to automatically determine the optimal p , d , and q parameters for each variable. A new model was trained for each variable and each rolling window. After the training model process, a forecast was generated for the subsequent 15-minute period, and then the predictions were compared to the actual sensor readings and performance metrics were calculated to evaluate the **ARIMA** model. These metrics are presented in Table 5.10, which ranges from the minimum to the maximum of the model's forecasting performance across different variables.

Table 5.10: Range Minimum and Maximum of the ARIMA Models Metrics for WSE period in Dataset 1.

Variable	MAPE [%]	MAE	MSE	RMSE
Temperature	[4.87, 28.01]	[0.25, 4.15]	[0.08, 21.10]	[0.28, 4.59]
Humidity	[0.15, 29.30]	[0.13, 17.71]	[0.02, 402.27]	[0.15, 20.06]
IR1	[9.30, 79.61]	[36.91, 83.24]	[2863.42, 14293.28]	[53.51, 119.55]
IR2	[7.90, 77.38]	[32.33, 84.50]	[2121.61, 14842.95]	[46.06, 121.83]
IR3	[7.72, 87.70]	[34.68, 82.09]	[2532.76, 13342.23]	[50.33, 115.51]
IR4	[11.71, 79.60]	[48.59, 108.59]	[3370.95, 15612.54]	[58.06, 124.95]
IR5	[9.21, 71.29]	[38.91, 86.37]	[2453.22, 15120.32]	[49.53, 122.96]

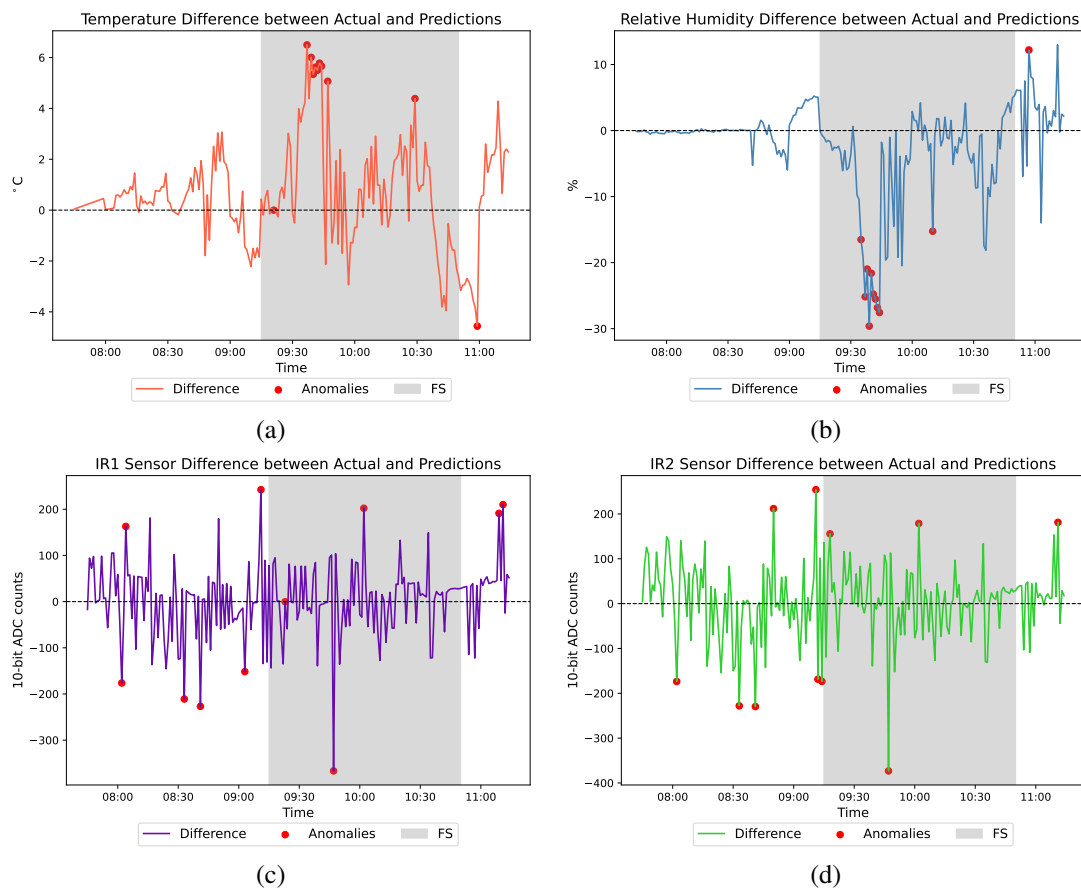
The range of metric values highlights the **ARIMA** model's strengths and limitations across the different sensor readings. The model exhibited reasonable accuracy for Temperature, with **MAPE** values ranging from 4.87% to 28.01%. The relatively low error metrics (**MAE** up to 4.15 and **RMSE** up to 4.59) indicate that the **ARIMA** model was effective at capturing the general trends in temperature data. However, the high error metrics may suggest that the model has difficulty capturing the deviations during abrupt changes.

Humidity displayed higher variability in model performance, as evidenced by a broader range of error metrics. The **MAPE** values extended up to 29.30%, and the maximum **MAE** reached 17.71, reflecting the model's challenges in accurately predicting sudden changes in humidity during the **WSE**. Nevertheless, the lower end of the error ranges suggests that the model could still provide reliable predictions during more stable periods.

The **IR** (IR1-IR5) showed significant differences in performance. For all **IR** sensors, the **MAPE** values ranged from single digits (e.g., 7.90% for IR2) to high values exceeding 70%, indicating considerable variability in the model's ability to capture patterns. The error metrics (**MAE**, **MSE**, and **RMSE**) highlight that the **ARIMA** model has difficulty in capturing the **IR** pattern data during some runs, particularly during dynamic conditions introduced by the **WSE**. For example, IR4 exhibited the highest maximum **MAE** (108.59) and **RMSE** (124.95), reflecting challenges in adapting to abrupt changes or noise in the data.

To provide additional information into the model's behavior, graphical plots are included in Appendix D - Figure D.18. These plots visually compare the predicted and actual values across all rolling windows, highlighting moments where the model's predictions aligned closely with the collected data and where significant divergences occurred, providing a clearer understanding of the model's accuracy and limitations.

The discrepancies in performance metrics among different variables underscore the need for careful interpretation of **ARIMA** model outputs. Therefore, the anomaly detection process, illustrated in Figure 5.50, shows the plots for each variable, highlighting the detected anomalies during the **ARIMA** model's forecasting. The anomalies were identified by calculating the difference between the forecasted and actual sensor readings and applying the **IForest** method. The **IForest** algorithm analysed these differences, and a threshold based on the 95th percentile of anomaly scores was used to indicate significant deviations. Anomalous points are marked on the plots to emphasize the areas where the **ARIMA** model's predictions significantly diverged from the actual data.



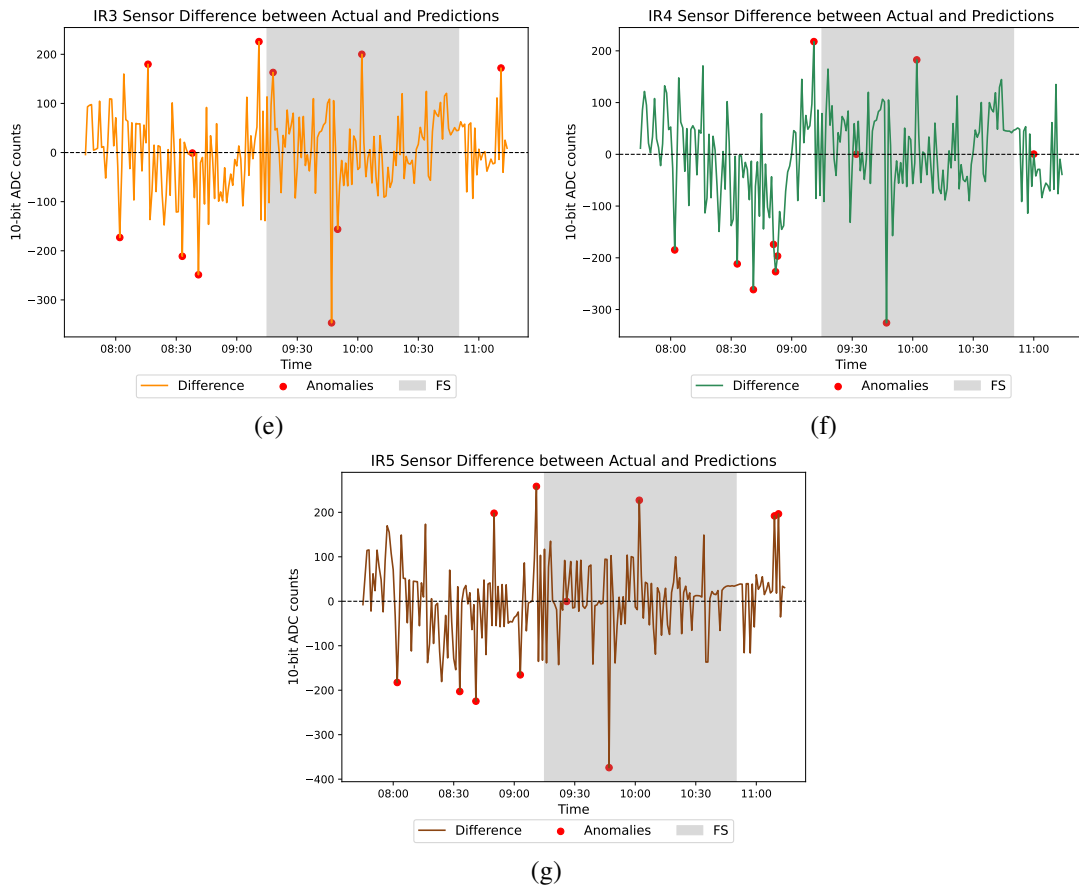


Figure 5.50: Graphical representation of the difference between actual and ARIMA predicted variables' values in Dataset 1. The plots display the actual versus predicted values for Temperature (a), Humidity (b), and IR1 through IR5 (c)-(g).

The Temperature plot reveals variations between the forecasted and actual values ranging from -2°C to 2°C before **WSE**. However, no anomalies were detected before the **WSE**, indicating that the model maintained its performance during stable conditions. Once the **WSE** begins, around 9h15, anomalies are identified, with a total of 10 anomalies flagged during the fire event. This indicates that the **ARIMA** model, combined with **IForest**, was effective in detecting fire-related changes within minutes of the event's onset.

For Humidity, a similar behavior is observed. The differences between forecasted and actual values remain minor before the **WSE**, with no anomalies flagged. However, when the **WSE** starts, there is an intense drop in the differences, and **IForest** flags 10 anomalies during the event. One anomaly appears after the **WSE**, suggesting that the model has difficulty handling the effects of **WSE** data during the rolling window updates. This highlights the need for future strategies to exclude **WSE** affected data to improve post-event modeling performance.

The **IR** sensor plots exhibit differences between the **ARIMA** forecast and actual values ranging from approximately -400 to 200 . While there is a large number of anomalies detected in **WSE** period from Temperature and Humidity anomalies, some false alarms occur in all **IR** plots before and after the **WSE**. Before the **WSE**, IR1 detected 6 anomalies, IR2 and IR4 detected 7 each, IR3

and IR5 detected 6 each. During the **WSE**, IR1, IR2, IR3, IR4, and IR5 detected 3, 3, 4, 3, and 3 anomalies, respectively. After the **WSE**, 2 anomalies were flagged for IR1 and IR5, while 1 anomaly was detected for IR2, IR3, and IR4.

This behavior suggests that, despite false positives, the **ARIMA** model coupled with **IForest** anomaly detection effectively highlights critical periods of change, particularly during fire events. Consequently, this combined approach can serve as a corroborative mechanism for fire detection when anomalies are simultaneously identified across Temperature, Humidity, and all **IR** sensors. This strategy could be a strong indicator of fire, helping to reduce false positives and enhancing the system's reliability.

5.6.2.2 ARIMA models for WSN in High TCD scenario

The **ARIMA** model was also applied to the **WSN** data collected in the high **TCD** scenario (Dataset 2) to analyse the model's performance in a denser forested environment. This dataset includes additional days before the first **WSE**, enabling a broader analysis of pre-event conditions. As part of an initial exploration, rolling windows of 24h, 48h, and 72h were tested to determine the optimal window length for capturing patterns and training the model. The metrics showed that the 48h interval before **WSE1** achieved the best performance, as it yielded the most favorable evaluation metrics compared to the other intervals. This choice effectively balanced the trade-off between capturing sufficient historical data and maintaining model responsiveness to changes. Consequently, the following analysis focuses on applying the **ARIMA** model using the 48h interval as the training window.

Similar to the low **TCD** case, the data were divided into a distinct train and test period. The training data started from May 10 at 13h00 until 13h30 on May 12, and the testing period began at 13h45, and both (train and test) extended until May 12, 15h45. The **ARIMA** model parameters were configured based on the **ACF** and **PACF** analyses, as discussed earlier. AutoARIMA method optimized the model's parameters for each variable regarding potential seasonal effects. Each model was trained using the most recent data for each rolling and tested in the subsequent 15-minute window. These observations are further quantified in Table 5.11, which summarizes the range of performance metrics for the **ARIMA** model applied to each variable during the first **WSE** period.

Table 5.11: Range Minimum and Maximum of the ARIMA Models Metrics for the first WSE period in Dataset 2.

Variable	MAPE [%]	MAE	MSE	RMSE
Temperature	[0.39, 5.19]	[0.05, 0.85]	[0.00, 1.75]	[0.07, 1.32]
Humidity	[1.06, 14.31]	[0.59, 7.43]	[0.50, 59.02]	[0.71, 7.68]
IR1	[6.79, 13.24]	[28.99, 56.44]	[1173.13, 3918.53]	[34.25, 62.60]
IR2	[7.25, 12.40]	[27.54, 49.90]	[1181.14, 3982.99]	[34.37, 63.11]
IR3	[6.18, 11.77]	[24.79, 48.83]	[1014.96, 3219.62]	[31.86, 56.74]
IR4	[6.98, 12.75]	[27.62, 52.25]	[1160.14, 3817.24]	[34.06, 61.78]
IR5	[5.62, 14.22]	[27.94, 71.24]	[1151.68, 7671.59]	[33.94, 87.59]

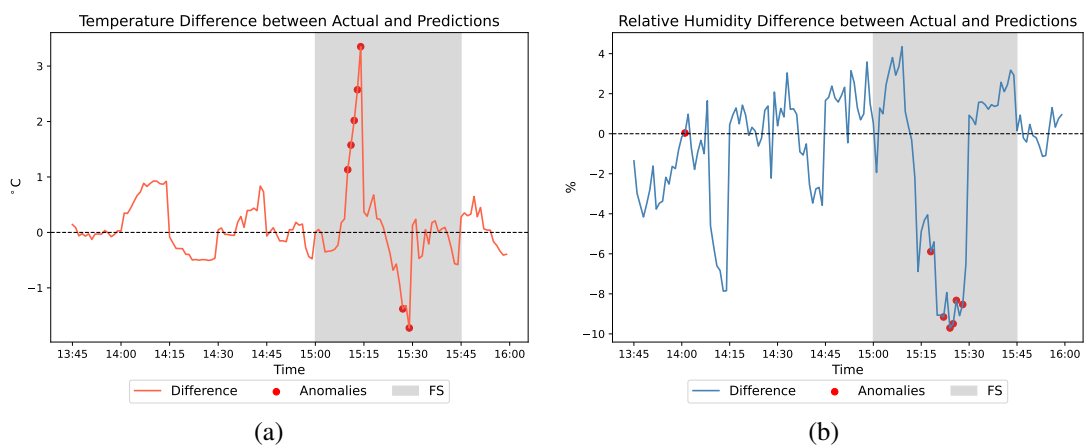
The range of the metrics indicates the model's ability to handle environmental complexity while also highlighting its limitations for certain variables. For Temperature, the **ARIMA** model outputs

low **MAPE** values ranging from 0.39% to 5.19%. The low error metrics (**MAE** up to 0.85 and **RMSE** up to 1.32) indicate that the model effectively captured temperature trends in the high **TCD** scenario, even during periods of environmental variability. Humidity exhibited higher variability in performance, with **MAPE** values extending up to 14.31%. The error metrics (**MAE** up to 7.43 and **RMSE** up to 7.68) suggest challenges in accurately predicting humidity during dynamic conditions, particularly those introduced by the **WSE**.

The higher **MAPE**, **MAE**, and **RMSE** values for the **IR** sensors highlight even more significant challenges in forecasting **IR** readings. IR1, IR2, and IR4 showed similar behavior, with **MAPE** values ranging from approximately 6.79% to 13.24%. Their **MAE** values reached a maximum of 56.44, 49.90, and 52.25, respectively, while **RMSE** values extended up to 62.60, 63.11, and 61.78. IR3 displayed slightly better performance, with lower maximum **MAE** (48.83) and **RMSE** (56.74), suggesting that the model was marginally more effective for this sensor compared to others in the same category. IR5, however, exhibited the highest variability, with a maximum **MAPE** of 14.22%, **MAE** of 71.24, and **RMSE** of 87.59, reflecting the model's difficulty in capturing patterns for this variable during the **WSE** period.

To complement the analysis of the **ARIMA** model's performance in the high **TCD** scenario, Appendix D - Figure D.19 provides a graphical comparison between the predicted and actual values for each variable during the first **WSE** period. These plots visually highlight the moments where the model's predictions aligned closely with the observed data and where significant deviations occurred.

The anomaly detection plots (Figure 5.51) illustrate the differences between forecasted and actual values, emphasizing the performance of the **ARIMA** model. These plots showcase the discrepancy between predicted and observed values over time. Anomalies were identified by calculating the differences between forecasted and actual values and applying the **IForest** method. A threshold based on the 95th percentile of the anomaly scores was used to flag significant deviations. The shaded regions in the plots indicate the **WSE** period, and also, the detected anomalies are marked as red dots.



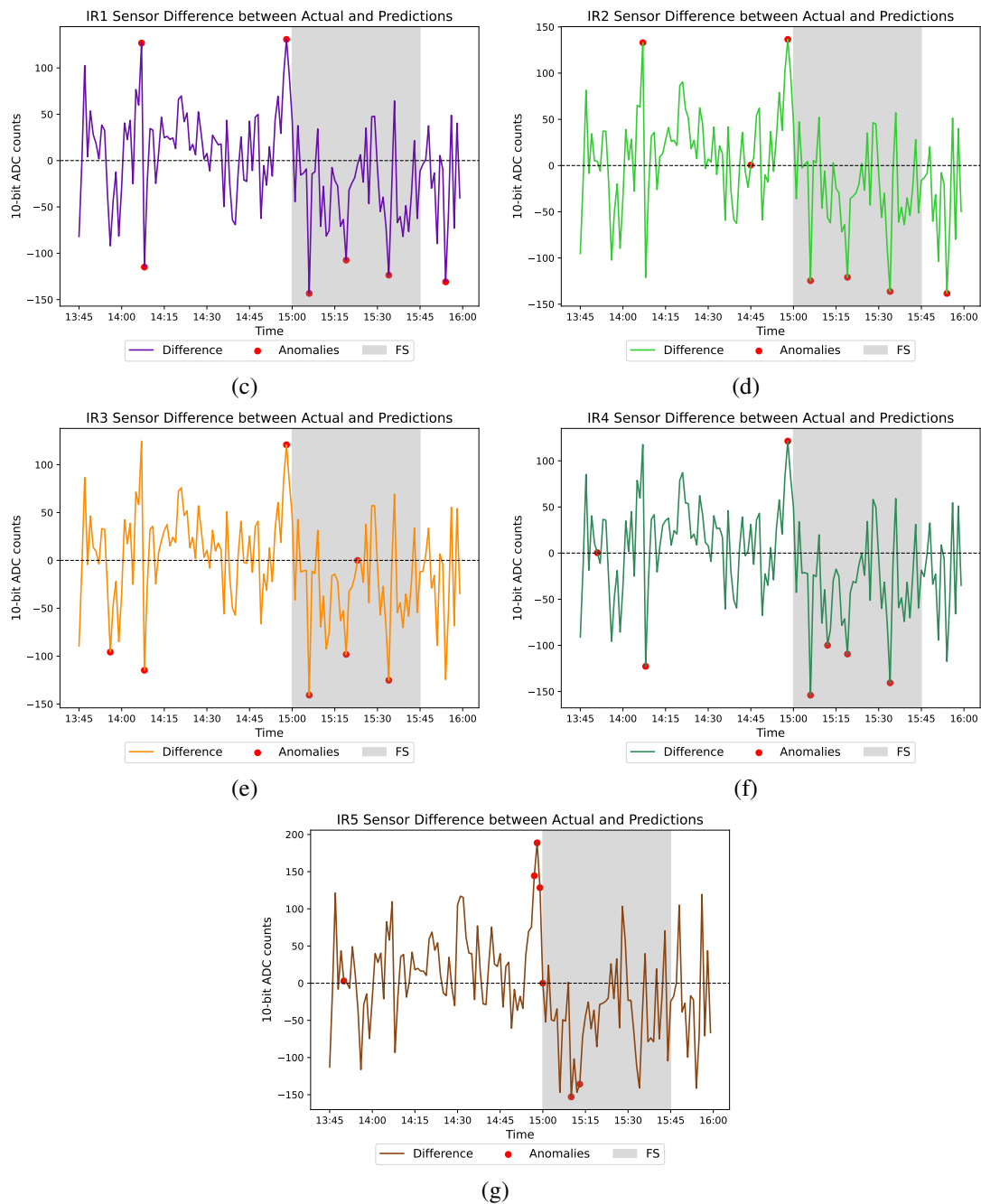


Figure 5.51: Graphical representation of the difference between actual and ARIMA predicted variables' values during the first WSE in Dataset 2. The plots display the actual versus predicted values for Temperature (a), Humidity (b), and IR1 through IR5 (c)-(g).

The Temperature anomaly detection plot shows that anomalies only appear during the **WSE** period, with a total of 7 anomalies detected. This indicates that the **ARIMA** model, combined with the **IForest** anomaly detection method using a 95th percentile threshold, effectively identified fire indications shortly after 15h00. For Humidity, 6 anomalies were detected during the **WSE** period, and one anomaly appeared before the **WSE**. Similar to the Temperature plot, a series of anomalies were identified around 15h15.

On the other hand, all **IR** sensor plots display anomalies not only during the **WSE** but also before and following it. IR1 detected 3 anomalies during the fire event, 3 anomalies before the **WSE**, and 1 anomaly after. IR2 flagged 3 anomalies during the fire, 3 before, and 1 after. IR3 registered 4 anomalies during the fire and 3 before it, with no anomalies detected post-**WSE**. IR4 also showed 4 anomalies during the fire, 3 before, and none afterward. IR5, however, flagged 3 anomalies during the fire but exhibited higher variability, detecting 5 anomalies before the **WSE** and none after.

Additionally, the **IR** sensors detected anomalies near the beginning of the **WSE**, suggesting their high sensitivity to changes in ambient **IR**. This could imply that the **IR** sensors quickly responded to the initial ignition of the fire. Despite this early detection, IR1 through IR5 consistently identified anomalies around 15h00, shortly after the **WSE** started.

5.6.3 RNN

Given the temporal nature of **WSN** data, **RNNs** are well-suited for modeling and predicting sequential patterns. **RNNs** can retain a memory of past data points, which is critical for capturing the dependencies and trends in environmental monitoring data. In this context, this subsection describes an approach to the implementation and evaluation of **RNN** models on Dataset 1 and Dataset 2. The **RNN** models were configured with hyperparameters optimized for the data characteristics through an empirical process of trial and error. This approach allowed for iterative refinements based on observed performance across different configurations. Multiple metrics, including **MAPE**, **MAE**, **MSE**, and **RMSE**, were used to assess and compare model effectiveness, ensuring that the selected hyperparameters yielded the best trade-off between predictive accuracy and generalization.

The default configurations of the **RNN** model included `torch.nn.MSELoss()` as the loss function, the `torch.optim.Adam` optimizer, and a default batch size of 32. Probabilistic forecasts were not used, as the likelihood parameter was left unset, and the `random_state` parameter was not specified, allowing weights initialization to remain random. Additionally, the `training_length` parameter always matched the `input_chunk_length`, and the model's validation loss was evaluated every epoch during training.

Despite these default settings, specific hyperparameters were iteratively adjusted to achieve the best performance based on anomaly detection and evaluation metrics. These included `input_chunk_length`, `hidden_dim`, `n_rnn_layers`, `n_epochs`, and `dropout`. In this context, parameters such as `hidden_dim` and `n_rnn_layers` were fine-tuned for each case to provide sufficient model capacity, avoiding underfitting while capturing the complexities of the **WSN** data. At the same, to mitigate overfitting, the **RNN** model used incorporated dropout regularization, validation loss monitoring, and a rolling window training approach, which ensures exposure to diverse data segments. This method ensures that the models generalize well to unseen data while learning meaningful temporal patterns. For this reason, the selected hyperparameter values will be detailed in the subsequent analyses.

The **RNN** model analysis was performed in low and high **TCD** scenarios, highlighting its adaptability to different environmental conditions. As previously described, both datasets were

resampled to a uniform temporal resolution of 60s. Univariate analysis was applied to the Temperature and Humidity variables, while a multivariate approach was employed for IR1-IR5 due to the high correlation observed among these sensors in both datasets.

A rolling window approach was adopted to account for the dynamic nature of WSN data. Then, each dataset was divided into overlapping train and test windows with a 15 min increment. This iterative method helps evaluate the model's real-time adaptability to changing environmental conditions. Additionally, hyperparameters were individually selected for each dataset and analysis (Temperature, Humidity, and the aggregation IR sensors) to ensure optimal model performance in each case. The predicted values were compared with actual readings for each rolling window, and differences were analysed using the IForest anomaly detection method. A 95th percentile threshold was applied to identify significant deviations that could indicate environmental events such as fire.

5.6.3.1 RNN models for WSN in Low TCD scenario

The RNN models were applied to the WSN data collected in the low TCD scenario. Similar to the RWR and ARIMA analysis, the period for training and testing was set distinctly. The train data started from February 22 at 15h25 to February 23 at 7h30. And the test data started on February 23 at 7h45. This setup allowed the RNN models to be trained on initial data segments and validated iteratively over 15-minute increments, capturing the dynamic nature of the dataset until February 23 11h00.

To achieve optimal performance, hyperparameters were fine-tuned based on the dataset's characteristics and the evaluation metrics. Table 5.12 summarizes the best hyperparameter configurations for each variable analysed.

Table 5.12: Best RNN hyperparameters for variables in Dataset 1.

Variable	Input Chunk Length	Hidden Dim	RNN Layers	Epochs	Dropout
Temperature	100	100	4	100	0.1
Humidity	50	50	1	100	0.1
IR1-IR5	15	130	1	150	0.1

Each variable was analysed individually for Temperature and Humidity using a univariate approach, while a multivariate approach was applied for IR1-IR5 due to their intercorrelations (correlation matrix in Figure 5.18). With a 15min increment, the rolling window technique enabled the evaluation of the models' adaptability to changing environmental conditions, as predictions were iteratively compared to actual readings for each window. These observations are quantitatively supported by the range of metrics presented in Table 5.13, which summarizes the model's performance across variables during the rolling window approach.

Table 5.13: Range Minimum and Maximum of RNN Models Metrics for WSE period in Dataset 1.

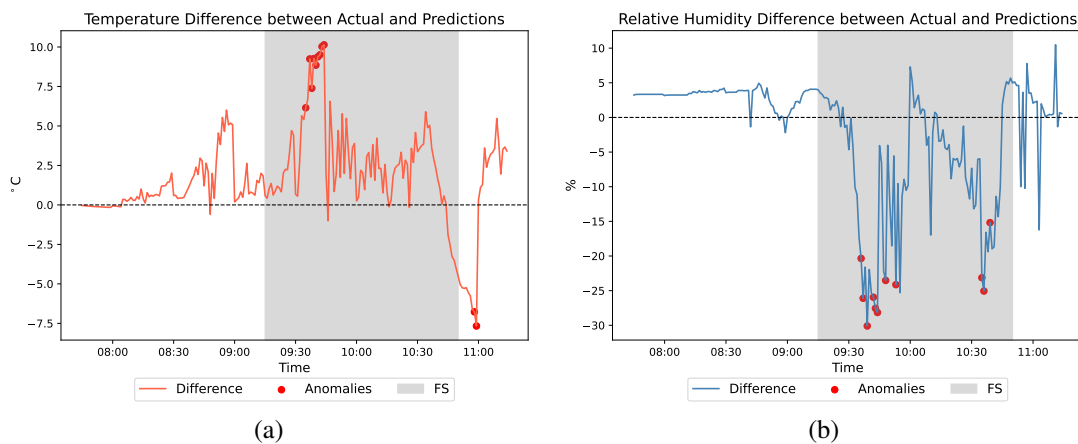
Variable	MAPE [%]	MAE	MSE	RMSE
Temperature	[8.61, 48.00]	[0.11, 6.78]	[0.01, 55.84]	[0.11, 7.47]
Humidity	[2.54, 40.64]	[2.19, 18.31]	[5.65, 425.55]	[2.38, 20.63]
IR1	[11.77, 109.01]	[55.72, 287.55]	[3959.79, 84433.60]	[62.93, 290.57]
IR2	[13.48, 102.61]	[57.53, 301.20]	[6075.73, 92517.74]	[77.95, 304.17]
IR3	[14.19, 133.51]	[61.29, 300.12]	[6098.26, 91784.06]	[78.09, 302.96]
IR4	[16.6, 131.57]	[69.07, 303.73]	[7580.8, 94067.35]	[87.07, 306.70]
IR5	[11.41, 93.35]	[45.51, 297.78]	[3578.42, 90945.24]	[59.82, 301.57]

The metrics highlight the variability in the performance of the **RNN** model across different sensor variables during the **WSE** period. For Temperature and Humidity, the relatively lower ranges of **MAPE**, **MAE**, **MSE**, and **RMSE** indicate that the model effectively captured general trends, even under dynamic conditions. However, the higher end of these metrics – such as the maximum **MAPE** of 48.00% for Temperature and 40.64% for Humidity – reflects the challenges the model faced in accurately predicting abrupt changes, particularly those observed after the **WSE** onset.

The metrics exhibit significantly higher ranges for the **IR** sensors (IR1-IR5), emphasizing the model's difficulty in handling the highly dynamic and noisy patterns. The **MAPE** values for **IR**sensors range from 11.41% to 133.51%, while the **RMSE** reaches up to 306.70, particularly for IR4. These higher values highlight the pronounced variability in forecast errors, probably driven by rapid fluctuations and peaks in the **IR** sensor readings during and immediately after the **WSE**.

The graphical analysis underscores the effectiveness of the **RNN** models and their challenges in predicting sensor readings under low **TCD** conditions, emphasizing patterns and discrepancies between observed and forecasted values. Correspondingly, Appendix D - Figure D.20 illustrates the performance of the **RNN** model with the specified hyperparameters, showcasing the comparison of predicted versus actual values for each variable across the sliding windows.

The impact of these forecasting challenges is further illustrated in the anomaly detection results, as depicted in Figure 5.52. The anomalies were identified (red dots) based on the differences between the actual and predicted values using the **IForest** method with a 95th percentile threshold.



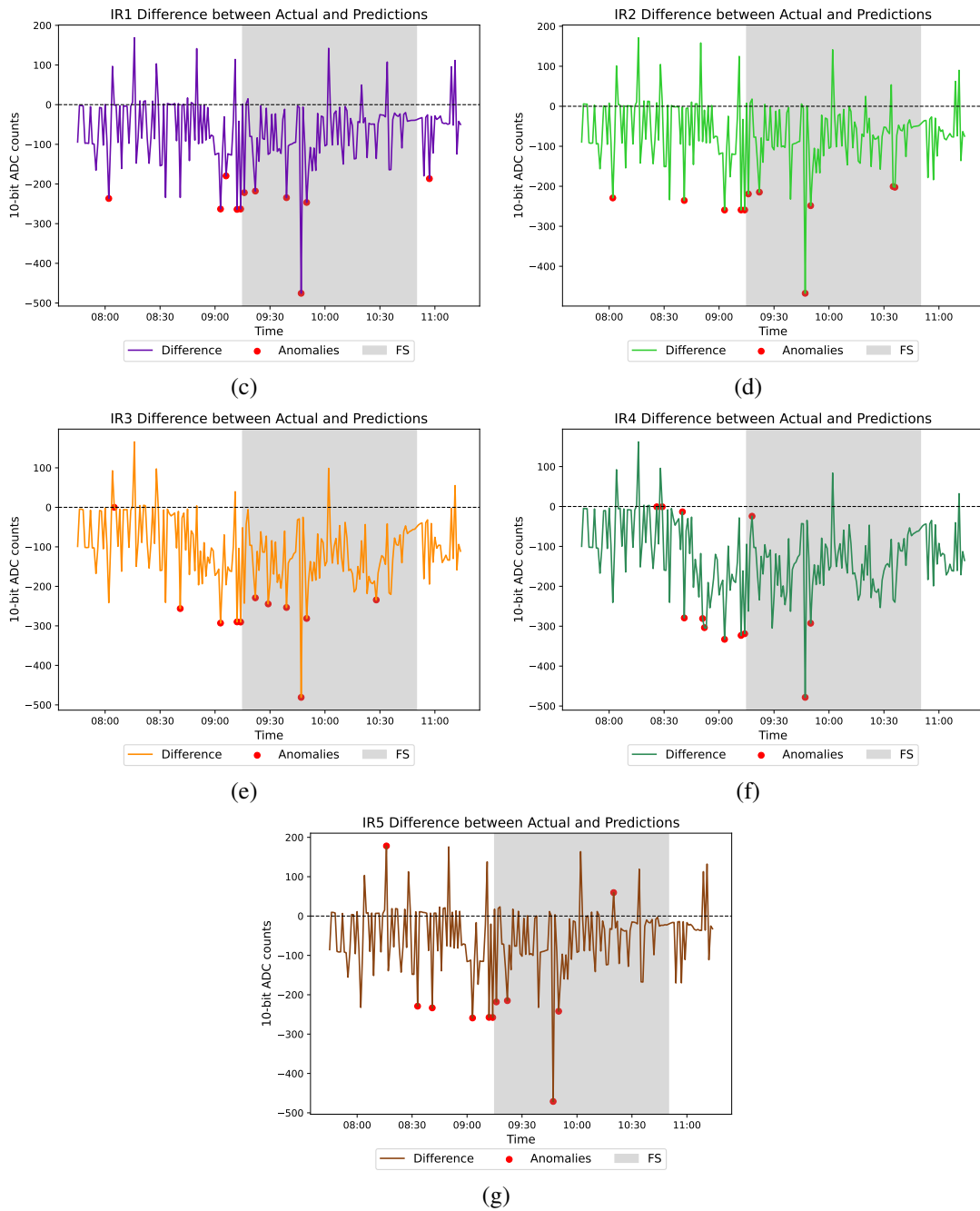


Figure 5.52: Graphical representation of the difference between actual and RNN predicted variables' values in Dataset 1. The plots display the actual versus predicted values for Temperature (a), Humidity (b), and IR1 through IR5 (c)-(g).

For Temperature, a total of 9 anomalies were identified during the **WSE** period, with most occurring shortly after 9h30, highlighting the disparity between the forecasted and actual values. Additionally, 2 anomalies were recorded immediately after the **WSE** ended, reflecting the model's difficulty in readjusting to the non-**WSE** data patterns.

Humidity anomalies were only detected within the **WSE** period, in total 11 anomalies were detected exclusively during the **WSE** period. This indicates that the model effectively identified the

fire-related events, with most anomalies registered shortly after 9h30, further supporting its ability to flag environmental changes during fire scenarios accurately.

For all **IR** sensors (IR1-IR5), anomalies were identified throughout the entire analysis window – before, during, and after the **WSE**. IR1 detected 5 anomalies before the **WSE**, 5 during, and 1 after. IR2 flagged 5 anomalies before and 6 during, with none after the **WSE**. IR3 registered 5 anomalies before and 6 during, with none after the **WSE**. IR4 showed the highest number of anomalies before the event, with 9 anomalies, 3 during, and none after. IR5 displayed 6 anomalies before and 5 during, with no anomalies post-**WSE**. Additionally, around 9h45, all **IR** sensors simultaneously flagged anomalies, which could be advantageous if a strategy is implemented where anomalies across all five sensors are treated as a robust indicator of fire presence.

5.6.3.2 RNN models for WSN in High TCD scenario

The **RNN** model was applied to Dataset 2, which contains the **WSN** data collected in the high **TCD** scenario. Similar to the **RWR** and **ARIMA** approaches with this dataset, intervals of 24h, 48h, and 72h were tested to determine the optimal training window. The metrics indicated that using a 72h interval before **WSE1** achieved the best performance, effectively capturing sufficient historical data to improve the model's predictive capabilities.

The dataset was divided into train and test periods to account for the temporal nature of the data. The first training period, reflecting the 72-hour interval, started on May 9, 13h30 and ended on May 12, 13h30. The first testing period began at 13h45 on May 12, and the rolling window extended until May 12, 15h45. In this sense, the rolling window approach with 15-minute increments was adopted, ensuring dynamic adaptability to evolving environmental conditions during the experiment.

The **RNN** model's performance was guaranteed by fine-tuning hyperparameters based on iterative testing and evaluation. Table 5.14 presents the best-found parameters for Temperature, Humidity, and the aggregated **IR** sensors.

Table 5.14: Best RNN hyperparameters for variables in Dataset 2.

Variable	Input Chunk Length	Hidden Dim	RNN Layers	Epochs	Dropout
Temperature	80	80	2	80	0.3
Humidity	100	50	2	50	0.3
IR1-IR5	30	205	2	55	0.05

These parameters were iteratively adjusted based on the metrics to balance the model complexity and generalization capability, which are summarized in Table 5.15. By adapting the hyperparameters to the specific characteristics of each variable, the **RNN** model indicated its ability to capture temporal dependencies and adapt to dynamic environmental changes during the first **WSE** period. Additionally, Temperature and Humidity parameters were used the univariate approach, while a multivariate approach was applied for IR1-IR5 due to their intercorrelations (correlation matrix in Figure 5.26)

Table 5.15: Range Minimum and Maximum of the RNN Models Metrics for WSE period in Dataset 2.

Variable	MAPE [%]	MAE	MSE	RMSE
Temperature	[1.21, 19.04]	[0.17, 3.48]	[0.03, 12.48]	[0.18, 3.53]
Humidity	[1.59, 16.89]	[1.04, 8.79]	[2.07, 80.25]	[1.44, 8.96]
IR1	[4.06, 66.75]	[17.84, 287.55]	[422.59, 84433.60]	[20.56, 290.57]
IR2	[7.87, 71.49]	[31.98, 301.20]	[1323.00, 92517.74]	[36.37, 304.17]
IR3	[5.34, 71.63]	[22.56, 300.12]	[643.71, 91784.06]	[25.37, 302.96]
IR4	[6.61, 73.81]	[26.98, 303.73]	[959.09, 94067.35]	[30.97, 306.70]
IR5	[7.14, 60.96]	[33.27, 297.78]	[1556.36, 90945.24]	[39.45, 301.57]

Temperature and Humidity exhibit relatively low error ranges, with **MAPE** values starting at 1.21% and 1.59%, respectively, indicating good model performance in capturing general trends under stable conditions. However, the maximum **MAPE** for Temperature reaches 19.04%, while Humidity reaches 16.89%, reflecting the challenges the model faced when predicting sudden environmental changes introduced by the **WSE**. Similarly, the **MAE** and **RMSE** values remain low for both variables in most cases, but the upper limits (3.48 for **MAE** and 8.96 for **RMSE** in Humidity) highlight areas for improvement in handling abrupt variations.

The **IR** sensors (IR1-IR5) reveal significantly higher error metrics, consistent with the complexity of predicting their highly variable and spiky data patterns. For instance, **MAPE** values range from 4.06% to 73.81%, with the most significant errors observed in IR4 and IR5. Similarly, the **MAE** and **RMSE** metrics reach maximum values exceeding 300 for IR2 through IR5, indicating that the model presented some difficulty in predicting these variables during dynamic conditions. IR1, while showing better performance than the other **IR** sensors, still indicates substantial variability, with maximum **MAE** and **RMSE** values of 287.55 and 290.57, respectively.

The visual comparison provides models' effectiveness and difficulties in forecasting sensor readings under high **TCD** conditions, highlighting patterns and potential areas of divergence between predicted and observed values. In this sense, Appendix D - Figure D.21 illustrates the performance with the mentioned hyperparameters of the **RNN** model, comparing the predicted and actual values for each variable across the sliding windows.

The differences between the predicted and actual values for all variables were analysed using the **IForest** anomaly detection method, with a 95th percentile threshold applied to identify significant deviations. In Figure 5.53, the plots illustrate the identified anomalies for each variable, providing visual confirmation of the **RNN** model's performance in detecting abrupt changes in the data patterns.

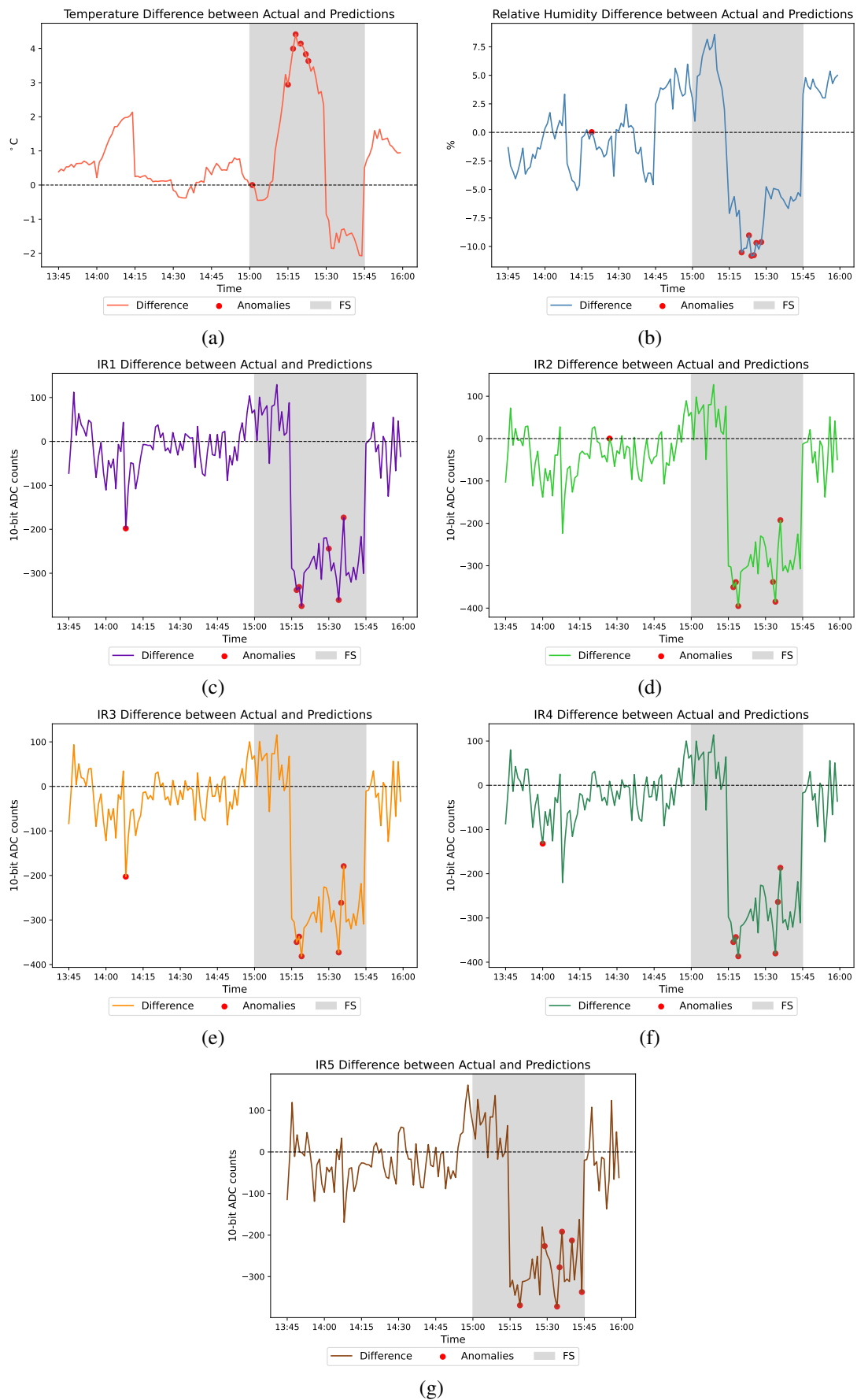


Figure 5.53: Graphical representation of the difference between actual and RNN predicted variables' values in Dataset 2. The plots display the actual versus predicted values for Temperature (a), Humidity (b), and IR1 through IR5 (c)-(g).

The anomalies identified for the Temperature variable are concentrated between 15h15 and 15h30, directly corresponding to the **WSE** period. A total of 7 anomalies were detected during this interval, including one very close to the beginning of the **WSE**. During this time, the model had difficulty accurately predicting abrupt changes, as evidenced by significant deviations between actual and forecasted values. Post 15h45, the model stabilized, with differences falling below $\pm 2^\circ\text{C}$, indicating recovery and improved performance after the event.

Similar behavior is observed for Humidity, where 6 anomalies were detected during the **WSE** period and one anomaly occurred before the event. Before the **WSE**, the differences between actual and forecasted values remained around $\pm 5\%$. However, after 15h00, these differences grew rapidly, peaking at -10% around 15h15. Anomalies persisted until 15h30, after which no further anomalies were detected. Nonetheless, the increased amplitude of differences post-**WSE** suggests that the model's forecasting was influenced by data patterns introduced during the event, as post-**WSE** variability was noticeably higher than pre-**WSE** levels.

For IR1-IR5, 6 anomalies were detected during the **WSE** period for IR1 through IR4, and 7 anomalies for IR5. Additionally, one anomaly was flagged before the **WSE** for IR1-IR4, while IR5 showed no anomalies in this period. Anomalies emerged at exactly 15h15 across all **IR** sensor plots, highlighting the sensors' sensitivity to the ignition of the fire. Following the **WSE**, no anomalies were detected for any of the **IR** sensors, indicating that the model successfully adapted to normal conditions after being trained on data that included the **WSE** period.

5.6.4 TimesFM

This subsection describes the implementation and configuration of **TimesFM**, focusing on its application to Dataset 1 and Dataset 2. The **TimesFM** implementation was based on the documentation provided by Google and the code available in the GitHub repository. Detailed documentation [284] provides guidance on setting parameters and using the model, making it easy to adapt to different data types and contexts. The GitHub [285] repository contains practical examples and the source code needed to implement the model. And it is possible to download a model checkpoint in [286]. Therefore, when starting the `TimesFM()` class, it is necessary to assign some values:

- `context_len`: defines the number of time points in the past that the model considers when making future predictions.
- `horizon_len`: Specifies the number of forecast time points the model should predict. This test was fixed at 15 time steps to align with the forecasting interval.
- `input_patch_len`: it segments the time series into smaller blocks, allowing the model to process the inputs efficiently. It was fixed at 32 to ensure efficient segmentation of the time series data.
- `output_patch_len` represents the number of time points the model predicts at once in each prediction step. The setup used was 128, adhering to the requirement that it be a multiple of 32.

- `num_layers`: it provides adequate modeling capacity, balancing model complexity with performance. Set to 20, ensuring sufficient depth for capturing temporal patterns.
- `model_dims`: allows the model to capture the nuances of temporal data without taxing computational resources. The default is 1280, and it was decided to maintain.
- `backend`: Depending on the available resources, its possible to choose “gpu”, “cpu” or “tpu” to improve computational performance. Here, “cpu” was used.

The `context_len` parameter, critical for defining the temporal scope of the model, was iteratively tested with multiples of 32 up to 512. Given the specific characteristics of **WSN** data, such as daily variations and the diverse types of sensors, it was essential to evaluate this range of `context_len` values while keeping other parameters fixed. This systematic approach aimed to determine whether the pre-trained model could effectively explore the dataset given its historical contexts and adapt to the temporal dependencies inherent in **WSN** data. The outcomes of these configurations are discussed in the subsequent subsections, where the optimal `context_len` values for each dataset and variable are detailed.

The model’s performance was evaluated using metrics such as **MAPE**, **MAE**, **MSE**, and **RMSE**, providing a comprehensive view of its forecasting accuracy. As with the **RWR**, **ARIMA** and **RNN** models, the **TimesFM** was tested using sliding windows with a 15 min increment. The datasets were split into overlapping training and testing windows, ensuring robust evaluation under low and high **TCD** scenarios. Further analyses of the predicted values, their deviations, and detected anomalies are presented in subsequent subsections dedicated to each scenario.

5.6.4.1 TimesFM models for WSN in Low TCD scenario

The analysis of the **TimesFM** model in the low **TCD** scenario followed a similar structure to that employed in the **RWR**, **ARIMA** and **RNN** models, allowing for direct comparisons across methodologies. The data from Dataset 1 was divided into training and testing periods. The training data included observations from from February 22 at 15h25 to February 23 at 7h30. Testing data began February 23 at 7h45 and extended to 11 h00 on the same day, with predictions generated for 15-minute intervals.

The optimal `context_len` values identified for this dataset changed across the variables, reflecting the distinct temporal dependencies within the **WSN** data. For Humidity, the best `context_len` was found to be 256, while Temperature performed optimally with a value of 288. Regarding the **IR** sensors, the ideal `context_len` values were as follows: 512 for IR1, 256 for IR2, 416 for IR3, 160 for IR4, and 32 for IR5.

These context lengths were determined through iterative testing and evaluation to maximize the **TimesFM** model’s accuracy. The model’s performance, reflecting the differences between predicted and actual values, is quantitatively summarized in Table 5.16. The table presents the range of metrics for each variable during the **WSE** period, highlighting the variability in the model’s predictive capabilities across different types of data.

Table 5.16: Range Minimum and Maximum of the TimesFM Models Metrics for WSE period in Dataset 1.

Variable	MAPE [%]	MAE	MSE	RMSE
Temperature	[0.04, 0.67]	[0.1, 6.09]	[0.02, 43.3]	[0.14, 6.58]
Humidity	[0.0, 0.81]	[0.02, 35.52]	[0.0, 1386.45]	[0.02, 37.24]
IR1	[0.1, 0.98]	[45.28, 184.31]	[3715.22, 40705.54]	[60.95, 201.76]
IR2	[0.07, 0.9]	[32.47, 129.74]	[2548.93, 22738.65]	[50.49, 150.79]
IR3	[0.12, 0.91]	[41.13, 121.28]	[3603.38, 20904.65]	[60.03, 144.58]
IR4	[0.11, 0.78]	[42.76, 129.27]	[2965.52, 22829.19]	[54.46, 151.09]
IR5	[0.08, 0.73]	[39.4, 86.91]	[2831.72, 15142.23]	[53.21, 123.05]

For Temperature and Humidity, the **MAPE** values remain low, starting at 0.04% and 0.00%, respectively. These results indicate the model's ability to effectively handle smoother and less volatile variables. However, during the **WSE**, the upper bounds of the error metrics rise significantly, with Temperature reaching a maximum **MAPE** of 0.67% and Humidity at 0.81%. This reflects the model's difficulty in accurately predicting abrupt changes introduced by the fire event. Additionally, the maximum **MAE** for Humidity is 35.52, and the **RMSE** peaks at 37.24, underscoring the increased forecasting errors during this period.

The metrics for the **IR** sensors (IR1-IR5) reflect the challenges posed by these variables' more complex and spiky nature. The **MAPE** values range from 0.07% to 0.98%, significantly higher than those for Temperature and Humidity. Additionally, the maximum **MAE**, **MSE**, and **RMSE** values further emphasize the difficulties in predicting these variables. For instance, IR1 shows a maximum **RMSE** of 201.76, highlighting substantial errors during rapid fluctuations, particularly during the **WSE** period. IR2 and IR4 also display significant error ranges, with maximum **RMSE** values of 150.79 and 151.09, respectively.

In alignment with this, Appendix D - Figure D.22 presents the performance of the **TimesFM** model using the specified hyperparameters for each variable, offering a detailed comparison of actual and forecasted values across the rolling windows for each variable.

The anomaly detection process further analyses the differences between predicted and actual values. Using the **IForest** algorithm with a 95th percentile threshold, anomalies were identified during the **WSE**, emphasizing the significant deviations caused by the fire event. The anomaly detection plots in Figure 5.54 highlight the periods of rapid change, clearly indicating the **TimesFM** model's predictive challenges during these critical intervals.

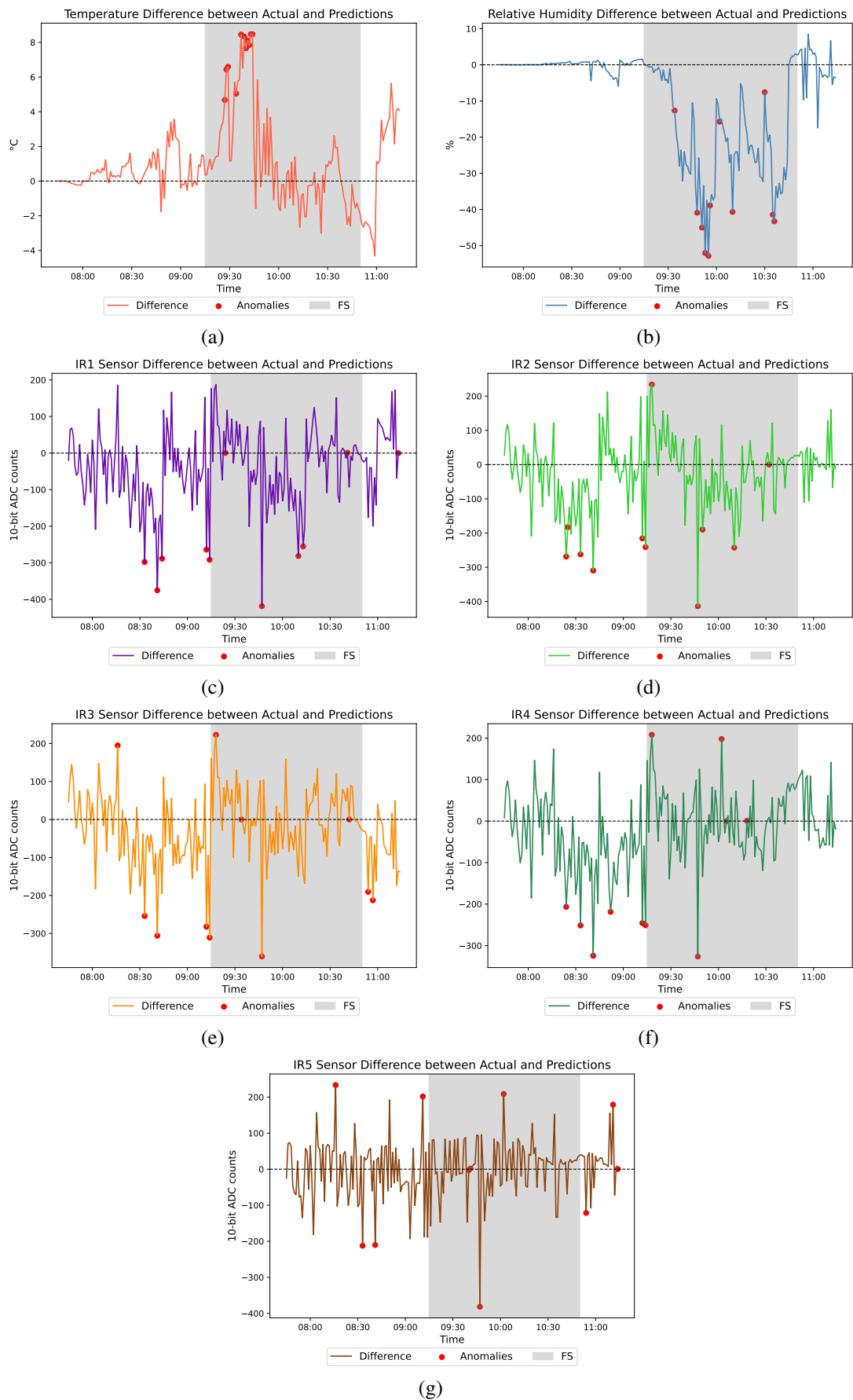


Figure 5.54: Graphical representation of the difference between actual and TimesFM predicted variables' values in Dataset 1. The plots display the actual versus predicted values for Temperature (a), Humidity (b), and IR1 through IR5 (c)-(g).

The anomaly detection plots for the differences between predicted and actual values reveal the behavior of the **TimesFM** model during the **WSE**. For Temperature, 11 anomalies were flagged exclusively during the **WSE** execution period. These anomalies appear as red flags between 9h30 and 9h45, highlighting the model's challenge in adapting to abrupt temperature changes introduced by the fire event. Similar to Temperature, Humidity anomalies are primarily detected during the **WSE**. Similarly, Humidity anomalies, also totaling 11 during the **WSE**, extend over a longer period, starting at approximately 9h30 and continuing until shortly after 10h30. This indicates the model's sensitivity to sustained environmental changes in this variable, with no anomalies flagged before or after the **WSE** for either variable.

The anomaly plots for the **IR** variables (IR1-IR5) exhibit a consistent pattern. IR1 flagged 5 anomalies during the **WSE**, 5 before, and 1 after. IR2 registered 5 anomalies during the **WSE**, 6 before, and none after. IR3 flagged 4 anomalies during the **WSE**, 5 before, and 2 after. IR4 identified 5 anomalies during the **WSE**, 6 before, and none after. IR5 recorded 4 anomalies during the **WSE**, 4 before, and 3 after. Anomalies were detected almost at the start of the **WSE**, which can be attributed to the sensitivity of the sensors to abrupt changes in **IR** intensity caused by lighting the fire. After the **WSE**, anomalies are observed at different time intervals across all plots. Notably, a period around 9h45 stands out where all **IR** sensors simultaneously flag anomalies, reinforcing the potential of a multi-sensor anomaly strategy for identifying critical events.

An essential observation is that the pre-trained nature of the **TimesFM** model ensures that **WSE** data does not influence its learning. Consequently, across all plots, predicted values tend to return closely matching actual values once the **WSE** concludes, highlighting the model's ability to maintain prediction accuracy post-event.

5.6.4.2 TimesFM models for WSN in High TCD scenario

The analysis of the **TimesFM** model in the high **TCD** scenario mirrors the methodology employed in the low **TCD** scenario, ensuring consistency and comparability between the two settings. Dataset 2 was utilized for this analysis, with different rolling window sizes tested to optimize the **TimesFM** model's performance. Specifically, rolling windows of 24h, 48h, and 72h were evaluated, with the 24h window achieving the best performance.

With the selected 24h window, the first training period started from May 11 at 13h00 to May 12 at 13h30. The first testing period began at 13h45 on May 12. Both train and test data extended until May 12 at 15h45, following a rolling window approach with increments of 15 min to adapt to the dynamic nature of the dataset.

Specific hyperparameters were iteratively adjusted to fit in with distinct temporal patterns and environmental variations characteristic of the high **TCD** scenario. These configurations aimed to enhance the model's forecasting performance under complex environmental conditions. The best-performing `context_len` values for each variable in this scenario were as follows: 96 for Temperature, 480 for Humidity, 480 for IR1-IR2, 512 for IR3, 64 for IR4, and 96 for IR5. The observed differences between actual and predicted values across all variables are quantitatively summarized in the range of metrics presented in Table 5.17.

Table 5.17: Range Minimum and Maximum of the TimesFM Models Metrics for first WSE period in Dataset 2.

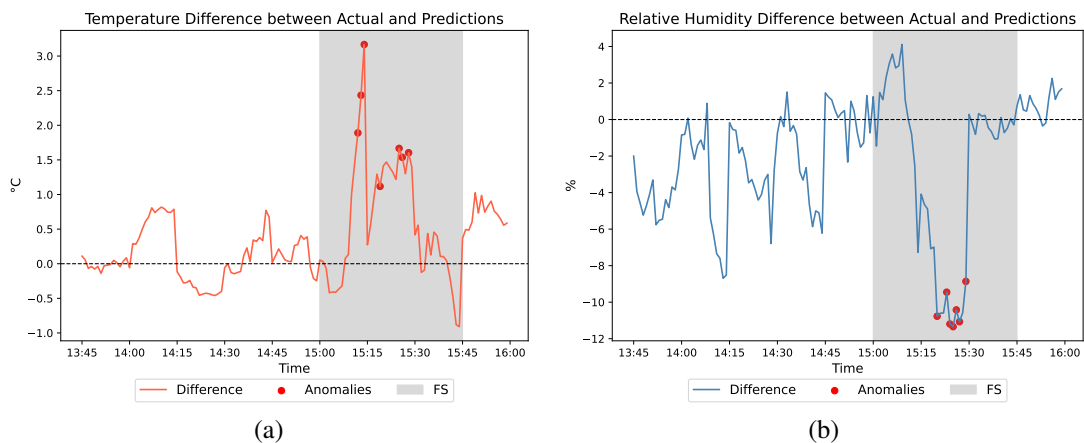
Variable	MAPE [%]	MAE	MSE	RMSE
Temperature	[0.0, 0.07]	[0.05, 1.23]	[0.0, 1.66]	[0.06, 1.29]
Humidity	[0.01, 0.17]	[0.46, 8.84]	[0.31, 84.45]	[0.56, 9.19]
IR1	[0.04, 0.12]	[19.77, 48.64]	[667.26, 3467.18]	[25.83, 58.88]
IR2	[0.06, 0.12]	[26.03, 42.35]	[1116.37, 3221.76]	[33.41, 56.76]
IR3	[0.05, 0.12]	[20.84, 46.67]	[723.75, 3500.0]	[26.9, 59.16]
IR4	[0.05, 0.15]	[23.13, 49.71]	[995.39, 4242.14]	[31.55, 65.13]
IR5	[0.05, 0.12]	[23.82, 60.57]	[882.62, 5927.25]	[29.71, 76.99]

The metrics indicate the variability in the **TimesFM** model's forecasting accuracy across different variables during the high **TCD** scenario. For Temperature and Humidity, the relatively low range of **MAPE** values, starting at 0.0% and peaking at 0.07% for Temperature and 0.17% for Humidity, indicates good performance in capturing general trends outside the **WSE**. However, higher maximum **MAE** and **RMSE** values, particularly during the **WSE**, reflect the challenges in predicting abrupt changes, with Humidity showing notable deviations caused by the sudden drop in readings during the event.

For the **IR** sensors (IR1-IR5), the broader ranges for all metrics, particularly **MAE** and **RMSE**, underscore the difficulty in forecasting their volatile and complex data patterns. For example, the maximum **RMSE** values range from 58.88 for IR1 to 76.99 for IR5, emphasizing the challenges posed by the **WSE**. These results highlight the need for model refinements to better handle the dynamic and noisy nature of **IR** sensor data in high **TCD** scenarios.

The comparison between actual and forecasted values under low **TCD** conditions is presented in Appendix D - Figure D.23. The plots illustrate the performance of the **TimesFM** model using the specified hyperparameters, providing a detailed comparison of actual and predicted values across the rolling windows for each variable.

The differences between actual and predicted values were passed through an anomaly detection process using the **IForest** algorithm with a 95th percentile threshold to understand these discrepancies better. This approach identifies significant deviations during the **WSE**, where abrupt changes can occur. The resulting anomalies are visualized in Figure 5.55, showing the model's detection capabilities and performance under high **TCD** conditions.



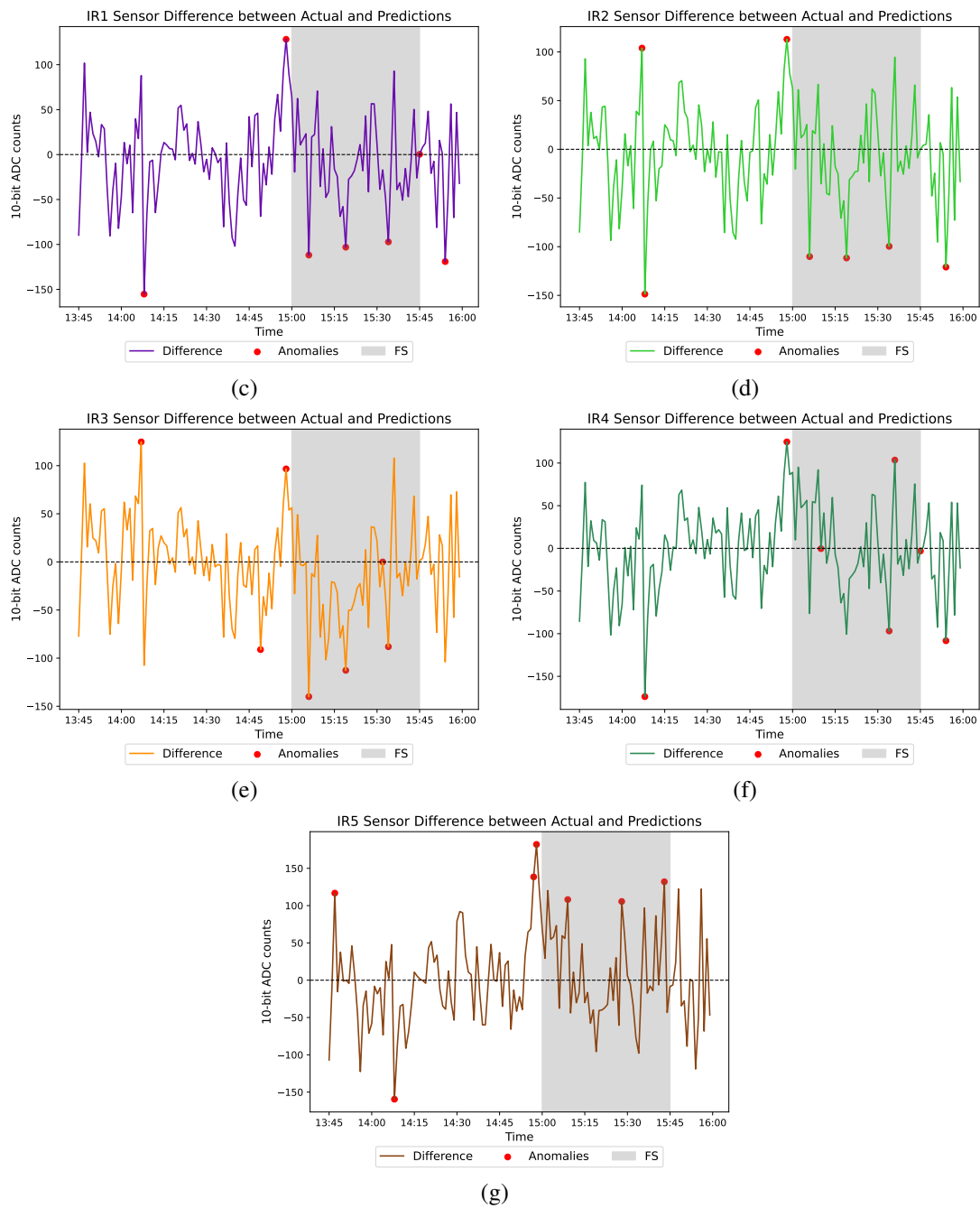


Figure 5.55: Graphical representation of the difference between actual and TimesFM predicted variables' values during first WSE in Dataset 2. The plots display the actual versus predicted values for Temperature (a), Humidity (b), and IR1 through IR5 (c)-(g).

The anomaly detection plots for the differences between actual and predicted values reveal distinct behaviors across the variables. For Temperature, anomalies were detected exclusively during the **WSE**, with a total of 7 anomalies identified. Before the **WSE**, differences remained minimal, under 1°C . However, around 15h15, a significant increase in differences was observed, leading to the identification of anomalies. These persisted until approximately 15h30, reflecting the

model's difficulty in adapting to the sudden environmental changes introduced during the fire event.

Similarly, for Humidity, 7 anomalies were flagged during the **WSE** period, with no anomalies detected before or after the event. Differences between actual and predicted values stayed within $\pm 4.5\%$ before the **WSE**. The first anomaly was flagged around 15h15 due to a rapid deviation in values, and anomalies continued consistently until 15h30, mirroring the behavior observed in the Temperature variable.

Notably, around 15h15, all **IR** sensors (IR1-IR4) simultaneously flagged anomalies, indicating a collective response to the ignition of the fire. This behavior reinforces the potential utility of combining anomaly signals across sensors to improve the reliability of fire detection. For the **IR** sensors, anomalies were detected across the analysis window: IR1 flagged 4 anomalies during the **WSE**, 2 before, and 2 after. IR2 identified 3 anomalies during, 3 before, and 1 after. IR3 registered 4 anomalies during and 3 before, with none after. IR4 displayed 4 anomalies during, 2 before, and 2 after. IR5 flagged 3 anomalies during, 4 before, and none after. After the **WSE**, anomalies became sporadic across all **IR** sensors, reflecting residual challenges in the model's predictions as it adapted to the post-**WSE** conditions.

5.7 Combination of Techniques

Building on the outputs from the individual analyses of **RWR**, **ARIMA**, **RNN**, and **TimesFM**, this section explores the potential advantages of combining these techniques to improve the detection and prediction of anomalies in **WSN** data. The primary objective is to complement the strengths of each method, addressing their individual limitations and enhancing the overall performance of anomaly detection, particularly during abrupt environmental events such as fires.

Each method presented specific strengths: **RWR** effectively captured linear patterns but had difficulty with non-linear changes; **ARIMA** excelled in stationary time series but faced challenges with dynamic shifts; **RNN** provided robust modeling of temporal dependencies but exhibited variability under extreme conditions; and **TimesFM**, as a pre-trained transformer-based approach, showed strong adaptability across datasets. Therefore, combining these models aims to achieve a more balanced performance that integrates their respective advantages.

This analysis maintains the distinction between the low and high **TCD** scenarios, ensuring that the combined approach can cover different forest densities. For each scenario, the anomaly outputs of the individual models are aggregated, and the majority voting for anomaly detection is explored [287]. In this sense, two approaches are applied: unweighted and weighted votes. The first approach intends to verify **True** values from the anomaly output aggregated data at the same timestamp for each variable. The second approach is similar to the first but with weight values for Temperature, Humidity and IR1-IR5. These strategies aim to optimize the trade-off between sensitivity and specificity in anomaly detection, particularly during fire events.

Furthermore, comparing weighted and unweighted approaches aims to explore strategies to reduce false positives while maintaining sensitivity to relevant anomalies. In the previous section, it was observed that the **IR** sensors frequently generated anomalies in periods without **WSE**. However,

the five **IR** sensors indicated high sensitivity, consistently flagging anomalies near the onset of fire. Therefore, a weighted approach was implemented to account for this information without disregarding it entirely.

In this weighted strategy, higher weights were assigned to the Temperature and Humidity variables, as these consistently presented anomalies during **WSE** periods. This weighting reflects the reliability of these variables in detecting critical events. At the same time, despite their sensitivity, the **IR** sensors were assigned lower weights due to their higher incidence of false positives. Consequently, the weighted strategy seeks to balance the contributions of the **IR** sensors while reducing the impact of false positives, thereby improving the precision and relevance of the combined model.

Additionally, the analysis focuses on three distinct time intervals for both low and high **TCD** scenarios: pre-fire, during fire, and post-fire. These intervals enable a detailed assessment of the combined model's effectiveness in detecting anomalies associated with abrupt environmental changes. Furthermore, the outputs are visualized through time series plots, highlighting the aggregated anomaly scores and the corresponding fire intervals.

5.7.1 Low TCD Scenario: Combining Techniques

In the low **TCD** scenario, the combination of anomaly detection techniques applied to Dataset 1 – analysed using **RWR**, **ARIMA**, **RNN**, and **TimesFM** – was implemented to evaluate the effectiveness of aggregated outputs in detecting anomalies. This approach builds upon the outputs of each individual model and introduces two aggregation strategies: unweighted and weighted voting. The objective is to enhance anomaly detection precision by combining the strengths of each model while addressing their limitations, particularly in environments with sparse tree cover.

The unweighted aggregation strategy involved summing the outputs of the four models for each variable at the same timestamp. The aggregated sum was normalized, enabling thresholds to be established for identifying strong anomalies during the **WSE**. This approach signifies that anomalies flagged concurrently by multiple methods are considered more significant.

In contrast, the weighted aggregation strategy assigned specific weights to each variable based on their observed behavior. Higher weights were given to the Temperature and Humidity variables due to their consistent anomaly patterns during the fire event. Conversely, lower weights were applied to the **IR** sensors, considering their higher sensitivity but frequent false positives. Specifically, a weight of 10 and 5 were assigned to anomalies in Temperature and Humidity, respectively, across all models, while a weight of 0.5 was applied to anomalies in **IR** sensor data. These weight values were determined through trial and error, experimentally adjusting them to achieve the best balance in anomaly aggregation. Figure 5.56 illustrates the outputs of both aggregation techniques.

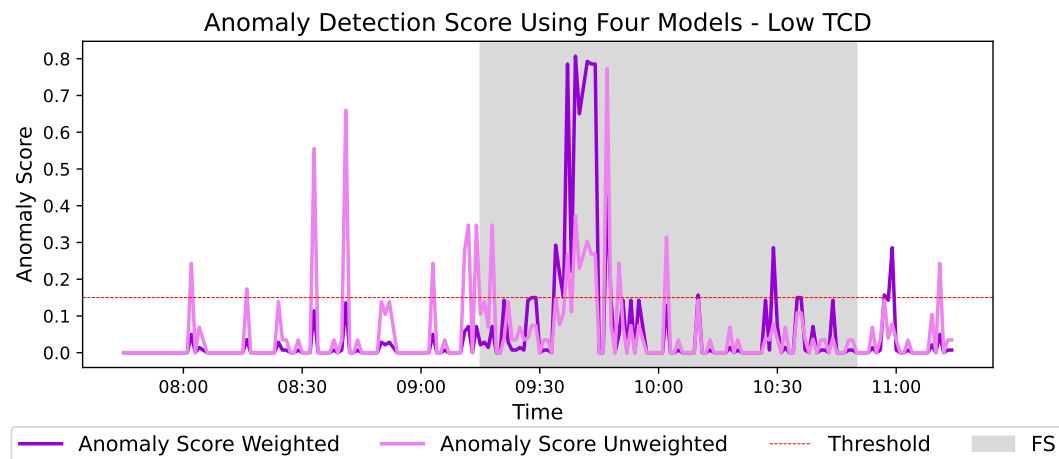


Figure 5.56: Aggregated Anomaly Scores from Dataset 1 (Low TCD Scenario) Using Unweighted and Weighted Strategies.

The plot illustrates the anomaly detection scores in the low **TCD** scenario, comparing unweighted and weighted aggregation strategies. The weighted approach prioritizes contributions from Temperature and Humidity anomalies, effectively reducing the influence of false positives from **IR** sensors. This adjustment results in a more precise identification of anomalies during the **WSE** interval (marked by **FS**), closely aligning with the expected environmental changes caused by the fire.

The distribution of anomaly scores suggests that the most suitable detection threshold is greater than 0.15. The count of anomalies exceeding this threshold further reveals the impact of the weighting strategy on anomaly detection. During the **WSE**, the weighted approach identified 13 anomalies, compared to 11 for the unweighted approach (both above of threshold). Notably, the weighted strategy reduced false alerts significantly, with zero anomalies detected before the **WSE** and only two after it. In contrast, the unweighted approach registered 8 anomalies before the **WSE** and one anomaly afterward, reflecting the heightened sensitivity and false positives predominantly from **IR** sensors.

The first anomalies detected during the **WSE** for each strategy occurred at the following times:

- **Weighted Approach:** at 9h34 with a score of 0.29, at 9h35 with a score of 0.21, and at 9h37 with a score of 0.79.
- **Unweighted Approach:** at 9h18 with a score of 0.35, at 9h37 with a score of 0.27, and at 9h39 with a score of 0.37.

The weighted method presents greater effectiveness in isolating fire-related anomalies, ensuring a more reliable detection process while minimizing noise from non-fire events. However, the weighted strategy exhibited a slightly delayed response compared to the unweighted approach, which identified anomalies earlier. Over extended periods, such as days or weeks, the unweighted strategy's higher sensitivity could lead to an accumulation of false positives, potentially impacting the long-term reliability and performance of the **WSN** in detecting forest ignition events.

5.7.2 High TCD Scenario: Combining Techniques

In the high **TCD** scenario, the combination of anomaly detection techniques from Dataset 2 analysed by **RWR**, **ARIMA**, **RNN**, and **TimesFM** was implemented to evaluate the effectiveness of aggregated outputs in detecting anomalies. Similar to the low **TCD** scenario, this process explores two aggregation strategies: unweighted and weighted voting. The objective is to adapt anomaly detection techniques to denser forest environments, addressing the complexities and challenges of high tree density.

The unweighted aggregation strategy sums the outputs of the four models for each variable at the same timestamp. A normalization process is applied, and thresholds can be established to identify anomalies during the **WSE** period. This approach allows for identifying anomalies where multiple models converge, highlighting moments of significant environmental deviations.

The weighted aggregation strategy introduces specific weights for each variable, considering their observed behavior in the high **TCD** environment. As in the low **TCD** scenario, Temperature and Humidity anomalies are given higher weights due to their consistent performance during the fire event, while lower weights are applied to **IR** sensors to balance their high sensitivity and tendency for false positives. In this scenario, weights of 10 and 5 were used for Temperature and Humidity, respectively, while a weight of 0.5 was applied to **IR** sensor anomalies. These weight values were determined through trial and error, experimentally adjusting them to achieve the best balance in anomaly aggregation. Figure 5.57 illustrates the aggregated anomaly scores for both strategies.

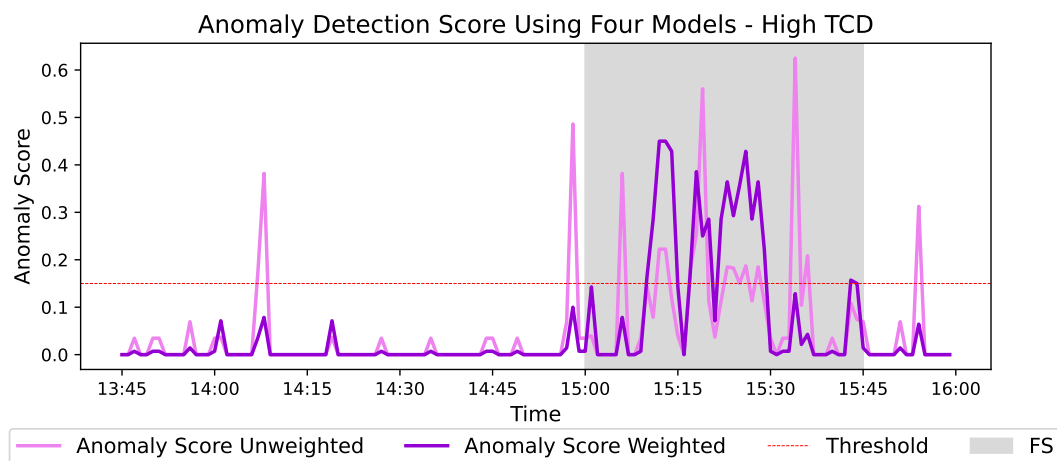


Figure 5.57: Aggregated Anomaly Scores from Dataset 2 (High TCD Scenario) Using Unweighted and Weighted Strategies.

The anomaly detection scores for the high **TCD** scenario, as illustrated in the plot, highlight the differences between unweighted and weighted aggregation strategies. By assigning higher weights to Temperature and Humidity anomalies, the weighted approach effectively decrease the impact of the **IR** sensors' frequent false positives. This adjustment results in a clearer detection of anomalies within the **WSE** interval (marked by **FS**), corresponding to the fire event. Conversely, the

unweighted scores maintain a higher baseline before and after the **WSE**, driven by the heightened sensitivity of the **IR** sensors.

The distribution of anomaly scores indicates that a detection threshold above 0.15 is more suitable, providing a basis for evaluating the performance of the aggregation strategies. Above this threshold, the weighted approach identified 18 anomalies during the **WSE**, compared to 13 for the unweighted method. Notably, the weighted strategy eliminated false positives entirely, with zero anomalies detected both before and after the **WSE**. In contrast, the unweighted approach recorded 3 anomalies before and 1 anomaly after the **WSE**, reflecting the **IR** sensors' susceptibility to noise. The first three anomalies detected during the **WSE** for each strategy occurred at the following times:

- **Weighted Approach:** at 15h10 with a score of 0.16, at 15h11 with a score of 0.28, and at 15h12 with a score of 0.45.
- **Unweighted Approach:** at 15h06 with a score of 0.38, at 15h12 with a score of 0.227, and at 15h13 with a score of 0.22.

The weighted strategy has a slightly delayed response compared to the unweighted method, with its first detection occurring four minutes later. However, this delay is counterbalanced by its ability to reduce false positives, providing a more reliable identification of fire-related anomalies. The unweighted strategy reacts faster but is more susceptible to noise from non-fire events. This trade-off underscores the importance of selecting a strategy that balances immediate detection with long-term reliability, particularly in high **TCD** environments where precise anomaly detection is crucial.

5.8 Testing and Validating ML Strategies for Forest Fire Scenarios

This section evaluates the effectiveness of combining **RWR**, **ARIMA**, **RNN**, and **TimesFM** in identifying the onset of wildfires. Building on the analyses from Dataset 1 and Dataset 2, which were conducted in controlled test scenarios with low and high **TCD**, respectively, this section applies the same methodology to three additional datasets: Dataset 3, Dataset 4, and Dataset 5. These datasets represent scenarios closer to real wildfire events, offering a more robust validation of the proposed approach in practical applications.

- **Dataset 3:** Collected in September, a month characterized by high temperatures, low humidity, and increased risk of wildfires. The environmental conditions in this dataset are representative of high-risk periods for forest fires, making it a valuable test case for the model's ability to respond to real-world wildfire precursors.
- **Dataset 4:** Captured during a controlled burn organized by firefighting authorities to manage vegetation in a low **TCD** environment. This dataset provides an opportunity to test the combined model's capacity to detect fire-related anomalies in a scenario with sparse vegetation, simulating real-world conditions with reduced forest density.

- **Dataset 5:** Also obtained during a controlled burn, but in a high **TCD** environment. This dataset is essential for assessing the combined model's performance in densely forested areas, where fire detection can be more challenging due to the complex interaction of environmental variables.

For all three datasets, the process described in Section 5.6 was applied, with adaptations to reflect the distinct characteristics of each dataset. The analysis considered the respective **TCD** levels of each scenario, and the hyperparameters identified in earlier sections were reused for consistency and optimization. Additionally, the analysis includes a comparative evaluation of unweighted and weighted aggregation strategies, as previously demonstrated, to further refine the anomaly detection process and minimize noise from false positives.

5.8.1 Applying the approach in Dataset 3

Dataset 3 was collected under high wildfire risk conditions, typical during the summer months. According to local firefighting authorities, conducting field tests during these periods is considered highly unsafe due to the elevated risk of fire spread. Despite these challenges, the firefighters authorized and supervised a controlled test on the campus of **IPB**. This enabled the collection of valuable data during the summer months, capturing fire-related anomalies through a **WSN** installed at four points.

The dataset was collected over 16 days, with a central bonfire serving as the primary heat source during a **WSE**. Four modules, oriented toward the bonfire, were continuously operational during this period. The modules were equipped with the **DROWSIER** filter, ensuring reliable communication and data collection. In this sense, each module transmitted data at least once every 15 min under stable conditions. In the presence of abrupt changes in the data, transmissions occurred every minute.

A forward-filling methodology was applied to regularize the dataset's temporal resolution. This ensured that each minute within the dataset was populated with the most recent recorded value. For example, if a module transmitted data at 12h01 and the next transmission occurred at 12h05, the values from 12h01 were carried forward to fill the intervening minutes. Importantly, no data overlaps or omissions occurred, as no four modules transmitted data at the same minute.

RWR

Following data regularization, **ML** techniques were applied to Dataset 3, adhering to the same methodology outlined in Section 5.6. Initially, the **RWR** model was implemented using the same lag intervals and training-testing window structure as Dataset 2: training window with 24h, and forecast horizon and incremental training-testing with 15 min. Despite using the established hyperparameters from Dataset 2, the performance metrics indicated not satisfactory results for Dataset 3. Consequently, new lag values were identified based on the dataset's characteristics, optimizing the model's predictive accuracy. The best lag values identified for Dataset 3 were 15 for Temperature, 13 for Humidity, and 5, 5, 8, 4, and 8 for IR1-IR5, respectively. The training process

began on September 20 at 10h00 and ended on September 21 at 10h30, with test data starting at 10h45. The train and test window extended until 12h00 on the same day. Table 5.18 presents the range of the metrics obtained with the refined lags, indicating the performance of the **RWR** model on Dataset 3.

Table 5.18: Range Minimum and Maximum of RWR Models Metrics for WSE in Dataset 3.

Variable	MAE	MedAE	Explained Variance	MSE	RMSE
Temperature	[0.3487, 2.2862]	[0.3572, 2.3273]	[-0.9235, 0.3307]	[0.1561, 5.3466]	[0.3951, 2.3123]
Humidity	[1.0865, 6.0462]	[1.0199, 6.1421]	[-13.7562, 0.7935]	[1.8221, 36.6481]	[1.3498, 6.0538]
IR1	[13.7060, 50.3652]	[10.4243, 42.6757]	[-58.6944, 0.3890]	[297.5039, 3728.2502]	[17.2483, 61.0594]
IR2	[5.8234, 11.6173]	[5.7870, 11.4294]	[-24.0419, -1.4320]	[46.9416, 151.8969]	[6.8514, 12.3246]
IR3	[6.2556, 32.2501]	[7.3256, 30.4046]	[-15.6597, -0.3766]	[50.7171, 1055.6196]	[7.1216, 32.4903]
IR4	[35.1208, 176.6719]	[31.0603, 178.5404]	[-7.1871, 0.2175]	[1897.3396, 36766.3665]	[43.5585, 191.7456]
IR5	[7.5118, 125.7283]	[9.4879, 119.2089]	[-41.1244, 0.5455]	[73.0288, 17053.7274]	[8.5457, 130.5899]

The range of the metrics indicate that the **RWR** model performs adequately for Temperature and Humidity, with relatively low **MAE** and **RMSE** values, suggesting its ability to capture general trends before the **WSE** conditions. However, the **IR** sensors (IR1-IR5) exhibit significantly higher error metrics and negative explained variance values, highlighting the model's difficulty in accurately predicting these variables' spiky and dynamic patterns during the fire event.

ARIMA

The **ARIMA** model was initially applied to Dataset 3 using the same configuration as Dataset 2. This included a rolling training window of 48h with a forecast horizon of 15 min, followed by incremental updates to the training-testing windows every 15 min. The AutoARIMA configuration, optimized for Dataset 2, was also maintained for consistency.

However, the initial results were unsatisfactory, as two variables, IR1 and IR5, failed to exhibit anomalies during the **WSE**, indicating that the model has difficulty capturing relevant patterns for these sensors under the applied configuration. To address this limitation, the training window was reduced to 24h, starting on September 20 at 10h00 and ending on September 21 at 10h30, with test data beginning at 10h45. The analyses was extended until 12h00 on September 20. This adjustment aimed to enhance the model's responsiveness to the dynamic conditions of the fire event.

This adjustment improved performance metrics, highlighting the model's enhanced ability to detect anomalies across variables. The revised results for the **ARIMA** model on Dataset 3, using the 24h training interval, are summarized in Table 5.19.

Table 5.19: Range Minimum and Maximum of ARIMA Models Metrics for WSE in Dataset 3.

Variable	MAPE [%]	MAE	MSE	RMSE
Temperature	[2.69, 12.51]	[0.41, 2.25]	[0.2, 5.16]	[0.45, 2.27]
Humidity	[0.83, 10.17]	[0.56, 6.71]	[0.57, 49.18]	[0.75, 7.01]
IR1	[31.35, 85.72]	[12.44, 68.16]	[210.8, 5394.02]	[14.52, 73.44]
IR2	[7.90, 77.38]	[32.33, 84.50]	[2121.61, 14842.95]	[46.06, 121.83]
IR3	[2.29, 59.63]	[1.0, 27.61]	[1.61, 780.96]	[1.27, 27.95]
IR4	[16.2, 248.36]	[36.89, 173.97]	[3413.29, 33123.77]	[58.42, 182.0]
IR5	[4.33, 271.99]	[1.95, 123.1]	[10.61, 15935.98]	[3.26, 126.24]

The range of the metrics indicates the variability in the **ARIMA** model's performance across different variables in Dataset 3 after applying the adjusted 24h training window. Temperature and Humidity exhibited relatively low **MAPE**, **MAE**, and **RMSE** values, indicating the model's ability to capture their trends effectively. In contrast, the **IR** sensors (IR1-IR5) displayed a wider range of error metrics, with **MAPE** values reaching up to 271.99% for IR5 and **RMSE** values as high as 182.0 for IR4.

RNN

The **RNN** model was initially applied to Dataset 3 using the best hyperparameters identified in Dataset 2. The training window covered the period from September 18 at 10h00 to September 21 at 10h30, with test data starting at 10h45. The runs comprise an increment of 15 min until 12h00. While this configuration outputs satisfactory results for Temperature and Humidity, the performance for the **IR** sensors (IR1-IR5) was suboptimal, failing to detect anomalies effectively during the **WSE**. To overcome this, two hyperparameters were adjusted for the **IR** sensors: `hidden_dim` was reduced from 205 to 160, and `n_epochs` was increased from 55 to 100.

These adjustments aimed to improve the model's capacity to detect anomalies in the **IR** sensor data while maintaining its effectiveness for Temperature and Humidity. Table 5.20 summarizes the performance metrics of the **RNN** model after applying the adjusted hyperparameters. The results reveal a significant improvement in the model's ability to handle the highly variable and noisy patterns characteristic of the **IR** sensors while providing robust predictions for Temperature and Humidity.

Table 5.20: Range Minimum and Maximum of RNN Models Metrics for WSE in Dataset 3.

Variable	MAPE [%]	MAE	MSE	RMSE
Temperature	[1.94, 15.38]	[0.35, 2.76]	[0.17, 7.88]	[0.41, 2.81]
Humidity	[1.71, 7.36]	[1.15, 4.48]	[1.86, 20.42]	[1.37, 4.52]
IR1	[27.23, 180.74]	[13.99, 703.17]	[221.05, 7082.04]	[14.87, 841.6]
IR2	[5.11, 172.1]	[2.19, 705.81]	[13.89, 7084.51]	[3.73, 841.67]
IR3	[17.22, 171.75]	[7.61, 704.57]	[98.75, 7100.02]	[9.94, 842.65]
IR4	[8.1, 61.03]	[40.32, 747.76]	[261.0, 8238.88]	[51.14, 907.65]
IR5	[20.9, 164.2]	[10.2, 705.59]	[197.14, 7134.12]	[14.04, 844.63]

The **RNN** model, after hyperparameter adjustments, exhibited relatively low **MAPE** and **RMSE** values for Temperature and Humidity, which indicates the substantial predictive accuracy and stability during the **WSE** period. However, the **IR** sensors (IR1-IR5) showed significantly higher error ranges, which could be due to the intensity levels of **IR** during this period of the year. Despite these elevated errors, the adjustments enabled the detection of anomalies under highly dynamic conditions. The metrics emphasize the continued challenges of modeling the complex and spiky data patterns of **IR** sensors. However, the model's refined configuration enhanced sensitivity to fire-related anomalies in this scenario.

TimesFM

The **TimesFM** model was initially applied to Dataset 3 using the hyperparameter values identified for Dataset 2. The training window extended from September 20 at 10h00 to September 21 at 10h30, with testing window starting at 10h45. This was repeated with an increment of 15 min on both until 12h00. While this configuration achieved significant results, it became evident that further refinements could enhance the model's performance across all variables, particularly under the dynamic conditions of the **WSE**. Consequently, an optimization process was conducted to identify the best-performing values for the `context_len` parameter, which is critical for capturing temporal dependencies in the dataset.

The refined `context_len` values for each variable were determined as follows: 32 for Temperature, 352 for Humidity, and 512, 512, 512, 32, and 288 for IR1, IR2, IR3, IR4, and IR5, respectively. These adjustments allowed the **TimesFM** model to better adapt to the unique characteristics of Dataset 3 and improve its sensitivity to detecting fire-related anomalies during the **WSE** interval.

Table 5.21: Range Minimum and Maximum of TimesFM Models Metrics for WSE in Dataset 3.

Variable	MAPE [%]	MAE	MSE	RMSE
Temperature	[0.03, 0.14]	[0.51, 2.51]	[0.48, 6.64]	[0.69, 2.58]
Humidity	[0.01, 0.12]	[0.38, 7.01]	[0.32, 49.33]	[0.57, 7.02]
IR1	[0.3, 3.71]	[12.28, 175.08]	[191.18, 554.51]	[13.83, 235.39]
IR2	[0.04, 0.12]	[1.89, 5.32]	[3.88, 36.27]	[1.97, 6.02]
IR3	[0.14, 1.92]	[5.62, 78.9]	[49.38, 71.99]	[7.03, 84.37]
IR4	[0.28, 1.6]	[76.89, 176.68]	[109.57, 392.98]	[104.42, 198.16]
IR5	[0.52, 3.14]	[21.55, 145.83]	[57.99, 214.09]	[23.94, 146.53]

The effectiveness of the refined **TimesFM** configuration for Dataset 3 output Temperature and Humidity metrics exceptionally low **MAPE** values, ranging from 0.03% to 0.14% for Temperature and 0.01% to 0.12% for Humidity, indicating high accuracy in capturing their patterns during the **WSE**.

For the **IR** sensors, the metrics reflect a significant range of values, with **MAPE** values ranging from 0.3% to 3.71%. Despite the broader ranges for **MAE**, **MSE**, and **RMSE**, these results underscore the model's ability to adapt to the spiky and dynamic nature of the **IR** data.

Combining ML models

Based on the anomaly counts identified for each model and variable, the combination strategy outlined in Section 5.7.2 was applied to Dataset 3. This strategy aggregates the outputs of the four models (**RWR**, **ARIMA**, **RNN**, and **TimesFM**) through two approaches: unweighted and weighted voting. The unweighted approach considers each model equally, summing their anomaly outputs to detect periods where multiple models identify anomalies simultaneously. On the other hand, the weighted approach assigns higher importance to Temperature and Humidity anomalies – due to their stronger correlation with fire events – while reducing the influence of **IR** sensor outputs, which tend to produce more false positives. Figure 5.58 illustrates the aggregated anomaly scores for both strategies.

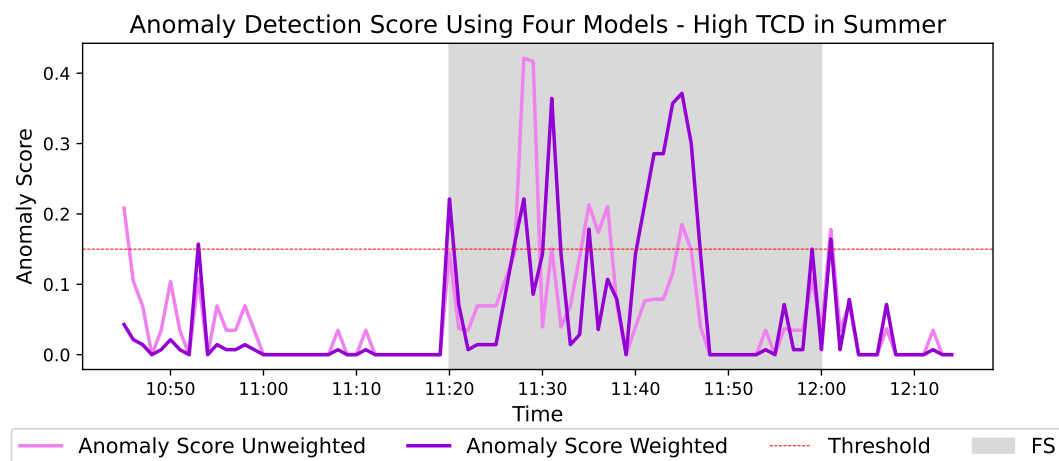


Figure 5.58: Aggregated Anomaly Scores from Dataset 3 (High TCD Scenario in Summer) Using Unweighted and Weighted Strategies.

The plot illustrates the combined anomaly detection scores for Dataset 3, generated through unweighted and weighted aggregation strategies. A threshold of 0.15 was applied to identify significant anomalies during the **WSE** period.

Analysing the results, the weighted approach identified 11 anomalies during the **WSE**, compared to 7 for the unweighted method. This highlights the weighted strategy's greater sensitivity in capturing fire-related anomalies while maintaining a similar count of anomalies 1 before and 1 after the **WSE** period. Conversely, the unweighted approach also flagged 1 anomaly before and after the **WSE**.

The anomaly detected before the **WSE** in both strategies can be attributed to minor variations in environmental conditions recorded by the modules, which momentarily exceeded the defined threshold. This highlights the models' sensitivity to abrupt but non-fire-related changes, such as fluctuations in temperature or **IR**s sensor noise. Despite this, the results indicate a strong alignment between detected anomalies and the fire event during the **WSE**, particularly for the weighted approach, effectively reducing false positives while maintaining sensitivity to critical events.

During the **WSE** period, both the weighted and unweighted aggregation strategies identified anomalies that exceeded the threshold of 0.15, as detailed below:

- **Weighted Approach:** at 11 h20 - Score: 0.221, at 11 h27 - Score: 0.157, and at 11 h28 - Score: 0.221.
- **Unweighted Approach:** at 11 h28 - Score: 0.421, at 11 h29 - Score: 0.417, and at 11 h31 - Score: 0.150.

The weighted approach has a faster response, identifying three anomalies within an interval of 8 min during the fire event. This more rapid detection makes it a robust solution for identifying fires in high-density forest environments. By prioritizing key variables such as Temperature and Humidity and reducing the influence of **IR** sensors, the weighted strategy ensures both timely and accurate anomaly detection, which is crucial for early fire identification and mitigation.

5.8.2 Applying the approach in Dataset 4

Dataset 4 was collected during a controlled burn procedure in Montesinho Natural Park. This data collection was conducted under the supervision of local firefighting authorities as part of a forest management strategy to mitigate wildfire risks. The dataset provides an invaluable opportunity to evaluate the performance of the anomaly detection techniques in a realistic scenario closely resembling actual wildfire events.

Dataset 4 comprises a period of two days, during which a **WSN** composed of four modules was deployed in strategic locations surrounding the burn site. The region where the **WSN** was installed has low **TCD**, making it suitable for testing anomaly detection techniques under conditions resembling sparse vegetation. Each module transmitted data using the **DROWSIER** filter, ensuring a minimum transmission frequency of once every 15 min under stable conditions, and increasing to every 1 min in the presence of abrupt changes in the data.

The temporal resolution of Dataset 4 was regularized by resampling to 60s intervals using a forward-filling method. This approach ensured that intervals between transmissions were populated with the most recent recorded values, avoiding gaps in the time series. Additionally, no overlap or omission of data occurred, preserving the integrity of the dataset for subsequent analysis.

RWR

Given the characteristics of the region, the application of **ML** techniques followed the methodology defined for low **TCD** scenarios, as detailed in Section 5.6. This approach involves using models optimized for sparse vegetation, considering the specific dynamics and environmental conditions of such areas. Therefore, the **RWR** model was applied to Dataset 4 using a rolling window approach, with a 24h training window (from January 26 at 11 h30 to January 27 at 11 h45), a forecast horizon of 15 min, and a test window incremented every 15 min. This iterative process continued until January 27 at 16h45.

Initially, the lag configuration mirrored the values identified as optimal for Dataset 1. However, adjustments were necessary to better align with the unique characteristics of Dataset 4. The revised lag configuration included 1 for Temperature, Humidity, and IR1-IR4, while IR5 was 4. Table 5.22 summarizes the performance metrics of the **RWR** model.

Table 5.22: Range Minimum and Maximum of RWR Models Metrics for WSE in Dataset 4.

Variable	MAE	MedAE	Explained Variance	MSE	RMSE
Temperature	[0.0473, 7.4579]	[0.0144, 4.6125]	[-1.0552, 0.7151]	[0.002447, 116.6521]	[0.04947, 10.8006]
Humidity	[0.7575, 12.0684]	[0.5137, 10.883]	[-0.8809, 0.6944]	[0.9823, 268.2]	[0.9911, 16.3768]
IR1	[15.0599, 457.7391]	[17.0337, 623.8951]	[-159.3766, 0.6097]	[251.47, 3472.7642]	[15.8578, 589.2459]
IR2	[12.2488, 605.9366]	[13.0078, 745.7555]	[-612.0647, 0.2565]	[169.7671, 4302.2447]	[13.0295, 655.9369]
IR3	[10.8206, 644.8963]	[9.8927, 959.1139]	[-358.7507, 0.4114]	[128.1885, 6164.4534]	[11.322, 785.134]
IR4	[14.7848, 710.9267]	[14.5668, 932.1934]	[-2526.5374, 0.3601]	[246.5324, 6336.1396]	[15.7014, 796.0]
IR5	[2.9796, 641.8031]	[2.4774, 951.6861]	[-6511.6922, 0.5697]	[12.3889, 6083.506]	[3.5198, 779.9926]

Table 5.22 presents the range of performance metrics for the **RWR** model applied to Dataset 4. The results highlight a significant variation in the model's ability to capture anomalies across

different variables. Temperature and Humidity achieved relatively lower error ranges, with **MAE** and **RMSE** values indicating the model's capacity to accurately forecast these variables under low **TCD** conditions. However, the **IR** sensors (IR1-IR5) exhibited substantially higher error metrics, particularly for IR3 and IR4, where **RMSE** values exceeded 700, reflecting the increased difficulty in predicting abrupt changes in **IR** readings during the **WSE**.

ARIMA

The AutoARIMA approach was tested with a 24h training window, followed by a 15 min forecast and testing. In this sense, the first train window started on January 26 at 11h30 and ended on January 27 at 11h45. This configuration is similar to the best parameters found for Dataset 1. Table 5.23 presents the performance metrics for the **ARIMA** model applied to Dataset 4.

Table 5.23: Range Minimum and Maximum of ARIMA Models Metrics for WSE in Dataset 4.

Variable	MAPE [%]	MAE	MSE	RMSE
Temperature	[0.45, 24.71]	[0.08, 8.34]	[0.01, 128.96]	[0.09, 11.36]
Humidity	[2.48, 45.62]	[1.35, 12.66]	[2.88, 310.59]	[1.7, 17.62]
IR1	[2.68, 1599.87]	[27.36, 562.44]	[896.72, 39175.43]	[29.95, 625.9]
IR2	[2.67, 6463.68]	[26.93, 622.42]	[892.74, 41913.3]	[29.88, 647.44]
IR3	[1.66, 1746.61]	[16.23, 643.22]	[304.27, 59253.96]	[17.44, 769.78]
IR4	[2.24, 1381.6]	[22.81, 703.63]	[643.77, 61974.57]	[25.37, 787.26]
IR5	[1.29, 1764.16]	[12.71, 644.86]	[223.95, 60468.09]	[14.96, 777.62]

The results show relatively low error metrics for Temperature and Humidity, with **MAPE** values starting at 0.45% and 2.48%, respectively. This highlights the model's ability to capture general trends for these variables under low **TCD** conditions. However, the **IR** sensors (IR1-IR5) indicate significantly higher error ranges, indicating challenges in modeling their noisy and dynamic data patterns. The high **RMSE** values, particularly for IR2 through IR5, further underscore the difficulty of accurately forecasting **IR** sensor readings during the **WSE** period.

RNN

The **RNN** model was applied to Dataset 4 using the same configurations identified as optimal for Dataset 1. The training process employed a rolling window of 24h, a 15 min forecast horizon, and incremental testing updates every 15 min. The training began on January 26 at 11h30 and concluded on January 27 at 11h45, with the test data extending until January 27 at 16h45. Table 5.24 summarizes the performance metrics obtained for the RNN model, showcasing its ability to detect anomalies across variables in a low **TCD** scenario.

Table 5.24: Range Minimum and Maximum of RNN Models Metrics for WSE in Dataset 4.

Variable	MAPE [%]	MAE	MSE	RMSE
Temperature	[0.52, 34.39]	[0.1, 10.9]	[0.01, 180.82]	[0.11, 13.45]
Humidity	[1.02, 36.42]	[0.56, 13.73]	[0.32, 257.42]	[0.56, 16.04]
IR1	[12.66, 3097.89]	[117.89, 743.82]	[2632.26, 56387.54]	[162.25, 749.93]
IR2	[17.80, 5156.31]	[176.19, 731.34]	[3340.82, 54441.26]	[182.78, 737.66]
IR3	[17.59, 2818.71]	[109.36, 739.98]	[3237.12, 55659.33]	[179.94, 746.10]
IR4	[17.38, 2279.23]	[177.77, 728.79]	[3163.73, 54357.71]	[177.77, 735.09]
IR5	[17.63, 2924.51]	[114.54, 740.84]	[2331.59, 57907.59]	[154.70, 746.93]

The performance metrics summarized in Table 5.24 reveal the variability in the **RNN** model's ability to detect anomalies in Dataset 4. For Temperature and Humidity, the **RNN** model exhibited relatively low error ranges, with MAPE values from 0.52% to 34.39% for Temperature and from 1.02% to 36.42% for Humidity. These results indicate that the model effectively captured anomalies related to these smoother variables, even under the dynamic conditions of the **WSE**. In contrast, the **IR** sensors (IR1-IR5) showed significantly larger error ranges. The high sensitivity of the **IR** sensors to abrupt environmental changes contributed to these elevated error metrics, highlighting the challenges of accurately predicting their behavior during sudden fire-related events.

TimesFM

The **TimesFM** model was initially applied to Dataset 4 using the best-performing `context_len` values identified for Dataset 1. These configurations, specifically optimized for the previous dataset, included 448 for Temperature, 224 for Humidity, and 512, 352, 480, 320, and 448 for IR1-IR5, respectively. While these parameters output significant results, it was evident that further adjustments could enhance the model's performance on Dataset 4.

After an optimization process adapted to the characteristics of Dataset 4, revised `context_len` values were identified, resulting in improved anomaly detection metrics. The optimal values for this dataset were 224 for Temperature, 160 for Humidity, and 352, 352, 352, 160, and 224 for IR1-IR5, respectively. These adjustments aimed to capture better the temporal dependencies and dynamic environmental conditions specific to the controlled burn scenario in Montesinho Natural Park. The performance metrics of the **TimesFM** model with the adjusted configurations are summarized in Table 5.25, highlighting its capacity to adapt to the unique demands of Dataset 4.

Table 5.25: Range Minimum and Maximum of TimesFM Models Metrics for WSE in Dataset 4.

Variable	MAPE [%]	MAE	MSE	RMSE
Temperature	[0.0, 0.31]	[0.05, 9.97]	[0.0, 158.62]	[0.06, 12.59]
Humidity	[0.0, 0.46]	[0.2, 16.35]	[0.04, 291.65]	[0.21, 17.08]
IR1	[0.0, 24.15]	[0.58, 591.68]	[0.47, 368.63]	[0.69, 60.12]
IR2	[0.0, 23.19]	[1.11, 780.77]	[1.52, 643.32]	[1.23, 80.92]
IR3	[0.01, 35.15]	[5.65, 924.32]	[37.36, 873.07]	[6.11, 93.44]
IR4	[0.0, 12.08]	[1.19, 617.93]	[2.63, 477.35]	[1.62, 69.05]
IR5	[0.0, 35.79]	[3.07, 909.38]	[15.15, 849.74]	[3.89, 92.69]

The performance metrics indicate the effectiveness of the adjusted `context_len` values for the **TimesFM** model in Dataset 4. The **MAPE** values remain low for Temperature and Humidity, indicating the model's substantial predictive accuracy for these variables, even under the dynamic conditions of the controlled burn. For the **IR** sensors, the metrics exhibit a wider range, reflecting these variables' inherent complexity and spikiness during the **WSE**. Notably, IR2 shows unusually large values, suggesting potential irregularities in the data or challenges in model adaptation. However, IR1 and IR3-IR5 present relatively consistent performance, with reasonable **MAE**, **MSE**, and **RMSE** values, despite the variability in the data.

Combining ML models

The process of combining multiple ML models was applied to Dataset 4 to enhance the precision and reliability of anomaly detection in a low TCD scenario (which was based on the method described in Section 5.7.1). The aggregated anomaly detection scores reflect the integration of the outputs from RWR, ARIMA, RNN, and TimesFM, utilizing both unweighted and weighted aggregation strategies.

The weighted strategy was designed to give more importance to Temperature and Humidity variables, which have consistent and reliable behavior in detecting fire-related anomalies. In contrast, the unweighted approach treats all variables equally, including the highly sensitive IR sensors, which are prone to false positives. By comparing these strategies, the objective is to identify the most effective method for accurately isolating anomalies linked to the controlled burn in Montesinho Nature Park. Figure 5.59 captures the evolution of these scores over time.

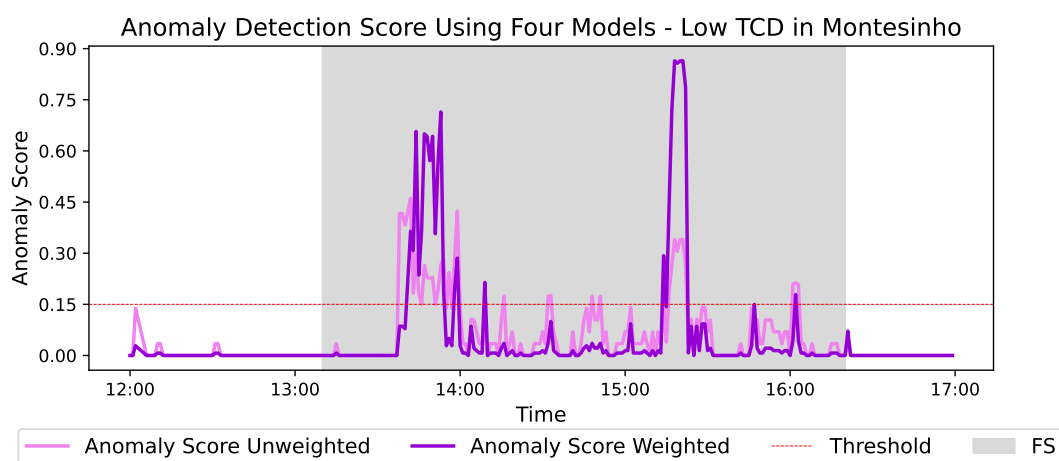


Figure 5.59: Aggregated Anomaly Scores from Dataset 4 (Low TCD Scenario in Montesinho) Using Unweighted and Weighted Strategies.

Using a threshold of 0.15, the results reveal distinct behaviors for the weighted and unweighted aggregation strategies. The weighted approach identified a total of 26 anomalies during the WSE period, while no anomalies were flagged before or after the fire event. This focused detection highlights the effectiveness of the weighted strategy in capturing anomalies directly linked to the fire, minimizing false positives outside the WSE interval.

In contrast, the unweighted strategy detected 35 anomalies during the WSE period, but similarly to the weighted approach, no anomalies were recorded before or after the fire event. While the unweighted method has higher sensitivity, it may also reflect greater susceptibility to noise, particularly from IR sensors, which are more prone to variability.

The anomaly detection timing during the WSE provides further responsiveness of the weighted and unweighted aggregation strategies. For the weighted approach, the first three anomalies exceeding the threshold of 0.15 were recorded at 13h41, 13h42, and 13h43, with anomaly scores

of 0.22, 0.36, and 0.30, respectively. This sequence indicates a focused and consistent identification of anomalies within a short timeframe.

In comparison, the unweighted approach identified its first three anomalies slightly earlier, at 11h38, 11h39, and 11h40, with corresponding scores of 0.41, 0.41, and 0.38. While the unweighted strategy detected anomalies quicker, its earlier responses may also reflect greater sensitivity to noise or abrupt changes unrelated to the fire event.

It is important to note that the fire started approximately 76m away from the first module deployment site (module 12-5FTH), as measured via satellite mapping tools. Given this distance, the detection times indicate how effectively each strategy captured the progression of the fire event. The unweighted approach flagged its first anomaly 28 min after the fire's initiation, whereas the weighted strategy identified its first anomaly 31 min after the fire began. This highlights the unweighted method's faster response but raises concerns about its potential sensitivity to environmental variability or noise.

5.8.3 Applying the approach in Dataset 5

Dataset 5 was collected during a controlled fire propagation procedure in Rebordãoinhos. This operation was part of a biomass volume management strategy the local firefighting authorities conducted. The test provided a valuable opportunity to evaluate the performance of the anomaly detection techniques in a high **TCD** environment. The primary objective was to assess whether the combination of anomalies from the four models, as outlined in Section 5.6, could effectively identify the presence of fire under real-world conditions.

The dataset extends from February 21 at 15h to February 23 at 23h59, with the **WSE** occurring on February 22 between 14h05 and 15h55. A **WSN** consisting of five modules strategically deployed at varying distances from the fire source, ranging from 0m to 47m. This configuration allowed for the collection of detailed data representing the fire's progression and impact across different proximities.

All modules were implemented with the **DROWSIER** filter to ensure reliable data transmission. This guaranteed a minimum payload every 15min under stable conditions and increased the frequency to every 1 min during abrupt changes in sensor readings. To address the difference in the time intervals, a forward-filling methodology was applied to standardize the dataset to 1 min intervals. In cases where no new payload was received, the last recorded value was carried forward to populate the subsequent minute. Importantly, no overlapping transmissions occurred, ensuring that data integrity was preserved throughout the collection period.

Given the high **TCD** of the Rebordãoinhos region, this dataset presents an excellent test case for applying **ML** techniques optimized for dense forest environments. Therefore, the following describes the application of **RWR**, **ARIMA**, **RNN**, and **TimesFM** models to Dataset 5 and analyses the results obtained.

RWR

The **RWR** model was initially applied to Dataset 5 using the best configuration found for Dataset 2. While some variables have satisfactory results, the performance for certain variables remained suboptimal. To address these discrepancies, an adjustment of the lag hyperparameters for each variable was conducted to enhance the model's accuracy.

The rolling training window was 24h, beginning on February 21 at 12h30 and ending on February 22 at 12h45. The first test window started on February 22 at 13h, with subsequent runs incrementing the training and testing windows by 15 min until February 22 at 16h. Based on the optimization process, the best-performing lags were determined to be 18 for Temperature, 12 for Humidity, and 7, 8, 8, 7, and 19 for IR1-IR5, respectively. Table 5.26 provides a detailed summary of the range of metrics obtained using these refined lag values.

Table 5.26: Range Minimum and Maximum of RWR Models Metrics for WSE in Dataset 5.

Variable	MAE	MedAE	Explained Variance	MSE	RMSE
Temperature	[0.7666, 9.9798]	[0.4243, 8.9745]	[-4.6242, 0.8428]	[0.7354, 134.4552]	[0.8576, 11.5955]
Humidity	[1.2363, 6.1339]	[0.5857, 6.9932]	[-7.7374, 0.5521]	[2.3629, 53.8095]	[1.5372, 7.3355]
IR1	[62.8253, 669.8956]	[54.7724, 849.0224]	[-238.4170, 0.5509]	[4185.2966, 517716.1256]	[64.6939, 719.5249]
IR2	[47.1601, 403.5556]	[46.1993, 619.0052]	[-535.3770, 0.5739]	[2323.5530, 274741.7609]	[48.2032, 524.1581]
IR3	[28.2422, 670.2557]	[26.5708, 800.0121]	[-246.2989, 0.5185]	[871.4224, 517776.1127]	[29.5199, 719.5666]
IR4	[67.8815, 662.9798]	[63.0799, 844.5038]	[-1477.7156, 0.4359]	[4827.7426, 483866.2323]	[69.4820, 695.6049]
IR5	[7.7606, 613.2524]	[6.8154, 703.5579]	[-359.7054, 0.5838]	[89.1632, 474819.0535]	[9.4426, 689.0712]

The refined lag configuration significantly improved the model's predictive performance for certain variables. Temperature variable achieved an **MAE** range between 0.76 and 9.98, indicating relatively accurate predictions even during the **WSE**, while Humidity exhibited less variability with **MAE** values ranging from 1.23 to 6.13. In contrast, the **IR** sensors showed a broader range of errors. IR1 and IR2, for example, displayed high **RMSE** values. These results suggest that the **IR** sensors are more susceptible to noise and rapid fluctuations.

ARIMA

The **ARIMA** model was applied to Dataset 5 with a training window of 24h, inspired by the configuration utilized for Dataset 2, where the best results were obtained with a 48h window. Therefore, the training process began on February 21 at 12h30 and ended on February 22 at 12h45. The first testing occurred on February 22 at 13h, with subsequent training and testing windows incrementing every 15 min. The process continued iteratively until February 22 at 16h00. This setup aligned with the configuration adopted for the **RWR** model, providing consistency in the analysis framework.

Table 5.27 presents the range of minimum and maximum performance metrics achieved by the **ARIMA** model on Dataset 5. These metrics illustrate the variability in predictive accuracy across different variables, reflecting the challenges and effectiveness of the model under the specific conditions of the dataset.

Table 5.27: Range Minimum and Maximum of ARIMA Models Metrics for WSE in Dataset 5.

Variable	MAPE [%]	MAE	MSE	RMSE
Temperature	[2.95, 28.70]	[0.65, 10.07]	[0.44, 145.77]	[0.66, 12.07]
Humidity	[5.08, 36.02]	[1.15, 5.49]	[2.33, 39.30]	[1.53, 6.27]
IR1	[51.36, 1757.08]	[18.31, 618.50]	[722.42, 4535.14]	[26.88, 67.34]
IR2	[51.04, 1285.69]	[12.37, 422.28]	[307.61, 3219.60]	[17.54, 56.74]
IR3	[29.57, 2341.05]	[6.57, 567.27]	[46.46, 3328.27]	[6.82, 57.69]
IR4	[104.73, 2545.84]	[36.16, 775.54]	[1352.83, 6291.79]	[36.78, 79.32]
IR5	[26.52, 2271.15]	[6.16, 557.24]	[46.65, 3176.54]	[6.83, 56.35]

The performance metrics for the **ARIMA** model on Dataset 5 show considerable variability across different variables, reflecting the model's challenges and effectiveness under the specific conditions of this dataset. For Temperature and Humidity, the metrics indicate relatively lower errors, with **MAPE** values ranging from 2.95% to 28.70% for Temperature and 5.08% to 36.02% for Humidity. These results suggest that **ARIMA** effectively captures the temporal patterns in these variables, particularly during stable periods.

The **IR** sensor variables (IR1-IR5) exhibit higher error ranges, with **MAPE** values. The high variability in **IR** sensor readings, coupled with the abrupt changes introduced by the **WSE**, likely contributes to these elevated error metrics. Despite this, some **IR** sensors, such as IR3 and IR5, show marginally better performance, with lower maximum **MSE** and **RMSE** values compared to IR1 and IR4.

RNN

The **RNN** model was initially applied to Dataset 5 using the best-performing hyperparameters identified for Dataset 2. However, while the configuration showed promising results for the Temperature variable, the performance for Humidity and **IR** sensors (IR1-IR5) was suboptimal. In Dataset 2, the RNN models were trained with a 72h window, but for Dataset 5, adjustments were necessary to optimize anomaly detection for the specific characteristics of these variables.

The refinement process for the **RNN** model followed the same methodology as before, with a 24h training window applied from February 21 at 12h30 to February 22 at 12h45, and a test window starting at 13h00. The forecast horizon and incremental training updates were set to 15 min. This iterative process continued until February 22 at 16h00. The optimal hyperparameters identified for Temperature, Humidity, and **IR** sensors are presented in Table 5.28.

Table 5.28: Best RNN Hyperparameters for Variables in Dataset 5.

Variable	Input Chunk Length	Hidden Dim	RNN Layers	Epochs	Dropout
Temperature	80	80	2	80	0.3
Humidity	45	50	2	50	0.3
IR1-IR5	30	250	2	70	0.1

The range of metrics achieved using these refined hyperparameters is presented in Table 5.29.

Table 5.29: Range Minimum and Maximum of RNN Models Metrics for WSE in Dataset 5.

Variable	MAPE [%]	MAE	MSE	RMSE
Temperature	[3.44, 29.71]	[0.72, 10.28]	[1.04, 146.16]	[1.02, 12.09]
Humidity	[5.57, 53.61]	[1.25, 7.65]	[2.65, 70.43]	[1.63, 8.39]
IR1	[498.62, 1664.94]	[213.08, 599.72]	[115123.00, 393304.25]	[33.90, 627.14]
IR2	[809.84, 3323.90]	[219.70, 626.34]	[116668.15, 394490.15]	[34.17, 628.08]
IR3	[724.11, 2514.01]	[218.70, 603.64]	[120046.65, 375028.38]	[34.68, 612.40]
IR4	[580.87, 1915.13]	[216.53, 608.43]	[119204.28, 394635.39]	[34.56, 628.20]
IR5	[718.35, 2451.62]	[216.53, 604.96]	[118580.40, 375017.10]	[34.46, 612.39]

The performance metrics for the **RNN** model on Dataset 5 reveal a contrast between variables. Temperature and Humidity exhibit relatively low error ranges, with **MAPE** values extending 3.44% to 29.71% for Temperature and 5.57% to 53.61% for Humidity. This indicates that the model effectively captures trends in these variables despite some variability. The **IR** sensors (IR1-IR5), however, display significantly higher error metrics. Underscoring the challenges in modeling the rapid fluctuations and high variability characteristic of **IR** data. IR2 shows the highest **MAPE** and **MSE** values, suggesting a heightened sensitivity to abrupt changes in fire-related conditions.

TimesFM

The **TimesFM** model was initially tested on Dataset 5 using the `context_len` identified as optimal for Dataset 2. However, these values did not perform satisfactorily for all variables in this dataset. Consequently, a refinement process was conducted to identify the best-performing `context_len` adapted to the specific characteristics of Dataset 5.

The fine-tuning followed the same methodology applied to the other models, with a rolling training window of 24h, with a forecast horizon and test window of 15 min. The training began on February 21 at 12h30 and ended on February 22 at 12h45, with the first test data occurring on February 22 at 13h00. Incremental updates to the training-testing windows occurred every 15 min until February 22 at 16h00. Through this iterative process, the following `context_lens` were determined as optimal for each variable: 448 for Temperature, 160 for Humidity, and 96, 64, 96, 96, and 32 for IR1-IR5, respectively. Table 5.30 presents the range of metrics achieved using these optimized `context_len`.

Table 5.30: Range Minimum and Maximum of TimesFM Models Metrics for WSE in Dataset 5.

Variable	MAPE [%]	MAE	MSE	RMSE
Temperature	[0.04, 0.30]	[0.92, 10.35]	[0.91, 152.30]	[0.96, 12.34]
Humidity	[0.02, 0.45]	[0.52, 6.39]	[0.64, 51.94]	[0.80, 7.21]
IR1	[0.10, 14.53]	[14.11, 588.70]	[332.23, 469299.52]	[18.23, 685.05]
IR2	[0.11, 17.51]	[2.69, 410.22]	[8.78, 315140.34]	[2.96, 561.37]
IR3	[0.17, 15.39]	[7.89, 530.41]	[96.47, 389417.37]	[9.82, 624.03]
IR4	[0.14, 15.98]	[11.37, 588.98]	[246.80, 485132.86]	[15.71, 696.51]
IR5	[0.13, 25.52]	[3.17, 608.16]	[11.66, 424879.87]	[3.42, 651.83]

The performance metrics for the **TimesFM** model on Dataset 5 highlight its strong predictive capabilities for Temperature and Humidity, with very low **MAPE** values ranging from 0.04% to 0.30% for Temperature and 0.02% to 0.45% for Humidity. These results indicate the model's accuracy in capturing stable trends for these variables, even in challenging high **TCD** environments.

The **TimesFM** model exhibits a wider range of errors for the **IR** sensors (IR1-IR5). IR1 and IR2 display **MAPE** values up to 17.51%, while IR4 and IR5 show peaks of 15.98% and 25.52%, respectively. These elevated error metrics suggest that, while **TimesFM** can effectively detect anomalies, the spiky and abrupt variations in **IR** sensor data present a challenge.

Combining ML models

The combination of **RWR**, **ARIMA**, **RNN**, and **TimesFM** was applied to Dataset 5 to assess the aggregated anomaly detection performance during the controlled fire event in Rebordaínhos. This approach considers each model's strengths while mitigating their weaknesses, using both unweighted and weighted aggregation strategies, as detailed in Section 5.7.2.

The combined anomaly scores were calculated by aggregating the outputs of the four models for each variable at each timestamp. In the unweighted strategy, each model contributed equally to the aggregated score, while the weighted approach prioritized Temperature and Humidity anomalies by assigning them higher weights, reducing the influence of **IR** sensor outputs. Figure 5.60 illustrates the anomaly scores derived from both strategies, with a threshold of 0.15 applied to identify significant anomalies. This threshold was selected based on prior analyses to maintain consistency across datasets.

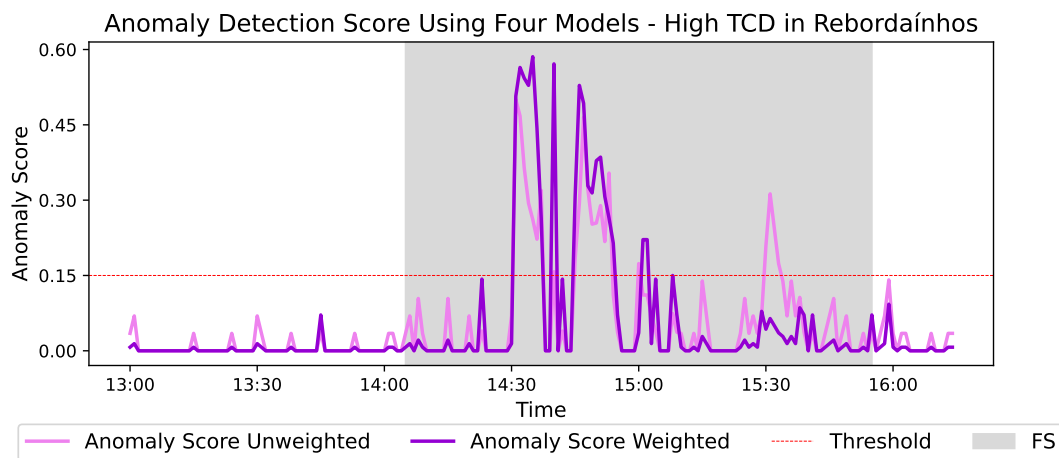


Figure 5.60: Aggregated Anomaly Scores from Dataset 5 (Low TCD Scenario in Rebordaínhos) Using Unweighted and Weighted Strategies.

Using a threshold of 0.15, the anomaly scores from both weighted and unweighted aggregation strategies were analysed to evaluate their performance during the **WSE** in Dataset 5. The weighted strategy identified a total of 20 anomalies during the **WSE** period, with no anomalies detected before or after. Similarly, the unweighted strategy flagged 22 anomalies during the same period, also without anomalies outside the **WSE** interval. These results indicate that both approaches effectively constrained their anomaly detection to the fire event, reflecting strong alignment with the actual environmental changes.

The first anomalies detected during the **WSE** for both strategies occurred at similar times. For the weighted strategy, the anomalies were identified at 14h31 with a score of 0.507, at 14h32 with a score of 0.564, and at 14h33 with a score of 0.543. For the unweighted strategy, the anomalies were flagged at 14h31 with a score of 0.500, at 14h32 with a score of 0.468, and at 14h33 with a score of 0.364.

Both strategies successfully detected anomalies at the same initial timestamp (14h31), suggesting their sensitivity to the fire event's early stages. The first detection occurred 26 min after the fire was ignited at (14h05). It is important to note that the first module in the fire propagation line was positioned approximately 121 m in a straight line from the ignition point. This distance accounts for the delay in detection, as the flames had to propagate and create sufficient environmental changes to trigger the anomaly detection. Despite the delay, both strategies provided timely alerts once the fire's impact reached the sensor modules.

These results highlight the robustness of the weighted strategy for detecting fire-related anomalies in high **TCD** scenarios. By reducing the influence of variables prone to noise, such as **IR** sensors, this approach ensures precise anomaly identification while maintaining sensitivity to the dynamics of the fire event.

5.9 Final considerations

This chapter has provided a comprehensive evaluation of **ML** techniques applied to **WSN** data for anomaly detection in forest fire scenarios. In the first step, a literature review was conducted to explore related works in the field, highlighting the strengths and limitations of previous studies. It was observed that while many studies focus on applying **ML** models to environmental data, few have explored the use of low-cost modules combined with **ML** for detecting fire events. Furthermore, several works lacked sufficient detail regarding the testing scenarios, including the specific conditions under which the modules were deployed, as well as the fire events. The main contribution of this chapter is the extensive testing, which includes fire detection during summer months – critical for capturing the effects of high temperatures and increased fire risk – and fire propagation experiments across two different **TCD** scenarios. These scenarios provide a more complete validation of the methods, ensuring their robustness in diverse real-world conditions.

The approach was first applied to two distinct datasets representing low and high **TCD** scenarios, then further validation on three additional datasets replicating real wildfire events. This ensured that the models' effectiveness was assessed in controlled conditions and more realistic settings, where environmental factors such as wind, temperature, and humidity fluctuate more unpredictably. In this sense, it assesses the effectiveness of various **ML** models, including **RWR**, **ARIMA**, **RNN**, and **TimesFM**, in identifying and predicting anomalies, particularly during controlled fire events. Each of the methods presented strengths and limitations, with **RWR** being highly effective for capturing linear trends, **ARIMA** excelling in stationary time series forecasting, **RNN** handling temporal dependencies with robustness, and **TimesFM** providing a versatile, pre-trained transformer-based solution capable of adapting well to different datasets.

The chapter then explored the combination of these models to enhance anomaly detection, where the unweighted and weighted aggregation strategies were tested. The results revealed that combining the models improved detection accuracy, particularly in reducing false positives and increasing sensitivity to actual fire events. The weighted aggregation strategy, in particular, proved beneficial by giving more importance to variables such as Temperature and Humidity, which strongly correlated with fire events while reducing the influence of noisy **IR** sensor data.

When analysing the time to detect anomalies, the weighted approach consistently identified fire-related events with greater precision across different datasets. For instance, in the low **TCD** scenario (Dataset 1), the first detection occurred at 9h34, while the unweighted method detected anomalies slightly earlier, at 9h18. In the high **TCD** scenario (Dataset 2), the weighted approach detected the fire at 15h10, while the unweighted approach flagged anomalies at 15h06. Similarly, in another high **TCD** dataset (Dataset 3), the weighted approach identified anomalies at 11h20, whereas the unweighted method detected them at 11h28. For Dataset 4, in a low **TCD** environment and the fire started from 76m, the weighted approach detected anomalies at 13h41, while the unweighted approach identified at 11h38. In Dataset 5, with high **TCD** and the fire ignition beginning from 121m, both strategies identified anomalies at the same time (14h31), 26 min after the fire started, showing the models' sensitivity to early fire stages despite the delay caused by the fire's propagation. The time of anomaly detection for each scenario and strategy is summarized below:

- **Dataset 1 (Low TCD):**
 - **Weighted Approach:** First detection at 9h34 (19 min after fire started).
 - **Unweighted Approach:** First detection at 9h18 (3 min after fire started).
- **Dataset 2 (High TCD):**
 - **Weighted Approach:** First detection at 15h10 (10 min after fire started).
 - **Unweighted Approach:** First detection at 15h06 (6 min after fire started).
- **Dataset 3 (High TCD):**
 - **Weighted Approach:** First detection at 11h20 (the same minute after fire started).
 - **Unweighted Approach:** First detection at 11h28 (8 min after fire started).
- **Dataset 4 (Low TCD):**
 - **Weighted Approach:** First detection at 13h41 (31 min after fire started from 76m).
 - **Unweighted Approach:** First detection at 11h38 (28 min after fire started from 76m).
- **Dataset 5 (High TCD):**
 - **Both Approaches:** First detection at 14h31 (26 min after fire started from 121m).

The comparisons between unweighted and weighted aggregation methods further highlighted the balance between sensitivity and specificity, making the unweighted approach the more robust strategy, especially in high **TCD** environments. The results were further validated in real-world scenarios, including controlled burns in low and high **TCD** environments. In these cases, the models effectively detected fire-related anomalies, with the unweighted aggregation method highlighting superior performance in terms of timeliness. Additionally, the findings from analysing different time windows (pre-fire, during fire, and post-fire) further emphasized the models' adaptability, particularly in capturing abrupt environmental changes during the fire event.

Ultimately, the findings from this chapter underline the potential of using **ML** models for anomaly detection in forest fire monitoring, offering a promising method for enhancing early fire detection capabilities. The combination of multiple models and the application of aggregation strategies holds significant promise for improving the reliability and effectiveness of fire detection systems in both controlled and real-world wildfire scenarios.

Chapter 6

Conclusions and Future Work

The main purpose of this chapter is to summarize the findings presented in this dissertation and explain what it means for wildfire monitoring systems. The chapter begins by discussing the issues that the research aims to solve as well as the problem statement and hypothesis that are explained in Chapter 1 followed by the recapitulation of the work's goals. The chapter finishes with the future research directions to build upon the advancements achieved in this thesis. The chapter is structured as follows:

- Section 6.1 – this section is the first and it gives general information about the whole final thesis. This part of the text is devoted to outlining the main features and their correspondence with the main goals set out in the chapters that opened the thesis.
- Section 6.2 – is concerned with the method by which each research question has been answered, it offers the main findings and validates the hypothesis through real-world deployment and experimental analysis.
- Section 6.3 – explores possible directions for future research and development.

6.1 General Overview

More and more wildfires, particularly in Portugal that are both more frequent and of greater magnitude have managed to show the world the demand for new technologies and strategies that are to be introduced for wildfire monitoring and management. This thesis was motivated by the challenges of wildfire detection and response, particularly due to the low cost, scalability, and efficient operation of systems in remote and resource-constrained areas.

The wildfire monitoring system of the original design includes WSN, cloud computing, and predictive analytics and has contributed greatly in dealing with fires and their signalling facilities. The system is developed with the use of energy efficiency and real-time data acquisition technologies and contains a proactive forecasting capability. It offers a large step forward compared to traditional approaches and state-of-the-art solutions. The development combines the low-cost sensor modules,

reliable communication infrastructure, and advanced cloud-based data processing to design an integrated system that fits well both wildfires.

The chapters in this thesis are subparts, each with a focus on a major factor addressed in the thesis. In Chapter 1, the context and motivation for this work were explained before delineating the most pressing issues with wildfire monitoring and setting the research objectives, hypotheses, and guiding questions. In Chapter 2 there was a full review of the literature of WSN-based monitoring systems, as well as, additional views on the present situation and the gaps spotted. Not only did the chapter propose a high-level architecture for the wildfire monitoring system, but it also gave emphasis to its scalability, energy efficiency, and adaptability to the complexities of forested environments.

Focused on developing and validating sensor modules as the system's core components, Chapter 3 described the multi-sensor capabilities' design and integration. In this sense, the chapter topics were about optimising energy usage through duty-cycling and event-driven transmissions and maintaining low-cost, high-performance characteristics. Following the data flow after its acquisition, Chapter 4 detailed the cloud-based data processing infrastructure, addressing how data flows from sensor nodes to the cloud, where it is processed, analysed, and visualised. This chapter emphasised the use of open-source frameworks, real-time data analytics, and the integration of ML models to enhance the system's predictive capabilities.

Taking advantage of the development of the nodes and the cloud system implemented, Chapter 5 explored the application of ML techniques for anomaly detection based on WSN's prediction data. By using historical data and real-time inputs, the proposed system has shown a promising fire detection capability and also looked very useful for emergencies. The tests that were conducted to replicate the real-world situation were positive, which means that the individual components of the monitoring system were integrated.

In summary, this thesis contributes a comprehensive solution to the complex problem of wildfire monitoring, advancing the field through a combination of innovative technologies and practical implementations. The proposed system bridges critical gaps in current approaches, offering a robust, efficient, and scalable real-time monitoring and decision support framework.

The following section provides a detailed reflection on the research questions and hypotheses outlined in Chapter 1, analysing how the findings from each chapter address these questions and contribute to achieving the goals of this research.

6.2 Main Contributions and Outcomes of Research Questions and Hypotheses

The foundation of the thesis presented in this paper is based on three primary research questions, as outlined in Section 1.2. Each research question addresses a fundamental challenge in wildfire monitoring, serving as the basis for the contributions detailed in this work. The following discusses how these contributions respond to the research questions and validate the associated hypotheses.

Research Question 1

Which types of sensors should be used in each module of the WSN, and will they be capable of identifying fire ignition within the critical six-minute interval?

An extensive review of related works indicated a wide array of sensor technologies applicable to wildfire monitoring. Based on this analysis, seven sensor categories were defined: Temperature Sensors, Optical Sensors, Gas Sensors, Atmospheric Pressure Sensors, Humidity Sensors, Biosensors, and Miscellaneous Sensors. These categories were evaluated for their applicability in detecting fire ignition in forest environments.

The module design for the scalable and rapid ignition detection technology, on the basis that each module would include the temperature, humidity, and IR sensors, has been concluded. The deterministic outcome that was met by the actual choices made is the operative flexibility being held in balance with the diagnostic capacity. In particular, the DHT11 sensor was used to measure both temperature and humidity, whereas the five BPV10NF sensors covering the spectrum of 760nm-1100nm were used for the IR detection. This configuration ensures that the modules are equipped with the most effective sensors for detecting early signs of combustion.

The selection of low-cost microcontrollers and transceivers was made to boost the system's scalability and to guarantee operational efficiency. In the literature, these components are found to be a reliable and affordable solution, which makes it possible to deploy a big number of modules without compromising quality. The ATmega328P microcontroller and the RFM95 transceiver, being compatible with LoRaWAN, were the parts chosen for this purpose. Those are the two key components that ensure about the proper performance characteristics of the nodes by having the capacity to collect and transmit data with high efficiency.

As the module design defined, additional factors critical to their operation in a WSN for wildfire monitoring emerged. One of the most significant challenges identified was battery consumption. Given the remote locations where the modules are deployed and the lack of solar panels – due to the dense forest canopy that would render photovoltaic systems inefficient – it was essential to maximize battery life. To overcome this, a filtering algorithm named DROWSIER was developed.

DROWSIER evaluates whether the data collected deviates significantly from recent values. The filter ensures that data is transmitted at least every 15 min or immediately when abrupt changes are detected in a one-minute interval. By implementing this filter in the firmware, the modules achieved an impressive 324 days of uninterrupted operation in a WSN composed of 20 modules. This result demonstrates the system's ability to balance energy efficiency and monitoring quality, even under real-world conditions.

In addition to energy optimization, communication performance was rigorously tested. To ensure reliable data transmission via LoRaWAN, a dedicated module was tested under favorable signal conditions relative to a LoRaWAN gateway. Over a year, only 0.38% of the data experienced transmission delays exceeding 15.5 min. This outcome highlights the combined effectiveness of the DROWSIER filter and the LoRaWAN communication infrastructure in maintaining data integrity and timeliness.

The results, taken together, can be viewed as the first step of Research Question 1. It refers to the sensors' ability to discover the start of fire effectively. On the other hand, the second part of the question, which is about the modules' responsiveness during the critical six-minute interval, is answered by means of the system implementation explained in Chapters 4 and 5 that touches every aspect of the system. In addition, testing the WSN with a fire under control and supervision provided (replicating real wildfire conditions) was an effective means of determining the system's performance level and thus validating the proposed approach.

Research Question 2

Are the raw data obtained by the WSN's modules suitable for identify a forest ignition?

Research Question 2 is connected to the core functionality of the proposed WSN: Does the raw data collected by the sensor modules reliably indicate forest ignition? This evaluation needs a twofold approach: implementing a cloud-based system to manage the data flow and analysing data collected under controlled conditions.

This thesis first focused on developing a robust cloud infrastructure for data acquisition, storage, and visualization. This system was designed using open-source frameworks, ensuring replicability and adaptability for several contexts. Through seamless integration of these frameworks, the system efficiently processes data transmitted via LoRaWAN using MQTT services from TTN. The cloud platform supports real-time monitoring and allows users to navigate through the entire historical dataset, facilitating in-depth analysis of the WSN dedicated to forest monitoring. This critical component, detailed in Chapter 4, forms the system's backbone, enabling the storage and accessibility of the raw data necessary for subsequent analyses.

With the WSN modules and cloud system defined, controlled and supervised fire tests were conducted in collaboration with firefighting teams. These experiments, referred to throughout this thesis as WSE, resulted in the generation of five datasets across four distinct scenarios. Through visualization of the raw data collected during these tests, initial observations revealed discernible alterations in sensor readings during fire events in two specific scenarios, as demonstrated in Figures 5.3, 5.7, and 5.8. However, these visualizations exposed critical limitations in identifying ignition events directly from raw data. For instance:

- While noticeable, significant deviations in sensor readings often required hours to differentiate from normal variations clearly.
- The analysis of individual modules provided limited information into ignition patterns, raising concerns about scalability and interpretability when a more extensive WSN is deployed.

To address these challenges, EDA techniques were applied to each of the five datasets, aiming to uncover meaningful patterns and characterize the data. Scatter plots, histograms, density distributions, boxplots, correlation matrices, trend analyses, and statistical summaries were utilized to examine the datasets comprehensively. Despite these efforts, the results revealed that raw data alone was insufficient to consistently and reliably identify forest ignition. While abrupt changes

in sensor readings could be linked to fire events in some cases, distinguishing these events from environmental noise or other anomalies remained ambiguous without further processing.

The findings presented in Chapters 4 and 5 collectively answer Research Question 2. While the raw data is as a foundational element for wildfire detection, its direct application for identifying ignition events proved inadequate. Instead, this research highlights the needs of applying analytical techniques and models to process and enhance the raw data. These findings underscore the importance of integrating predictive analytics and anomaly detection algorithms to extract actionable information and ensure the system's efficacy in real-world scenarios.

Research Question 3

Which approach would appropriate the necessities of analysing the multiple data sent by the WSN?

The Research Question 3 addresses the task of parsing the vast and continuous data streams in the distributed WSN. Numerous modules with many environmental parameters collected simultaneously produce the datasets that are the most complex as characterized by multidimensionality, heterogeneous data sources, and temporal dependencies. Besides, the existence of noises is also a challenge for deriving an accurate model. The manual examination of large volume of data is unrealistic, especially when fast response times are a must for wildfire identification. To overcome this problem, this thesis explored ML and data analytics frameworks, which are essentially tools used to forecast and detect anomalies.

These techniques were applied to datasets generated during the WSE experiments, showcasing their ability to identify patterns and predict potential fire ignition events. Time-series forecasting models, including RWR, ARIMA, and RNN, were trained to analyse historical and real-time data streams, capturing temporal dependencies and trends indicative of fire ignition. Additionally, the TimesFM model, pre-trained on external datasets, was validated within the context of this research to assess its applicability. These ML methods were implemented to explore data patterns in each dataset, enabling the identification of changes caused by fire ignition events through the difference between forecasted and actual values.

Anomaly detection using the IForest algorithm was employed to identify deviations from expected sensor behavior (forecasted data from each ML method) compared to actual values, effectively flagging potential ignition points with minimal false positives. A voting majority mechanism was implemented to consolidate decisions based on anomaly scores from the four models. This mechanism included two distinct branches: the first assigned greater weight to the Temperature and Humidity variables while giving less weight to the IR sensors, while the second branch computed voting based on any significant deviations across variables.

In both approaches, a threshold value of 0.15 was used to identify fire ignition. The weighted voting method reduced false positives caused by the IR sensors without compromising detection performance. On the other hand, the unweighted voting method identifies fires in less time. Tests conducted with supervised and controlled fires demonstrated that the system could reliably detect

fire events across different distances (approximately from 4 m to 121 m) and multiple time intervals (ranging from 1 min to 31 min).

In conclusion, Research Question 3 is evaluated at the conclusion of Chapter 5 by implementing four **ML** models, integrated through a weighted and unweighted voting majority process based on their corresponding anomaly scores. This way, it is shown here the high-endurance and reliability level of the obtainable system when it comes to wildfire ignition events, even an challenging condition.

Hypothesis

The previously proposed Research Questions have culminated into a Hypothesis:

*With the integration of a **WSN** installed in Portuguese forests with **ML** algorithms, it will be possible to alert the firefighters the correct positioning for decision-making.*

Not only the proposed system but the fact that it indeed worked in reality after it had been set up in Serra da Nogueira, with 20 module **WSN**, for three years is also a strong outcome. This system was installed and operated for almost three years. It included the periodic maintenance that the battery had to be replaced every 10 to 11 months, ensuring continuous operation under real-life conditions. During this period, its anomaly threshold of 0.15 – which was generated through the anomaly scores of the four **ML** models and used the weighted voting majority method – was never exceeded. This reinforces the effectiveness of the system to generate very few false alarms and at the same time being capable to detect the real anomalies.

Figure 6.1 illustrates the data collected and forecasted during real-time monitoring. The plot shows the predicted values from the **ML** models alongside the actual sensor data, indicating the system's accuracy and stability in real-world conditions.

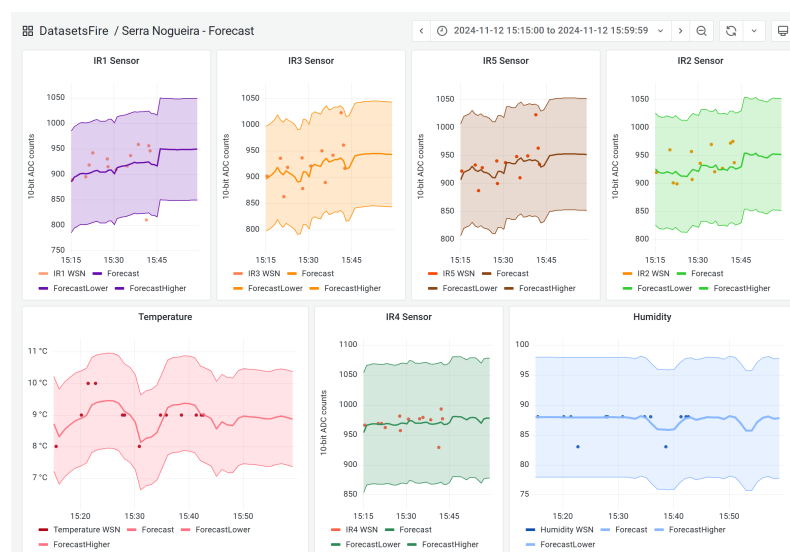


Figure 6.1: Long-term validation of the proposed system in Serra da Nogueira. The plot shows the real-time forecasted and collected data on the Grafana dashboard.

The proposed system fulfilled the core objectives outlined in the hypothesis by combining the strengths of **WSN** technology, open-source cloud systems, and cutting-edge **ML** algorithms. Further, the system by running consistently for an extended time frame, validates the hypothesis that a **WSN** and **ML** algorithms can deliver faultless information for wildfire management in reality. What is more, this demonstration is the proof that the proposed system is practical and scalable, as it provides wildfire monitoring in real-time. In conclusion, the hypothesis has been validated through design theories, practical application, as well as test empery, is the enabler the validation.

6.3 Future Work

While this thesis has successfully established a robust and scalable wildfire monitoring system, several areas remain open for exploration and improvement. These directions aim to enhance the system's efficiency, adaptability, and overall impact on wildfire detection and management. The following points outline key directions for future work:

- Optimizing the placement of **IR** sensors within the current module design presents an opportunity to improve the precision and coverage of fire ignition detection. Exploring different angular arrangements or positions on the module could enhance their sensitivity to environmental changes. Additionally, integrating an embedded **Field Programmable Gate Array (FPGA)**-based system into the modules could enable real-time data processing and pre-filtering directly at the edge. **FPGAs** would allow for faster anomaly detection, adaptive sensor control, and lower latency while significantly reducing energy consumption compared to traditional microcontroller-based systems.
- Extending the system capacity to cover the larger-sized areas with hundreds or thousands of sensor modules faces the challenge of communication, data aggregation, and network management. The main work in the future should be concentrated on developing advanced communication protocols, efficient routing algorithms, and robust data aggregation techniques to support high-density deployments without reducing reliability or data quality.
- **ML** models that are now used for anomaly detection and forecasting have the potential to be developed further. Furthermore, creating the models could result in a manifold increase in predictive capabilities and thus, aid in decision-making. Currently, the modules work based on the concept of omni-directional data transmission which means receiving nothing, only sending the data. Using **ML** in the communication system of the module could allow the detection and interpretation of data patterns to decide which operation mode in the module's firmware should be active. As a consequence, **ML** also could enable the in-time scheduling of data packets, which will reduce the communication needs between **LoRaWAN** gateways.
- Mentioning the wider field of **ML**, there is still a wide range of areas that can bring novelty to systems. One of the feasible models includes the creation of subgroups of modules that are analyzing data in the same **WSN**, which then could be implemented for the upcoming

targeted forecasting outputs. Also, there are said systems to have the ability to include vertical **AI** agents, and the probable way for a system is to include several singly specialized **AI** agents, each uniquely tailoring to the optimization of a specific module or subgroup. This approach would improve analysis accuracy, hasten decision-making processes, and bring about modularity, thus, allowing decades of **WSNs** to work with proper functionalities.

- Lastly, taking the policy and the bills that regulate these systems into account is the major point to guarantee the responsible implementation of those systems. This can be achieved through the observance of data privacy legislation, adherence to ethical rules right from fire monitoring, and following the local and international standards for the sensor networks and environmental monitoring. Of course, the security measures implemented will increase the confidence of the system users and enhance the ability for the stakeholders to use it.

These future directions highlight the potential to build upon this thesis's findings, ensuring that the proposed system continues to evolve and address emerging challenges in wildfire detection and management. Undertaking research as/with these objectives can contribute to further enlarging the system's scope, transforming, and consequently increasing its influence, hence safeguarding the system as a core weapon in the fight with the catastrophic effects of wildfires.

References

- [1] S. S. Sayedi, B. W. Abbott, B. Vannière, B. Leys, D. Colombaroli, G. G. Romera, M. Słowiński, J. C. Aleman, O. Blarquez, A. Feurdean, *et al.*, “Assessing changes in global fire regimes,” *Fire ecology*, vol. 20, no. 1, pp. 1–22, 2024.
- [2] T. J. Hoecker, S. A. Parks, M. Krosby, and S. Z. Dobrowski, “Widespread exposure to altered fire regimes under 2 c warming is projected to transform conifer forests of the western united states,” *Communications Earth & Environment*, vol. 4, no. 1, p. 295, 2023.
- [3] S. Hantson, N. Andela, M. L. Goulden, and J. T. Randerson, “Human-ignited fires result in more extreme fire behavior and ecosystem impacts,” *Nature communications*, vol. 13, no. 1, p. 2717, 2022.
- [4] I. Stavi, “Wildfires in grasslands and shrublands: A review of impacts on vegetation, soil, hydrology, and geomorphology,” *Water*, vol. 11, no. 5, 2019.
- [5] J. Balling, J. Verbesselt, V. De Sy, M. Herold, and J. Reiche, “Exploring archetypes of tropical fire-related forest disturbances based on dense optical and radar satellite data and active fire alerts,” *Forests*, vol. 12, no. 4, 2021.
- [6] R. Nelson, “Us wildfires and mental illness stress health systems,” *The Lancet*, vol. 396, no. 10262, pp. 1546–1547, 2020.
- [7] C. C. of Forest Ministers Conseil canadien des ministres des forêts, “National forestry database - base de données nationales sur les forêts - canada,” July 2021.
- [8] A. I. Filkov, T. Ngo, S. Matthews, S. Telfer, and T. D. Penman, “Impact of australia’s catastrophic 2019/20 bushfire season on communities and environment. retrospective analysis and current trends,” *Journal of Safety Science and Resilience*, vol. 1, no. 1, pp. 44–56, 2020.
- [9] W. J. Requia, “Fires in brazil: health crises and the failure of government action,” *The Lancet Regional Health–Americas*, vol. 39, 2024.
- [10] J. San-Miguel-Ayanz, T. Durrant, R. Boca, P. Maianti, G. Liberta, D. Jacome Felix Oom, A. Branco, D. De Rigo, M. Suarez-Moreno, D. Ferrari, E. Roglia, N. Scionti, M. Broglia, M. Onida, A. Tistan, and P. Löffler, “Forest fires in europe, middle east and north africa 2022,” Tech. Rep. KJ-NA-31-692-EN-N (online), KJ-NA-31-692-EN-C (print), Publications Office of the European Union, Luxembourg (Luxembourg), 2023.
- [11] F. Rego, F. Catry, M. Maia, T. Santos, A. Gravato, I. Castro, F. Moreira, P. Pinto, and J. Almeida, “Análise da rede nacional de postos de vigia em portugal,” *Relatório Final do Projecto. ADISA/CEABN–INESC/INOVAÇÃO. Iniciativa Incêndios Florestais, COTEC Portugal*, 2004.

- [12] F. X. Catry, F. C. Rego, F. L. Bação, and F. Moreira, “Modeling and mapping wildfire ignition risk in portugal,” *International Journal of Wildland Fire*, vol. 18, no. 8, pp. 921–931, 2010.
- [13] F. X. Catry, F. Rego, J. Silva, F. Moreira, A. Camia, C. Ricotta, and M. Conedera, “Fire starts and human activities,” *Towards Integrated Fire Management—Outcomes of the European Project fire Paradox*, Joensuu: European Forest Institute, 2010.
- [14] F. Catry, F. Moreira, J. Pausas, P. Fernandes, F. Rego, *et al.*, “Cork oak vulnerability to fire: The role of bark harvesting,” *Tree Characteristics and*, 2012.
- [15] S. Marques, J. Garcia-Gonzalo, B. Botequim, A. Ricardo, J. Borges, M. Tomé, and M. Oliveira, “Assessing wildfire occurrence probability in pinus pinaster ait. stands in portugal,” *Forest Systems*, vol. 21, pp. 111–120, 2012.
- [16] P. 2020, “Sistemas de videovigilância de prevenção de incêndios a partir de 2017 com apoio po seur.” https://poseur.portugal2020.pt/media/4140/plano_nacional_defesa_floresta_contra_incendios.pdf, 2023. March, 2020.
- [17] H. Kaur and S. K. Sood, “Soft-computing-centric framework for wildfire monitoring, prediction and forecasting,” *Soft Computing*, pp. 1–11, 2019.
- [18] Y. G. Sahin, “A sensor selection model in simultaneous monitoring of multiple types of disaster,” in *Geospatial Informatics IX*, vol. 10992, p. 109920C, International Society for Optics and Photonics, 2019.
- [19] T. Fukuhara, T. Kouyama, S. Kato, R. Nakamura, Y. Takahashi, and H. Akiyama, “Detection of small wildfire by thermal infrared camera with the uncooled microbolometer array for 50-kg class satellite,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 8, pp. 4314–4324, 2017.
- [20] K. Kyzirakos, M. Karpathiotakis, G. Garbis, C. Nikolaou, K. Bereta, I. Papoutsis, T. Herekakis, D. Michail, M. Koubarakis, and C. Kontoes, “Wildfire monitoring using satellite images, ontologies and linked geospatial data,” *Journal of web semantics*, vol. 24, pp. 18–26, 2014.
- [21] Y. E. Aslan, I. Korpeoglu, and Ö. Ulusoy, “A framework for use of wireless sensor networks in forest fire detection and monitoring,” *Computers, Environment and Urban Systems*, vol. 36, no. 6, pp. 614–625, 2012.
- [22] ICNF, “Plano nacional de defesa da floresta contra incêndios.” <http://www2.icnf.pt/portal/florestas/dfci/planos/PNDFCI>, 2023. March, 2020.
- [23] P. Relvas, J. Almeida, F. Rego, and F. Catry, “Estudo para implementação de um sistema de videovigilância florestal no distrito de viseu,” in *Silva, R., Páscoa, F.(eds.) Actas do 5º Congresso Florestal*, 2005.
- [24] J. Verde and J. Zêzere, “Assessment and validation of wildfire susceptibility and hazard in portugal,” *Natural Hazards & Earth System Sciences*, vol. 10, no. 3, 2010.
- [25] J. C. Verde, *Wildfire susceptibility modelling in mainland Portugal*. PhD thesis, Universidade de Lisboa (Portugal), 2015.

- [26] F. Müller, D. Jaeger, and M. Hanewinkel, "Digitization in wood supply—a review on how industry 4.0 will change the forest value chain," *Computers and Electronics in Agriculture*, vol. 162, pp. 206–218, 2019.
- [27] M. Ghobakhloo, "Industry 4.0, digitization, and opportunities for sustainability," *Journal of Cleaner Production*, vol. 252, p. 119869, 2020.
- [28] J. Ladeira, "Lad sensors." <https://www.ladsensors.com/>, 2023. Set, 2020.
- [29] T. Brito, A. I. Pereira, J. Lima, and A. Valente, "Wireless sensor network for ignitions detection: An iot approach," *Electronics*, vol. 9, no. 6, p. 893, 2020.
- [30] T. Brito, B. F. Azevedo, J. Mendes, M. Zorawski, F. P. Fernandes, A. I. Pereira, J. Rufino, J. Lima, and P. Costa, "Data acquisition filtering focused on optimizing transmission in a lorawan network applied to the wsn forest monitoring system," *Sensors*, vol. 23, no. 3, p. 1282, 2023.
- [31] B. F. Azevedo, T. Brito, J. Lima, and A. I. Pereira, "Optimum sensors allocation for a forest fires monitoring system," *Forests*, vol. 12, no. 4, 2021.
- [32] G. E. Mills and L. R. Gay, *Educational research : competencies for analysis and applications*. Boston etc: Pearson, 11th ed. ed., 2016.
- [33] D. Oliveira, M. Costa, S. Pinto, and T. Gomes, "The future of low-end motes in the internet of things: A prospective paper," *Electronics*, vol. 9, no. 1, 2020.
- [34] B. V. Philip, T. Alpcan, J. Jin, and M. Palaniswami, "Distributed real-time iot for autonomous vehicles," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 1131–1140, 2019.
- [35] L. F. P. de Oliveira, L. T. Manera, and P. D. G. D. Luz, "Development of a smart traffic light control system with real-time monitoring," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3384–3393, 2021.
- [36] A. Zielonka, M. Woźniak, S. Garg, G. Kaddoum, M. J. Piran, and G. Muhammad, "Smart homes: How much will they support us? a research on recent trends and advances," *IEEE Access*, vol. 9, pp. 26388–26419, 2021.
- [37] R. Minerva, G. M. Lee, and N. Crespi, "Digital twin in the iot context: A survey on technical features, scenarios, and architectural models," *Proceedings of the IEEE*, vol. 108, no. 10, pp. 1785–1824, 2020.
- [38] J. Lloret, M. Garcia, D. Bri, and S. Sendra, "A wireless sensor network deployment for rural and forest fire detection and verification," *Sensors*, vol. 9, no. 11, pp. 8722–8747, 2009.
- [39] K. S. Adu-Manu, J.-D. Abdulai, F. Engmann, M. Akazue, J. K. Appati, G. E. Baiden, and G. Sarfo-Kantanka, "Wsn architectures for environmental monitoring applications," *Journal of Sensors*, vol. 2022, no. 1, p. 7823481, 2022.
- [40] A. Sairi, S. Labed, B. Miles, and A. Kout, "A review on early forest fire detection using iot-enabled wsn," in *2023 International Conference on Advances in Electronics, Control and Communication Systems (ICAEECS)*, pp. 1–6, 2023.
- [41] J. Chen, J. Klein, Y. Wu, S. Xing, R. Flammang, M. Heibel, and L. Zuo, "A thermoelectric energy harvesting system for powering wireless sensors in nuclear power plants," *IEEE Transactions on Nuclear Science*, vol. 63, no. 5, pp. 2738–2746, 2016.

- [42] N. Saleh, A. Kassem, and A. M. Haidar, "Energy-efficient architecture for wireless sensor networks in healthcare applications," *IEEE Access*, vol. 6, pp. 6478–6486, 2018.
- [43] L. Muduli, D. P. Mishra, and P. K. Jana, "Application of wireless sensor network for environmental monitoring in underground coal mines: A systematic review," *Journal of Network and Computer Applications*, vol. 106, pp. 48–67, 2018.
- [44] T. Cao-hoang and C. N. Duy, "Environment monitoring system for agricultural application based on wireless sensor network," in *2017 Seventh International Conference on Information Science and Technology (ICIST)*, pp. 99–102, 2017.
- [45] M. Ayaz, M. Ammad-Uddin, Z. Sharif, A. Mansour, and E.-H. M. Aggoune, "Internet-of-things (iot)-based smart agriculture: Toward making the fields talk," *IEEE Access*, vol. 7, pp. 129551–129583, 2019.
- [46] D. Komarasamy, M. Gokuldhev, J. J. Hermina, M. Gokulapriya, and M. Manju, "Review for detecting smoke and fire in forest using different technologies," in *IOP Conference Series: Materials Science and Engineering*, vol. 993, p. 012056, IOP Publishing, 2020.
- [47] D. M. Doolin and N. Sitar, "Wireless sensors for wildfire monitoring," in *Smart Structures and Materials 2005: Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems* (M. Tomizuka, ed.), vol. 5765, pp. 477 – 484, International Society for Optics and Photonics, SPIE, 2005.
- [48] N. Sazak and H. Abdullah, "The importance of using wireless sensor networks for forest fire sensing and detection in turkey," in *5th International Advanced Technologies Symposium (IATS'09)*, pp. 13–15, Citeseer, 2009.
- [49] J. Zhang, W. Li, Z. Yin, S. Liu, and X. Guo, "Forest fire detection system based on wireless sensor network," in *2009 4th IEEE Conference on Industrial Electronics and Applications*, pp. 520–523, 2009.
- [50] G. E. Sakr, R. Ajour, A. Khaddaj, B. Saab, A. Salman, O. Helal, I. H. Elhaji, and G. Mitri, "Forest fire detection wireless sensor node," in *Advances in Forest Fire Research*, (Coimbra), Imprensa da Universidade de Coimbra, 2014.
- [51] X. Yan, H. Cheng, Y. Zhao, W. Yu, H. Huang, and X. Zheng, "Real-time identification of smoldering and flaming combustion phases in forest using a wireless sensor network-based multi-sensor system and artificial neural network," *Sensors*, vol. 16, no. 8, 2016.
- [52] Y. Deshpande, K. Savla, C. Lobo, S. Bhattacharjee, and J. Patel, "Forest monitoring system using sensors, wireless communication and image processing," in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, pp. 1–6, 2018.
- [53] P. Dasari, G. K. J. Reddy, and A. Gudipalli, "Forest fire detection using wireless sensor networks," *International journal on smart sensing and intelligent systems*, vol. 13, no. 1, pp. 1–8, 2020.
- [54] S. Sendra, L. García, J. Lloret, I. Bosch, and R. Vega-Rodríguez, "Lorawan network for fire monitoring in rural environments," *Electronics*, vol. 9, no. 3, 2020.
- [55] ICNF, "Instituto de conservação da natureza e das florestas." <https://www.icnf.pt>, 2023. Accessed October, 2023.

- [56] Copernicus Land Monitoring Service, “Tree cover density 2018 - 10m raster,” 2018. This publication has been prepared using European Union’s Copernicus Land Monitoring Service information.
- [57] T. Brito, A. I. Pereira, J. Lima, J. P. Castro, and A. Valente, “Optimal sensors positioning to detect forest fire ignitions,” in *Proceedings of the 9th International Conference on Operations Research and Enterprise Systems - ND2A*, pp. 411–418, INSTICC, SciTePress, 2020.
- [58] J. Vicente and P. Guillemant, “An image processing technique for automatically detecting forest fire,” *International Journal of Thermal Sciences*, vol. 41, no. 12, pp. 1113–1120, 2002.
- [59] M. A. Mahmoud, H. Ren, *et al.*, “Forest fire detection using a rule-based image processing algorithm and temporal variation,” *Mathematical Problems in Engineering*, vol. 2018, 2018.
- [60] Y. Wang, L. Dang, and J. Ren, “Forest fire image recognition based on convolutional neural network,” *Journal of Algorithms & Computational Technology*, vol. 13, p. 1748302619887689, 2019.
- [61] J. Pan, X. Ou, and L. Xu, “A collaborative region detection and grading framework for forest fire smoke using weakly supervised fine segmentation and lightweight faster-rcnn,” *Forests*, vol. 12, no. 6, 2021.
- [62] P. M. Hanamaraddi *et al.*, “A literature study on image processing for forest fire detection,” *IJITR*, vol. 4, no. 1, pp. 2695–2700, 2016.
- [63] A. A. Ahmad and O. S. Osman, “Image processing techniques-based fire detection,” *Sulaimania Journal for Engineering Sciences*, vol. 8, no. 1, 2021.
- [64] A. Gaur, A. Singh, A. Kumar, A. Kumar, and K. Kapoor, “Video flame and smoke based fire detection algorithms: A literature review,” *Fire technology*, vol. 56, pp. 1943–1980, 2020.
- [65] N. Varela, D.-M. Jorge L, A. Ospino, and N. A. Lizardo Zelaya, “Wireless sensor network for forest fire detection,” *Procedia Computer Science*, vol. 175, pp. 435–440, 2020. The 17th International Conference on Mobile Systems and Pervasive Computing (MobiSPC), The 15th International Conference on Future Networks and Communications (FNC), The 10th International Conference on Sustainable Energy Information Technology.
- [66] J. Granda Cantuña, D. Bastidas, S. Solórzano, and J.-M. Clairand, “Design and implementation of a wireless sensor network to detect forest fires,” in *2017 Fourth International Conference on eDemocracy & eGovernment (ICEDEG)*, pp. 15–21, April 2017.
- [67] N. Hamadeh, A. Karouni, B. Daya, and P. Chauvet, “Using correlative data analysis to develop weather index that estimates the risk of forest fires in lebanon & mediterranean: Assessment versus prevalent meteorological indices,” *Case Studies in Fire Safety*, vol. 7, pp. 8–22, 2017.
- [68] D. R. Amol, “Prediction of fire propagation in forest areas using genetic algorithm,” in *ITM Web of Conferences*, vol. 32, p. 03046, EDP Sciences, 2020.
- [69] J. Mabrouki, M. Azrour, D. Dhiba, Y. Farhaoui, and S. E. Hajjaji, “Iot-based data logger for weather monitoring using arduino-based wireless sensor networks with remote graphical application and alerts,” *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 25–32, 2021.

- [70] B. Son, Y.-s. Her, and J. Kim, "A design and implementation of forest-fires surveillance system based on wireless sensor networks for south korea mountains," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 6, no. 9, pp. 124–130, 2006.
- [71] M. Hefeeda and M. Bagheri, "Wireless sensor networks for early detection of forest fires," in *2007 IEEE International Conference on Mobile Adhoc and Sensor Systems*, pp. 1–6, Oct 2007.
- [72] J. Zhang, W. Li, N. Han, and J. Kan, "Forest fire detection system based on a zigbee wireless sensor network," *Frontiers of Forestry in China*, vol. 3, pp. 369–374, 2008.
- [73] A. J. N. Guillermo Molina, Francisco Luna and E. Alba, "An efficient local improvement operator for the multi-objective wireless sensor network deployment problem," *Engineering Optimization*, vol. 43, no. 10, pp. 1115–1139, 2011.
- [74] A. R. Alok Singh and M. Sevaux, "Matheuristic approaches for q-coverage problem versions in wireless sensor networks," *Engineering Optimization*, vol. 45, no. 5, pp. 609–626, 2013.
- [75] E. Zervas, O. Sekkas, S. Hadjiefthymiades, and C. Anagnostopoulos, "Fire detection in the urban rural interface through fusion techniques," in *2007 IEEE International Conference on Mobile Adhoc and Sensor Systems*, pp. 1–6, 2007.
- [76] O. S. da Penha and E. F. Nakamura, "Fusing light and temperature data for fire detection," in *The IEEE symposium on Computers and Communications*, pp. 107–112, 2010.
- [77] J. M. Willi, G. P. Horn, and D. Madrzykowski, "Characterizing a firefighter's immediate thermal environment in live-fire training scenarios," *Fire Technology*, vol. 52, pp. 1667–1696, 2016.
- [78] Y. Klyuchka, K. Afanasenko, and K. Hasanov, "Improvement of fire response efficiency by means of reducing the time of initial fire detection," in *Wood & Fire Safety: Proceedings of the 9th International Conference on Wood & Fire Safety 2020 9*, pp. 432–437, Springer, 2020.
- [79] V. Lawley, M. Lewis, K. Clarke, and B. Ostendorf, "Site-based and remote sensing methods for monitoring indicators of vegetation condition: An australian review," *Ecological Indicators*, vol. 60, pp. 1273–1283, 2016.
- [80] P. Talukdar, A. Das, R. Alagirusamy, *et al.*, "Heat and mass transfer through thermal protective clothing—a review," *International Journal of Thermal Sciences*, vol. 106, pp. 32–56, 2016.
- [81] G. Sankar, A. C. Rao, P. Seshadri, and K. Balasubramanian, "Techniques for measurement of heat flux in furnace waterwalls of boilers and prediction of heat flux—a review," *Applied Thermal Engineering*, vol. 103, pp. 1470–1479, 2016.
- [82] A. T. Lo' Ai and W. Bakhader, "A mobile cloud system for different useful applications," in *2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, pp. 295–298, IEEE, 2016.
- [83] M. I. Mobin, M. Abid-Ar-Rafi, M. N. Islam, and M. R. Hasan, "An intelligent fire detection and mitigation system safe from fire (sff)," *Int. J. Comput. Appl.*, vol. 133, no. 6, pp. 1–7, 2016.

- [84] D. Sulla-Menashe, M. A. Friedl, and C. E. Woodcock, "Sources of bias and variability in long-term landsat time series over canadian boreal forests," *Remote Sensing of Environment*, vol. 177, pp. 206–219, 2016.
- [85] E. C. Atwood, S. Englhart, E. Lorenz, W. Halle, W. Wiedemann, and F. Siegert, "Detection and characterization of low temperature peat fires during the 2015 fire catastrophe in indonesia using a new high-sensitivity fire monitoring satellite sensor (firebird)," *PloS one*, vol. 11, no. 8, p. e0159410, 2016.
- [86] C. Adolf, S. Wunderle, D. Colombaroli, H. Weber, E. Gobet, O. Heiri, J. F. van Leeuwen, C. Bigler, S. E. Connor, M. Gafka, *et al.*, "The sedimentary and remote-sensing reflection of biomass burning in europe," *Global Ecology and Biogeography*, vol. 27, no. 2, pp. 199–212, 2017.
- [87] M. Forkel, W. Dorigo, G. Lasslop, I. Teubner, E. Chuvieco, and K. Thonicke, "A data-driven approach to identify controls on global fire activity from satellite and climate observations (sofia v1)," *Geoscientific Model Development*, vol. 10, no. 12, pp. 4443–4476, 2017.
- [88] D. Fornacca, G. Ren, and W. Xiao, "Performance of three modis fire products (mcd45a1, mcd64a1, mcd14ml), and esa fire_cci in a mountainous area of northwest yunnan, china, characterized by frequent small fires," *Remote Sensing*, vol. 9, no. 11, p. 1131, 2017.
- [89] A. A. Pereira, J. M. Pereira, R. Libonati, D. Oom, A. W. Setzer, F. Morelli, F. Machado-Silva, and L. M. T. De Carvalho, "Burned area mapping in the brazilian savanna using a one-class support vector machine trained by active fires," *Remote Sensing*, vol. 9, no. 11, p. 1161, 2017.
- [90] C. Zhu, H. Kobayashi, Y. Kanaya, and M. Saito, "Size-dependent validation of modis mcd64a1 burned area over six vegetation types in boreal eurasia: Large underestimation in croplands," *Scientific Reports*, vol. 7, no. 1, p. 4181, 2017.
- [91] E. Whitman, M.-A. Parisien, D. K. Thompson, R. J. Hall, R. S. Skakun, and M. D. Flannigan, "Variability and drivers of burn severity in the northwestern canadian boreal forest," *Ecosphere*, vol. 9, no. 2, p. e02128, 2018.
- [92] E. Chuvieco, F. Mouillot, G. R. Van der Werf, J. San Miguel, M. Tanase, N. Koutsias, M. García, M. Yebra, M. Padilla, I. Gitas, *et al.*, "Historical background and current developments for mapping burned area from satellite earth observation," *Remote Sensing of Environment*, vol. 225, pp. 45–64, 2019.
- [93] D. P. Roy, H. Huang, L. Boschetti, L. Giglio, L. Yan, H. H. Zhang, and Z. Li, "Landsat-8 and sentinel-2 burned area mapping-a combined sensor multi-temporal change detection approach," *Remote Sensing of Environment*, vol. 231, p. 111254, 2019.
- [94] L. Wang, X. Quan, B. He, M. Yebra, M. Xing, and X. Liu, "Assessment of the dual polarimetric sentinel-1a data for forest fuel moisture content estimation," *Remote Sensing*, vol. 11, no. 13, p. 1568, 2019.
- [95] A. Beamish, M. K. Reynolds, H. Epstein, G. V. Frost, M. J. Macander, H. Bergstedt, A. Bartsch, S. Kruse, V. Miles, C. M. Tanis, *et al.*, "Recent trends and remaining challenges for optical remote sensing of arctic tundra vegetation: A review and outlook," *Remote Sensing of Environment*, vol. 246, p. 111872, 2020.

- [96] E. Chuvieco, I. Aguado, J. Salas, M. García, M. Yebra, and P. Oliva, "Satellite remote sensing contributions to wildland fire science and management," *Current Forestry Reports*, vol. 6, pp. 81–96, 2020.
- [97] J. Dugué, "Fired equipment safety in the oil & gas industry a review of changes in practices over the last 50 years," *Energy Procedia*, vol. 120, pp. 2–19, 2017.
- [98] P. K. Singh and A. Sharma, "An insight to forest fire detection techniques using wireless sensor networks," in *2017 4th International conference on signal processing, computing and control (ISPCC)*, pp. 647–653, IEEE, 2017.
- [99] S. Veraverbeke, P. Dennison, I. Gitas, G. Hulley, O. Kalashnikova, T. Katagis, L. Kuai, R. Meng, D. Roberts, and N. Stavros, "Hyperspectral remote sensing of fire: State-of-the-art and future perspectives," *Remote Sensing of Environment*, vol. 216, pp. 105–121, 2018.
- [100] M. Naderpour, H. M. Rizeei, N. Khakzad, and B. Pradhan, "Forest fire induced natech risk assessment: A survey of geospatial technologies," *Reliability Engineering & System Safety*, vol. 191, p. 106558, 2019.
- [101] L. Jiang, M. Creyssels, A. Mos, and P. Salizzoni, "Critical velocity in ventilated tunnels in the case of fire plumes and densimetric plumes," *Fire Safety Journal*, vol. 101, pp. 53–62, 2018.
- [102] S. Park, S. H. Park, L. W. Park, S. Park, S. Lee, T. Lee, S. H. Lee, H. Jang, S. M. Kim, H. Chang, *et al.*, "Design and implementation of a smart iot based building and town disaster management system in smart city infrastructure," *Applied Sciences*, vol. 8, no. 11, p. 2239, 2018.
- [103] S. Verma, N. Sood, and A. K. Sharma, "Design of a novel routing architecture for harsh environment monitoring in heterogeneous wsn," *IET Wireless Sensor Systems*, vol. 8, no. 6, pp. 284–294, 2018.
- [104] F. Bu and M. S. Gharajeh, "Intelligent and vision-based fire detection systems: A survey," *Image and vision computing*, vol. 91, p. 103803, 2019.
- [105] Q. Wang, B. Mao, S. I. Stoliarov, and J. Sun, "A review of lithium ion battery failure mechanisms and fire prevention strategies," *Progress in Energy and Combustion Science*, vol. 73, pp. 95–131, 2019.
- [106] J. McNay, R. Puisa, and D. Vassalos, "Analysis of effectiveness of fire safety in machinery spaces," *Fire safety journal*, vol. 108, p. 102859, 2019.
- [107] W. Ejaz, M. A. Azam, S. Saadat, F. Iqbal, and A. Hanan, "Unmanned aerial vehicles enabled iot platform for disaster management," *Energies*, vol. 12, no. 14, p. 2706, 2019.
- [108] A. Gaur, A. Singh, A. Kumar, K. S. Kulkarni, S. Lala, K. Kapoor, V. Srivastava, A. Kumar, and S. C. Mukhopadhyay, "Fire sensing technologies: A review," *IEEE Sensors Journal*, vol. 19, no. 9, pp. 3191–3202, 2019.
- [109] Y. Bao, Y. Huang, M. S. Hoehler, and G. Chen, "Review of fiber optic sensors for structural fire engineering," *Sensors*, vol. 19, no. 4, p. 877, 2019.
- [110] É. Stauffer, "Interpol review of fire investigation 2016–2019," *Forensic science international: Synergy*, vol. 2, pp. 368–381, 2020.

- [111] G. Ma and Z. Wu, "Bim-based building fire emergency management: Combining building users' behavior decisions," *Automation in Construction*, vol. 109, p. 102975, 2020.
- [112] L. Xu, L. Xu, J. Luo, Y. Yan, B.-E. Jia, X. Yang, Y. Gao, and Z. L. Wang, "Hybrid all-in-one power source based on high-performance spherical triboelectric nanogenerators for harvesting environmental energy," *Advanced Energy Materials*, vol. 10, no. 36, p. 2001669, 2020.
- [113] I. Stavroulas, G. Grivas, P. Michalopoulos, E. Liakakou, A. Bougiatioti, P. Kalkavouras, K. M. Fameli, N. Hatzianastassiou, N. Mihalopoulos, and E. Gerasopoulos, "Field evaluation of low-cost pm sensors (purple air pa-ii) under variable urban air quality conditions, in greece," *Atmosphere*, vol. 11, no. 9, p. 926, 2020.
- [114] F. C. Eugenio, C. T. Schons, C. L. Mallmann, M. S. Schuh, P. Fernandes, and T. L. Badin, "Remotely piloted aircraft systems and forests: a global state of the art and future challenges," *Canadian journal of forest research*, vol. 50, no. 8, pp. 705–716, 2020.
- [115] G. Roque and V. S. Padilla, "Lpwan based iot surveillance system for outdoor fire detection," *IEEE Access*, vol. 8, pp. 114900–114909, 2020.
- [116] S. Mandal, S. Gaan, M. Camenzind, S. Annaheim, and R. M. Rossi, "Thermal characterization of fire-protective fabrics," in *Thermal Analysis of Textiles and Fibers*, pp. 355–387, Elsevier, 2020.
- [117] K. Harikumar, J. Senthilnath, and S. Sundaram, "Multi-uav oxyrrhis marina-inspired search and dynamic formation control for forest firefighting," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 2, pp. 863–873, 2018.
- [118] P. S. Sri, B. Kiranmayee, and S. Nagini, "Lte: Advancement of 3g technology," *International Journal of Computer Applications*, vol. 153, no. 8, 2016.
- [119] L. Wan, Z. Guo, Y. Wu, W. Bi, J. Yuan, M. El Kashlan, and L. Hanzo, "4g5g spectrum sharing: Efficient 5g deployment to serve enhanced mobile broadband and internet of things applications," *IEEE Vehicular Technology Magazine*, vol. 13, no. 4, pp. 28–39, 2018.
- [120] J. Lee, S. Han, J. Lee, B. Kang, J. Bae, J. Jang, S. Oh, S. Ahn, S. Kang, Q.-D. Bui, K. Son, H. Lim, D. Jeong, R. Ni, Y. Zuo, I. Jong, C.-W. Yao, S. Heo, T. B. Cho, and I. Kang, "21.6 a sub-6ghz 5g new radio rf transceiver supporting en-dc with 3.15gb/s dl and 1.27gb/s ul in 14nm finfet cmos," in *2019 IEEE International Solid-State Circuits Conference - (ISSCC)*, pp. 354–356, 2019.
- [121] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: An overview," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 855–873, 2017.
- [122] K. Tsakos and E. G. M. Petrakis, "Service oriented architecture for interconnecting lora devices with the cloud," in *Advanced Information Networking and Applications* (L. Barolli, M. Takizawa, F. Xhafa, and T. Enokido, eds.), (Cham), pp. 1082–1093, Springer International Publishing, 2020.
- [123] Q. Zhou, K. Zheng, L. Hou, J. Xing, and R. Xu, "Design and implementation of open lora for iot," *IEEE Access*, vol. 7, pp. 100649–100657, 2019.

- [124] M. R. Seye, B. Ngom, B. Gueye, and M. Diallo, "A study of lora coverage: Range evaluation and channel attenuation model," in *2018 1st International Conference on Smart Cities and Communities (SCCIC)*, pp. 1–4, 2018.
- [125] K.-H. Phung, H. Tran, Q. Nguyen, T. T. Huong, and T.-L. Nguyen, "Analysis and assessment of lorawan," in *2018 2nd International Conference on Recent Advances in Signal Processing, Telecommunications & Computing (SigTelCom)*, pp. 241–246, 2018.
- [126] R. Kufakunesu, G. P. Hancke, and A. M. Abu-Mahfouz, "A survey on adaptive data rate optimization in lorawan: Recent solutions and major challenges," *Sensors*, vol. 20, no. 18, 2020.
- [127] N. Jovalekic, V. Drndarevic, E. Pietrosevoli, I. Darby, and M. Zennaro, "Experimental study of lora transmission over seawater," *Sensors*, vol. 18, no. 9, 2018.
- [128] L. Casals, B. Mir, R. Vidal, and C. Gomez, "Modeling the energy performance of lorawan," *Sensors*, vol. 17, no. 10, 2017.
- [129] K. E. Nolan, W. Guibene, and M. Y. Kelly, "An evaluation of low power wide area network technologies for the internet of things," in *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 439–444, 2016.
- [130] T. Bouguera, J.-F. Diouris, J.-J. Chaillout, R. Jaouadi, and G. Andrieux, "Energy consumption model for sensor nodes based on lora and lorawan," *Sensors*, vol. 18, no. 7, 2018.
- [131] Y. Baiji and P. Sundaravadivel, "iloleak-detect: An iot-based lorawan-enabled oil leak detection system for smart cities," in *2019 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS)*, pp. 262–267, 2019.
- [132] J. Guamán'Quiche, E. Guamán-Quinche, H. Torres-Carrión, W. Chamba-Zaragocin, and F. Alvaréz-Pineda, "Study of routing protocols used in sensor networks for forest fire detection," in *2020 15th Iberian Conference on Information Systems and Technologies (CISTI)*, pp. 1–6, 2020.
- [133] R. Vassoler and G. Zebende, "Dcca cross-correlation coefficient apply in time series of air temperature and air relative humidity," *Physica A: Statistical Mechanics and its Applications*, vol. 391, no. 7, pp. 2438–2443, 2012.
- [134] H. Vu and D. Shin, "Scheduled pre-heating of li-ion battery packs for balanced temperature and state-of-charge distribution," *Energies*, vol. 13, no. 9, p. 2212, 2020.
- [135] T. Brito, B. F. Azevedo, A. Valente, A. I. Pereira, J. Lima, and P. Costa, "Environment monitoring modules with fire detection capability based on iot methodology," in *Science and Technologies for Smart Cities* (S. Paiva, S. I. Lopes, R. Zitouni, N. Gupta, S. F. Lopes, and T. Yonezawa, eds.), (Cham), pp. 211–227, Springer International Publishing, 2021.
- [136] Parlamento Europeu, "O regulamento da ue para responder à crise dos semicondutores." <https://www.europarl.europa.eu/topics/pt/article/20230210ST074502/o-regulamento-da-ue-para-responder-a-crise-dos-semicondutores>. Accessed December, 2023.

- [137] Microchip, *megaAVR Data Sheet - ATmega48A/PA/88A/PA/168A/PA/328/P*. Microchip Technology Inc, 2020. Accessed October, 2023, from <http://ww1.microchip.com/downloads/en/DeviceDoc/ATmega48A-PA-88A-PA-168A-PA-328-P-DS-DS40002061B.pdf>.
- [138] E. K. Payne, S. Lu, Q. Wang, and L. Wu, "Concept of designing thermal condition monitoring system with zigbee/gsm communication link for distributed energy resources network in rural and remote applications," *Processes*, vol. 7, no. 6, 2019.
- [139] V. Semiconductors, *Silicon PIN Photodiode*. Vishay Intertechnology, Inc., 2019. Accessed December, 2023, from <https://www.vishay.com/docs/81503/bpv10nf.pdf>.
- [140] M. Electronics, *DHT11 Humidity & Temperature Sensor*. Mouser Electronics, Inc., 2023. Accessed December, 2023, from <https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf>.
- [141] D. Saha, "Dht11 arduino library." <https://doi.org/10.5281/zenodo.10633701>, 2024.
- [142] M. T. I. Juel, M. S. Ahmed, and T. Islam, "Design of iot based multiple hazards detection and alarming system," in *2019 4th International Conference on Electrical Information and Communication Technology (EICT)*, pp. 1–5, 2019.
- [143] HOPERF, *Low Power Long Range Transceiver Module - Model No.:RFM95W/96W/98W*. Shenzhen Hope Microelectronics Co., 2018. Accessed October, 2023, from <https://www.hoperf.com/modules/lora/RFM95W.html>.
- [144] T. T. Network, "Best practices." <https://www.thethingsnetwork.org/docs/devices/bestpractices/>. Accessed December, 2023.
- [145] T. Brito, M. Zorawski, J. Mendes, B. F. Azevedo, A. I. Pereira, J. Lima, and P. Costa, "Optimizing data transmission in a wireless sensor network based on lorawan protocol," in *International Conference on Optimization, Learning Algorithms and Applications*, pp. 281–293, Springer, 2021.
- [146] R. S. Electronics, "Lightweight power management library." <https://github.com/rocketscream/Low-Power>, 2020. Accessed December, 2023.
- [147] K. M. Tabor and M. B. Holland, "Opportunities for improving conservation early warning and alert systems," *Remote Sensing in Ecology and Conservation*, vol. 7, no. 1, pp. 7–17, 2021.
- [148] M. Lourenço, L. B. Oliveira, J. P. Oliveira, A. Mora, H. Oliveira, and R. Santos, "An integrated decision support system for improving wildfire suppression management," *ISPRS International Journal of Geo-Information*, vol. 10, no. 8, 2021.
- [149] A. G. Raišienė, "Advantages and limitations of integrated management system: the theoretical viewpoint," *Socialinės technologijos*, no. 1, pp. 25–36, 2011.
- [150] R. Birk, B. Baldauf, R. Ohlemacher, and L. Andreoli, "Evaluating a system of systems approach for integrated global weather, climate, and hazard monitoring," in *Atmospheric and Environmental Remote Sensing Data Processing and Utilization IV: Readiness for GEOSS II* (M. D. Goldberg, H. J. Bloom, P. E. Ardanuy, and A. H.-L. Huang, eds.), vol. 7085, p. 708508, International Society for Optics and Photonics, SPIE, 2008.

- [151] A. M. Madni and M. Sievers, "Systems integration: Key perspectives, experiences, and challenges," *Systems Engineering*, vol. 17, no. 1, pp. 37–51, 2014.
- [152] H. I. Kobo, A. M. Abu-Mahfouz, and G. P. Hancke, "A survey on software-defined wireless sensor networks: Challenges and design requirements," *IEEE Access*, vol. 5, pp. 1872–1899, 2017.
- [153] H. I. Kobo, A. M. Abu-Mahfouz, and G. P. Hancke, "A survey on software-defined wireless sensor networks: Challenges and design requirements," *IEEE Access*, vol. 5, pp. 1872–1899, 2017.
- [154] G. Marques and R. Pitarma, "A real-time noise monitoring system based on internet of things for enhanced acoustic comfort and occupational health," *IEEE Access*, vol. 8, pp. 139741–139755, 2020.
- [155] P. Sanguinetti, S. Abdelmohsen, J. Lee, J. Lee, H. Sheward, and C. Eastman, "General system architecture for bim: An integrated approach for design and analysis," *Advanced Engineering Informatics*, vol. 26, no. 2, pp. 317–333, 2012. Knowledge based engineering to support complex product design.
- [156] A. Musaddiq, Y. B. Zikria, O. Hahm, H. Yu, A. K. Bashir, and S. W. Kim, "A survey on resource management in iot operating systems," *IEEE Access*, vol. 6, pp. 8459–8482, 2018.
- [157] Y. Li, "An integrated platform for the internet of things based on an open source ecosystem," *Future Internet*, vol. 10, no. 11, 2018.
- [158] B. Montrucchio, E. Giusto, M. G. Vakili, S. Quer, R. Ferrero, and C. Fornaro, "A densely-deployed, high sampling rate, open-source air pollution monitoring wsn," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 15786–15799, 2020.
- [159] V. Villa, B. Naticchia, G. Bruno, K. Aliev, P. Piantanida, and D. Antonelli, "Iot open-source architecture for the maintenance of building facilities," *Applied Sciences*, vol. 11, no. 12, 2021.
- [160] A. Tamboli, *Build your own IoT platform*. Springer, 2019.
- [161] M. Milenkovic, *Chapter 7: IoT data standards and industry specifications*. Cham: Springer International Publishing, 2020.
- [162] A. Xu, *System Design Interview - An Insider's Guide*. Independently Published, 2020.
- [163] S. Greengard, *The internet of things*. MIT press, 2021.
- [164] J. Valacich and J. George, *Modern systems analysis and design*. London, England: Pearson Education, 9 ed., Aug. 2020.
- [165] S. Lam and A. Xu, *System Design Interview - An Insider's Guide: Volume 2*. Amazon Digital Services LLC - Kdp, 2022.
- [166] A. Nikoukar, S. Raza, A. Poole, M. Güneş, and B. Dezfouli, "Low-power wireless for the internet of things: Standards and applications," *IEEE Access*, vol. 6, pp. 67893–67926, 2018.
- [167] L. Varandas, J. Faria, P. D. Gaspar, and M. L. Aguiar, "Low-cost iot remote sensor mesh for large-scale orchard monitorization," *Journal of Sensor and Actuator Networks*, vol. 9, no. 3, 2020.

- [168] N. Peladarinos, V. Cheimaras, D. Piromalis, K. G. Arvanitis, P. Papageorgas, N. Monios, I. Dogas, M. Stojmenovic, and G. Tsamirsis, "Early warning systems for covid-19 infections based on low-cost indoor air-quality sensors and lpwans," *Sensors*, vol. 21, no. 18, 2021.
- [169] L. M. Fernández-Ahumada, J. Ramírez-Faz, M. Torres-Romero, and R. López-Luque, "Proposal for the design of monitoring and operating irrigation networks based on iot, cloud computing and free hardware technologies," *Sensors*, vol. 19, no. 10, 2019.
- [170] W. A. Jabbar, T. Subramaniam, A. E. Ong, M. I. Shu'Ib, W. Wu, and M. A. de Oliveira, "Lorawan-based iot system implementation for long-range outdoor air quality monitoring," *Internet of Things*, vol. 19, p. 100540, 2022.
- [171] S. M. Karunarathne, M. Dray, L. Popov, M. Butler, C. Pennington, and C. M. Angelopoulos, "A technological framework for data-driven iot systems: Application on landslide monitoring," *Computer Communications*, vol. 154, pp. 298–312, 2020.
- [172] J. Bravo-Arrabal, M. Toscano-Moreno, J. J. Fernandez-Lozano, A. Mandow, J. A. Gomez-Ruiz, and A. García-Cerezo, "The internet of cooperative agents architecture (x-ioca) for robots, hybrid sensor networks, and mec centers in complex environments: A search and rescue case study," *Sensors*, vol. 21, no. 23, 2021.
- [173] F. Delgado-Ferro, J. Navarro-Ortiz, N. Chinchilla-Romero, and J. J. Ramos-Munoz, "A lorawan architecture for communications in areas without coverage: Design and pilot trials," *Electronics*, vol. 11, no. 5, 2022.
- [174] L. Terray, L. Royer, D. Sarramia, C. Achard, E. Bourdeau, P. Chardon, A. Claude, J. Fuchet, P.-J. Gauthier, D. Grimbichler, J. Mezhoud, F. Ogereau, R. Vandaële, and V. Breton, "From sensor to cloud: An iot network of radon outdoor probes to monitor active volcanoes," *Sensors*, vol. 20, no. 10, 2020.
- [175] T. A. Khoa, M. M. Man, T.-Y. Nguyen, V. Nguyen, and N. H. Nam, "Smart agriculture using iot multi-sensors: A novel watering management system," *Journal of Sensor and Actuator Networks*, vol. 8, no. 3, 2019.
- [176] E.-T. Bouali, M. R. Abid, E.-M. Boufounas, T. A. Hamed, and D. Benhaddou, "Renewable energy integration into cloud & iot-based smart agriculture," *IEEE Access*, vol. 10, pp. 1175–1191, 2021.
- [177] C. Bersani, C. Ruggiero, R. Sacile, A. Soussi, and E. Zero, "Internet of things approaches for monitoring and control of smart greenhouses in industry 4.0," *Energies*, vol. 15, no. 10, 2022.
- [178] S. D. Putra, C. O. Sereati, and H. Sutrisno, "Design of iot monitoring system based on lorawan architecture for smart green house," *IOP Conference Series: Earth and Environmental Science*, vol. 1012, p. 012090, apr 2022.
- [179] S. Aydin and M. N. Aydin, "Design and implementation of a smart beehive and its monitoring system using microservices in the context of iot and open data," *Computers and Electronics in Agriculture*, vol. 196, p. 106897, 2022.
- [180] C. Dupont, M. Vecchio, C. Pham, B. Diop, C. Dupont, and S. Koffi, "An open iot platform to promote eco-sustainable innovation in western africa: Real urban and rural testbeds," *Wireless Communications and Mobile Computing*, vol. 2018, no. 1, p. 1028578, 2018.

- [181] M. E. R. Ismail, I. F. A. Shukri, A. Azmi, Y. Yahya, S. A. Ismail, and M. S. Sulaiman, "Development of agronomist station system for water table management at peatland," in *2019 6th International Conference on Research and Innovation in Information Systems (ICRIIS)*, pp. 1–6, 2019.
- [182] J. Yun and J. Woo, "Iot-enabled particulate matter monitoring and forecasting method based on cluster analysis," *IEEE Internet of Things Journal*, vol. 8, no. 9, pp. 7380–7393, 2021.
- [183] S. A. Shah, D. Z. Seker, S. Hameed, and D. Draheim, "The rising role of big data analytics and iot in disaster management: Recent advances, taxonomy and prospects," *IEEE Access*, vol. 7, pp. 54595–54614, 2019.
- [184] Y. Wang, Y. Liu, and M. Xia, "Construction of a multi-source heterogeneous hybrid platform for big data," *Journal of Computational Methods in Sciences and Engineering*, vol. 21, no. 3, pp. 713–722, 2021.
- [185] Y. Zhang and J. Beetz, "Building-cps: Cyber-physical system for building environment monitoring," in *Proceedings of the 38th International Conference of CIB W78*, (Luxembourg), pp. 481–489, University of Ljubljana, October 2021. ISSN: 2706-6568.
- [186] J. P. Lousado, I. M. Pires, E. Zdravevski, and S. Antunes, "Monitoring the health and residence conditions of elderly people, using lora and the things network," *Electronics*, vol. 10, no. 14, 2021.
- [187] J. Bravo-Arrabal, P. Zambrana, J. J. Fernandez-Lozano, J. A. Gomez-Ruiz, J. S. Barba, and A. García-Cerezo, "Realistic deployment of hybrid wireless sensor networks based on zigbee and lora for search and rescue applications," *IEEE Access*, vol. 10, pp. 64618–64637, 2022.
- [188] S. García, D. F. Larios, J. Barbancho, E. Personal, J. M. Mora-Merchán, and C. León, "Heterogeneous lora-based wireless multimedia sensor network multiprocessor platform for environmental monitoring," *Sensors*, vol. 19, no. 16, 2019.
- [189] I. Novkovic, G. B. Markovic, D. Lukic, S. Dragicevic, M. Milosevic, S. Djurdjic, I. Samardzic, T. Lezaic, and M. Tadic, "Gis-based forest fire susceptibility zonation with iot sensor network support, case study—nature park golija, serbia," *Sensors*, vol. 21, no. 19, 2021.
- [190] H. Noura, T. Hatoum, O. Salman, J.-P. Yaacoub, and A. Chehab, "Lorawan security survey: Issues, threats and possible mitigation techniques," *Internet of Things*, vol. 12, p. 100303, 2020.
- [191] M. S. Hadj Sassi and L. Chaari Fourati, "Comprehensive survey on air quality monitoring systems based on emerging computing and communication technologies," *Computer Networks*, vol. 209, p. 108904, 2022.
- [192] N. Y.-R. Douha, M. Bhuyan, S. Kashihara, D. Fall, Y. Taenaka, and Y. Kadobayashi, "A survey on blockchain, sdn and nfv for the smart-home security," *Internet of Things*, vol. 20, p. 100588, 2022.
- [193] L. García, L. Parra, J. M. Jimenez, M. Parra, J. Lloret, P. V. Mauri, and P. Lorenz, "Deployment strategies of soil monitoring wsn for precision agriculture irrigation scheduling in rural areas," *Sensors*, vol. 21, no. 5, 2021.

- [194] R. Singh, M. Baz, C. L. Narayana, M. Rashid, A. Gehlot, S. V. Akram, S. S. Alshamrani, D. Prashar, and A. S. AlGhamdi, “Zigbee and long-range architecture based monitoring system for oil pipeline monitoring with the internet of things,” *Sustainability*, vol. 13, no. 18, 2021.
- [195] A. Moradbeikie, A. Keshavarz, H. Rostami, S. Paiva, and S. I. Lopes, “Gnss-free outdoor localization techniques for resource-constrained iot architectures: A literature review,” *Applied Sciences*, vol. 11, no. 22, 2021.
- [196] T. Ojha, S. Misra, and N. S. Raghuwanshi, “Internet of things for agricultural applications: The state of the art,” *IEEE Internet of Things Journal*, vol. 8, no. 14, pp. 10973–10997, 2021.
- [197] The Things Network, “Lorawan®,” 2024. Accessed June, 2024.
- [198] Node-RED, “Node-red official webpage,” 2024. Accessed June, 2024.
- [199] ThingSpeak, “Thingspeak official webpage,” 2024. Accessed June, 2024.
- [200] Apache Cassandra, “Cassandra official webpage,” 2024. Accessed June, 2024.
- [201] InfluxData, “Influxdb official webpage,” 2024. Accessed June, 2024.
- [202] MongoDB, “Mongodb official webpage,” 2024. Accessed June, 2024.
- [203] Prometheus, “Prometheus official webpage,” 2024. Accessed June, 2024.
- [204] Elastic Kibana, “Kibana official webpage,” 2024. Accessed June, 2024.
- [205] Grafana Labs, “Grafana official webpage,” 2024. Accessed June, 2024.
- [206] T. Brito, J. Mendes, M. Zorawski, B. F. Azevedo, A. Khalifeh, F. P. Fernandes, A. I. Pereira, J. Lima, and P. Costa, “Map coverage of lorawan signal’s employing gps from mobile devices,” in *Optimization, Learning Algorithms and Applications* (A. I. Pereira, A. Košir, F. P. Fernandes, M. F. Pacheco, J. P. Teixeira, and R. P. Lopes, eds.), (Cham), pp. 589–603, Springer International Publishing, 2022.
- [207] F. F. Jurado-Lasso, L. Marchegiani, J. F. Jurado, A. M. Abu-Mahfouz, and X. Fafoutis, “A survey on machine learning software-defined wireless sensor networks (ml-sdwsns): Current status and major challenges,” *IEEE Access*, vol. 10, pp. 23560–23592, 2022.
- [208] M. Mohinur Rahaman and M. Azharuddin, “Wireless sensor networks in agriculture through machine learning: A survey,” *Computers and Electronics in Agriculture*, vol. 197, p. 106928, 2022.
- [209] A. Bagwari, J. Logeshwaran, K. Usha, K. Raju, M. H. Alsharif, P. Uthansakul, and M. Uthansakul, “An enhanced energy optimization model for industrial wireless sensor networks using machine learning,” *IEEE Access*, vol. 11, pp. 96343–96362, 2023.
- [210] K. Upreti, N. Kumar, M. S. Alam, A. Verma, M. Nandan, and A. K. Gupta, “Machine learning-based congestion control routing strategy for healthcare iot enabled wireless sensor networks,” in *2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, pp. 1–6, 2021.
- [211] Y. Liu, X. Ma, Y. Li, Y. Tie, Y. Zhang, and J. Gao, “Water pipeline leakage detection based on machine learning and wireless sensor networks,” *Sensors*, vol. 19, no. 23, 2019.

- [212] M. P. Nath, S. N. Mohanty, and S. B. B. Priyadarshini, "Application of machine learning in wireless sensor network," in *2021 8th International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 7–12, 2021.
- [213] R. Kaur, J. Kaur Sandhu, and L. Sapra, "Machine learning technique for wireless sensor networks," in *2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, pp. 332–335, 2020.
- [214] A. Gangwar, S. Singh, R. Mishra, and S. Prakash, "The state-of-the-art in air pollution monitoring and forecasting systems using iot, big data, and machine learning," *Wireless Personal Communications*, vol. 130, pp. 1699–1729, June 2023.
- [215] M. Mittal, R. P. de Prado, Y. Kawai, S. Nakajima, and J. E. Muñoz-Expósito, "Machine learning techniques for energy efficiency and anomaly detection in hybrid wireless sensor networks," *Energies*, vol. 14, no. 11, 2021.
- [216] S. Ifzarne, H. Tabbaa, I. Hafidi, and N. Lamghari, "Anomaly detection using machine learning techniques in wireless sensor networks," *Journal of Physics: Conference Series*, vol. 1743, p. 012021, jan 2021.
- [217] A. Srivastava and M. R. Bharti, "Hybrid machine learning model for anomaly detection in unlabelled data of wireless sensor networks," *Wireless Personal Communications*, vol. 129, pp. 2693–2710, April 2023.
- [218] I. G. A. Poornima and B. Paramasivan, "Anomaly detection in wireless sensor network using machine learning algorithm," *Computer Communications*, vol. 151, pp. 331–337, 2020.
- [219] A. Haque, N.-U.-R. Chowdhury, H. Soliman, M. S. Hossen, T. Fatima, and I. Ahmed, "Wireless sensor networks anomaly detection using machine learning: A survey," in *Intelligent Systems and Applications* (K. Arai, ed.), (Cham), pp. 491–506, Springer Nature Switzerland, 2024.
- [220] R. K. Dwivedi, A. K. Rai, and R. Kumar, "Outlier detection in wireless sensor networks using machine learning techniques: A survey," in *2020 International Conference on Electrical and Electronics Engineering (ICE3)*, pp. 316–321, 2020.
- [221] W. Benzekri, A. E. Moussati, O. Moussaoui, and M. Berrajaa, "Early forest fire detection system using wireless sensor network and deep learning," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 11, no. 5, 2020.
- [222] H. Sharma, A. Haque, and F. Blaabjerg, "Machine learning in wireless sensor networks for smart cities: A survey," *Electronics*, vol. 10, no. 9, 2021.
- [223] W. Chen, Z. Milosevic, F. A. Rabhi, and A. Berry, "Real-time analytics: Concepts, architectures, and ml/ai considerations," *IEEE Access*, vol. 11, pp. 71634–71657, 2023.
- [224] K. Guravaiah, A. Kavitha, and R. L. Velusamy, "Data collection protocols in wireless sensor networks," in *Wireless Sensor Networks* (S. S. Yellampalli, ed.), ch. 6, Rijeka: IntechOpen, 2020.
- [225] O. Gurewitz, M. Shifrin, and E. Dvir, "Data gathering techniques in wsn: A cross-layer view," *Sensors*, vol. 22, no. 7, 2022.

- [226] P. Arroyo, J. L. Herrero, J. I. Suárez, and J. Lozano, “Wireless sensor network combined with cloud computing for air quality monitoring,” *Sensors*, vol. 19, no. 3, 2019.
- [227] S. K. Jaiswal and A. K. Dwivedi, “A security and application of wireless sensor network: A comprehensive study,” in *2023 International Conference on IoT, Communication and Automation Technology (ICICAT)*, pp. 1–5, 2023.
- [228] B.-S. Kim, K.-I. Kim, B. Shah, F. Chow, and K. H. Kim, “Wireless sensor networks for big data systems,” *Sensors*, vol. 19, no. 7, 2019.
- [229] B.-S. Kim, B. Suh, K.-I. Kim, I. J. Seo, H. B. Lee, and J. S. Gong, “An enhanced tree routing through multi-criteria decision making over wireless sensor networks,” in *2022 IEEE 19th International Conference on Mobile Ad Hoc and Smart Systems (MASS)*, pp. 570–576, 2022.
- [230] H. M. A. Fahmy, *Concepts, applications, experimentation and analysis of wireless sensor networks*. Springer Nature, 2020.
- [231] T. Wang, J. Zeng, Y. Lai, Y. Cai, H. Tian, Y. Chen, and B. Wang, “Data collection from wsns to the cloud based on mobile fog elements,” *Future Generation Computer Systems*, vol. 105, pp. 864–872, 2020.
- [232] R.-S. Liu and Y.-C. Chen, “Robust data collection for energy-harvesting wireless sensor networks,” *Computer Networks*, vol. 167, p. 107025, 2020.
- [233] X. Ma, P. Zhang, O. Theel, and J. Wei, “Gathering data with packet-in-packet in wireless sensor networks,” *Computer Networks*, vol. 170, p. 107124, 2020.
- [234] M. Younan, S. Khattab, and R. Bahgat, “From the wireless sensor networks (wsns) to the web of things (wot): An overview,” *Journal of Intelligent Systems and Internet of Things*, vol. 4, no. 2, pp. 56–68, 2021.
- [235] W. Osamy, A. M. Khedr, A. Salim, A. I. AlAli, and A. A. El-Sawy, “Recent studies utilizing artificial intelligence techniques for solving data collection, aggregation and dissemination challenges in wireless sensor networks: A review,” *Electronics*, vol. 11, no. 3, 2022.
- [236] I. Ali, I. Ahmedy, A. Gani, M. U. Munir, and M. H. Anisi, “Data collection in studies on internet of things (iot), wireless sensor networks (wsns), and sensor cloud (sc): Similarities and differences,” *IEEE Access*, vol. 10, pp. 33909–33931, 2022.
- [237] T. Khan, “Ultra-low-power architecture for the detection and notification of wildfires using the internet of things,” *IoT*, vol. 4, no. 1, pp. 1–26, 2023.
- [238] A. A. A. Alkhatib and K. M. Jaber, “Fdpa internet of things system for forest fire detection, prediction and behaviour analysis,” *IET Wireless Sensor Systems*, vol. 14, no. 3, pp. 56–71, 2024.
- [239] S. Zhang and M. Pan, “Advancing forest-fire management: Exploring sensor networks, data mining techniques, and svm algorithm for prediction,” *Prevention and Treatment of Natural Disasters*, vol. 3, Jul. 2024.
- [240] S. M. Thaung, H. M. Tun, K. K. K. Win, and M. M. Than, “Exploratory data analysis based on remote health care monitoring system by using iot,” *Communications*, vol. 8, no. 1, pp. 1–8, 2020.

- [241] L. Ordonez-Ante, G. Van Seghbroeck, T. Wauters, B. Volckaert, and F. De Turck, “Explora: Interactive querying of multidimensional data in the context of smart cities,” *Sensors*, vol. 20, no. 9, 2020.
- [242] D. S. V. Medeiros, H. N. C. Neto, M. A. Lopez, L. C. S. Magalhães, N. C. Fernandes, A. B. Vieira, E. F. Silva, and D. M. F. Mattos, “A survey on data analysis on large-scale wireless networks: online stream processing, trends, and challenges,” *Journal of Internet Services and Applications*, vol. 11, no. 1, pp. 1–48, 2020.
- [243] W. Li, Y. Liu, L. Yang, and Y. Chen, “Deformation characteristics and exploratory data analysis of rainfall-induced rotational landslide: A case study of the zhutoushan landslide in nanjing, china,” *Natural Hazards and Earth System Sciences Discussions [preprint]*, pp. 1–11, 2020.
- [244] A. Kanáliková, “Data analysis in wsn and the draft of system for data analysis,” in *2020 18th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, pp. 276–282, 2020.
- [245] M. Kumar, A. Kumar, and V. S. Palaparthi, “Soil sensors-based prediction system for plant diseases using exploratory data analysis and machine learning,” *IEEE Sensors Journal*, vol. 21, no. 16, pp. 17455–17468, 2021.
- [246] S. Rizanov, A. Stoyanova, and D. Todorov, “A lora-based data acquisition system for wildfire early detection,” in *Advances in Computing and Network Communications* (S. M. Thampi, E. Gelenbe, M. Atiquzzaman, V. Chaudhary, and K.-C. Li, eds.), (Singapore), pp. 261–275, Springer Singapore, 2021.
- [247] J. Ananthi, N. Sengottaiyan, S. Anbukaruppusamy, K. Upreti, and A. K. Dubey, “Forest fire prediction using iot and deep learning,” *International Journal of Advanced Technology and Engineering Exploration*, vol. 9, no. 87, pp. 246–256, 2022.
- [248] S. S. Patil and B. Vidyavathi, “A machine learning approach to weather prediction in wireless sensor networks,” *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 13, no. 1, 2022.
- [249] S. Salmi and L. Oughdir, “Performance evaluation of deep learning techniques for dos attacks detection in wireless sensor network,” *Journal of Big Data*, vol. 10, no. 1, p. 17, 2023.
- [250] P. Cortez and A. Morais, “Forest Fires.” UCI Machine Learning Repository, 2007. DOI: <https://doi.org/10.24432/C5D88D>.
- [251] A. Kansal, Y. Singh, N. Kumar, and V. Mohindru, “Detection of forest fires using machine learning technique: A perspective,” in *2015 Third International Conference on Image Information Processing (ICIIP)*, pp. 241–245, 2015.
- [252] F. Abid, “A survey of machine learning algorithms based forest fires prediction and detection systems,” *Fire technology*, vol. 57, no. 2, pp. 559–590, 2021.
- [253] R. K. Dwivedi, “Density-based machine learning scheme for outlier detection in smart forest fire monitoring sensor cloud,” *International Journal of Cloud Applications and Computing (IJCAC)*, vol. 12, no. 1, pp. 1–16, 2022.

- [254] U. Dampage, L. Bandaranayake, R. Wanasinghe, K. Kottahachchi, and B. Jayasanka, “Forest fire detection system using wireless sensor networks and machine learning,” *Scientific reports*, vol. 12, no. 1, p. 46, 2022.
- [255] Y. Chabchoub, M. U. Togbe, A. Boly, and R. Chiky, “An in-depth study and improvement of isolation forest,” *IEEE Access*, vol. 10, pp. 10219–10237, 2022.
- [256] A. Secilmis, N. Aksu, F. A. Dael, I. Shayea, and A. A. El-Saleh, “Machine learning-based fire detection: A comprehensive review and evaluation of classification models,” *JOIV: International Journal on Informatics Visualization*, vol. 7, no. 3-2, pp. 1982–1988, 2023.
- [257] A. Sathiya, A. K. Velmurugan, P. Santhuja, and G. Indra, “Cognitive fire alert framework for realizing fire in unfriendly environment using wireless sensor network,” in *2023 7th International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 998–1003, 2023.
- [258] G. Ganapavarapu, S. Mukherjee, N. Martinez Gil, K. Sarpatwar, A. Rajasekharan, A. Dhurandhar, V. Arya, and R. Vaculin, “Ai explainability 360 toolkit for time-series and industrial use cases,” in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD ’23*, (New York, NY, USA), pp. 5777–5778, Association for Computing Machinery, 2023.
- [259] D. Li, J. Li, Y. Lin, H. Chen, G. Yang, and W. Chen, “Short-term power prediction for centralized photovoltaic plants based on l1stnet-attention,” in *2023 IEEEIAS Industrial and Commercial Power System Asia (I&CPS Asia)*, pp. 2374–2379, 2023.
- [260] W. Liu, P. Lei, D. Xu, and X. Zhu, “Anomaly recognition, diagnosis and prediction of massive data flow based on time-gan and dbscan for power dispatching automation system,” *Processes*, vol. 11, no. 9, 2023.
- [261] D. Ullmann, O. Taran, and S. Voloshynovskiy, “Multivariate time series information bottleneck,” *Entropy*, vol. 25, no. 5, 2023.
- [262] D. Park, A.-R. Kim, C.-S. Lee, and H.-J. Chang, “Short-term korea east-sea temperature forecasting approach based on seq2seq model using multi parameters,” in *2023 14th International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 1230–1234, 2023.
- [263] M. T. Pillay, N. Minakawa, Y. Kim, N. Kgalane, J. V. Ratnam, S. K. Behera, M. Hashizume, and N. Sweijd, “Utilizing a novel high-resolution malaria dataset for climate-informed predictions with a deep learning transformer model,” *Scientific Reports*, vol. 13, no. 1, p. 23091, 2023.
- [264] H. Xue and F. D. Salim, “Promptcast: A new prompt-based learning paradigm for time series forecasting,” *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–14, 2023.
- [265] Y. Shi, C. Feng, and S. Yang, “Predictive modeling of forest fires in yunnan province: An integration of arima and stepwise regression analysis,” *Applied Sciences*, vol. 14, no. 1, 2024.
- [266] A. F. Ansari, L. Stella, C. Turkmen, X. Zhang, P. Mercado, H. Shen, O. Shchur, S. S. Rangapuram, S. P. Arango, S. Kapoor, J. Zschiegner, D. C. Maddix, H. Wang, M. W. Mahoney, K. Torkkola, A. G. Wilson, M. Bohlke-Schneider, and Y. Wang, “Chronos: Learning the language of time series,” 2024.

- [267] V. Ekambaram, A. Jati, P. Dayama, S. Mukherjee, N. H. Nguyen, W. M. Gifford, C. Reddy, and J. Kalagnanam, “Tiny time mixers (ttms): Fast pre-trained models for enhanced zero/few-shot forecasting of multivariate time series.,” *CoRR*, 2024.
- [268] Y. Liang, H. Wen, Y. Nie, Y. Jiang, M. Jin, D. Song, S. Pan, and Q. Wen, “Foundation models for time series analysis: A tutorial and survey,” in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, (New York, NY, USA), pp. 6555–6565, Association for Computing Machinery, 2024.
- [269] J. A. Miller, M. Aldosari, F. Saeed, N. H. Barna, S. Rana, I. B. Arpinar, and N. Liu, “A survey of deep learning and foundation models for time series forecasting,” 2024.
- [270] A. Das, W. Kong, R. Sen, and Y. Zhou, “A decoder-only foundation model for time-series forecasting,” 2024.
- [271] J. D. Hunter, “Matplotlib: A 2d graphics environment,” *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.
- [272] T. M. D. Team, “Matplotlib: Visualization with python,” Nov. 2024. Software.
- [273] T. pandas development team, “pandas-dev/pandas: Pandas,” Sept. 2024. Software.
- [274] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İlhan Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, A. Vijaykumar, A. P. Bardelli, A. Rothberg, A. Hilboll, A. Kloeckner, A. Scopatz, A. Lee, A. Rokem, C. N. Woods, C. Fulton, C. Masson, C. Häggström, C. Fitzgerald, D. A. Nicholson, D. R. Hagen, D. V. Pasechnik, E. Olivetti, E. Martin, E. Wieser, F. Silva, F. Lenders, F. Wilhelm, G. Young, G. A. Price, G.-L. Ingold, G. E. Allen, G. R. Lee, H. Audren, I. Probst, J. P. Dietrich, J. Silterra, J. T. Webber, J. Slavič, J. Nothman, J. Buchner, J. Kulick, J. L. Schönberger, J. V. de Miranda Cardoso, J. Reimer, J. Harrington, J. L. C. Rodríguez, J. Nunez-Iglesias, J. Kuczynski, K. Tritz, M. Thoma, M. Newville, M. Kümmerer, M. Bolingbroke, M. Tartre, M. Pak, N. J. Smith, N. Nowaczyk, N. Shebanov, O. Pavlyk, P. A. Brodtkorb, P. Lee, R. T. McGibbon, R. Feldbauer, S. Lewis, S. Tygier, S. Sievert, S. Vigna, S. Peterson, S. More, T. Pudlik, T. Oshima, T. J. Pingel, T. P. Robitaille, T. Spura, T. R. Jones, T. Cera, T. Leslie, T. Zito, T. Krauss, U. Upadhyay, Y. O. Halchenko, Y. Vázquez-Baeza, and S. . Contributors, “Scipy 1.0: fundamental algorithms for scientific computing in python,” *Nature Methods*, vol. 17, no. 3, pp. 261–272, 2020.
- [275] M. L. Waskom, “seaborn: statistical data visualization,” *Journal of Open Source Software*, vol. 6, no. 60, p. 3021, 2021.
- [276] T. Cokelaer, A. Kravchenko, lahdjirayhan, msat59, V. Ferrari, A. Varma, B. L. C. E. Stringari, C. Brueffer, E. Broda, E. Pruesse, K. Singaravelan, S. A. Russo, Z. Li, mark padgham, and negodfre, “cokelaer/fitter: v1.7.1,” June 2024. Software.
- [277] Skipper Seabold and Josef Perktold, “Statsmodels: Econometric and Statistical Modeling with Python,” in *Proceedings of the 9th Python in Science Conference* (Stéfan van der Walt and Jarrod Millman, eds.), pp. 92 – 96, 2010.

- [278] C. Chen, Q. Zhang, M. H. Kashani, C. Jun, S. M. Bateni, S. S. Band, S. S. Dash, and K.-W. Chau, "Forecast of rainfall distribution based on fixed sliding window long short-term memory," *Engineering Applications of Computational Fluid Mechanics*, vol. 16, no. 1, pp. 248–261, 2022.
- [279] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422, 2008.
- [280] V. A. Villar, M. Cranmer, E. Berger, G. Contardo, S. Ho, G. Hosseinzadeh, and J. Y.-Y. Lin, "A deep-learning approach for live anomaly detection of extragalactic transients," *The Astrophysical Journal Supplement Series*, vol. 255, p. 24, aug 2021.
- [281] A. Petkovski and V. Shehu, "Anomaly detection on univariate sensing time series data for smart aquaculture using k-means, isolation forest, and local outlier factor," in *2023 12th Mediterranean Conference on Embedded Computing (MECO)*, pp. 1–5, 2023.
- [282] A. Bâra, A. G. Văduva, and S.-V. Oprea, "Anomaly detection in weather phenomena: News and numerical data-driven insights into the climate change in romania's historical regions," *International Journal of Computational Intelligence Systems*, vol. 17, no. 1, p. 134, 2024.
- [283] J. Herzen, F. Lässig, S. G. Piazzetta, T. Neuer, L. Tafti, G. Raille, T. V. Pottelbergh, M. Pasięka, A. Skrodzki, N. Huguenin, M. Dumonal, J. Kościsz, D. Bader, F. Gusset, M. Benheddi, C. Williamson, M. Kosinski, M. Petrik, and G. Grosch, "Darts: User-friendly modern machine learning for time series," *Journal of Machine Learning Research*, vol. 23, no. 124, pp. 1–6, 2022.
- [284] G. Research, "A decoder-only foundation model for time series forecasting." <https://research.google/blog/a-decoder-only-foundation-model-for-time-series-forecasting/>, 2024. Accessed: 2024-07-18.
- [285] G. Research, "Timesfm." <https://github.com/google-research/timesfm?tab=readme-ov-file>, 2024. Accessed: 2024-07-18.
- [286] G. Research, "Timesfm 1.0-200m." <https://huggingface.co/google/timesfm-1.0-200m>, 2024. Accessed: 2024-07-18.
- [287] S. Rayana and L. Akoglu, "Less is more: Building selective anomaly ensembles," *ACM Trans. Knowl. Discov. Data*, vol. 10, May 2016.
- [288] D. Schweiz, *RND 455-01087 Plastic Enclosure 90x115x55mm Dark Grey ABS IP67, RND*. Distrelec Schweiz AG, 2023. Accessed December, 2023, from https://media.distrelec.com/Web/Downloads/_d/wg/RND_455-01087_eng_dwg.pdf.

Appendix A

Wireless Module Schematic, PCB Layout and Gerber view

This appendix presents a detailed view of the module's schematic circuit, PCB layout, and Gerber files, whose development can be seen in Chapter 3. The content shown in this appendix makes it possible to obtain information about how each electronic component was arranged, connected, and organized in the circuit. The following subsections will present the complete circuit schematic in Section A.1, the PCB layout in Section A.2, and the top, bottom, and drill Gerber representations in Section A.3.

A.1 Wireless Module Schematic Circuit

In this subsection, the complete schematic circuit of the module is demonstrated in Figure A.1, highlighting the connection between the sensors, the transceiver, the microcontroller, and the other components. This project guides an understanding of the module's signal flow and power distribution, providing a visual map of the connections that allow all elements to operate together.

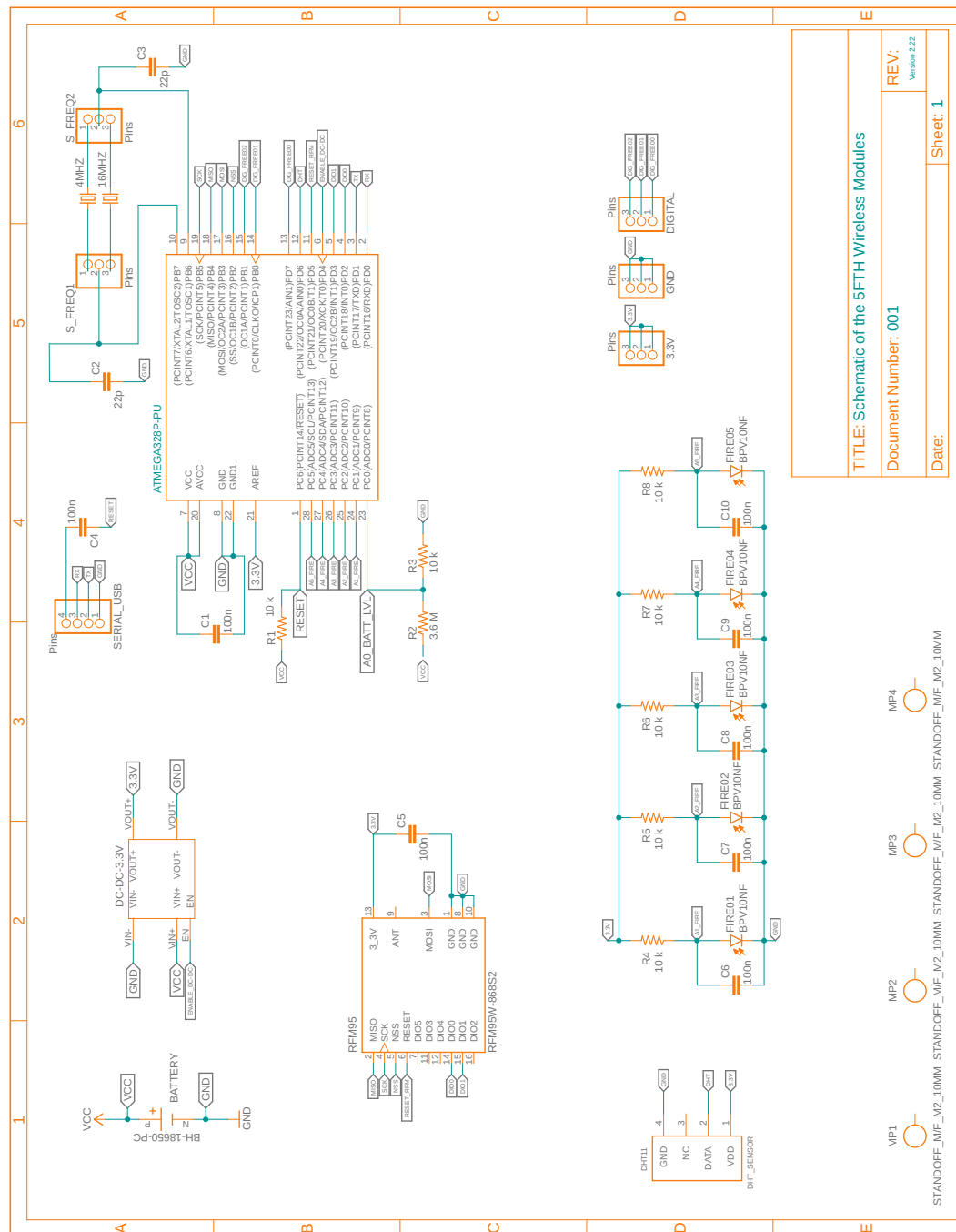


Figure A.1: Full Wireless module schematic.

A.2 Wireless Module PCB Layout

Figure A.2 presents the layout of the wireless module's PCB. This layout shows how each component was strategically positioned on the board to optimize its size, reduce possible noise (essential for the transceiver's proper functioning), and distribute the sensors according to their functionalities.

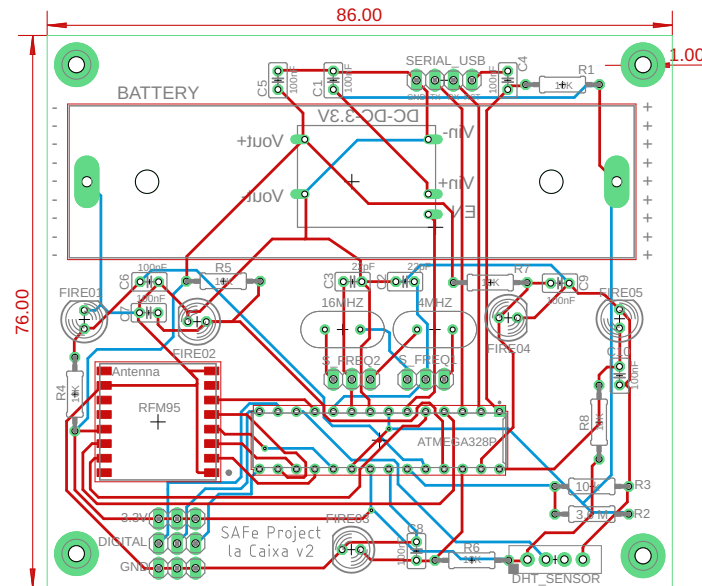
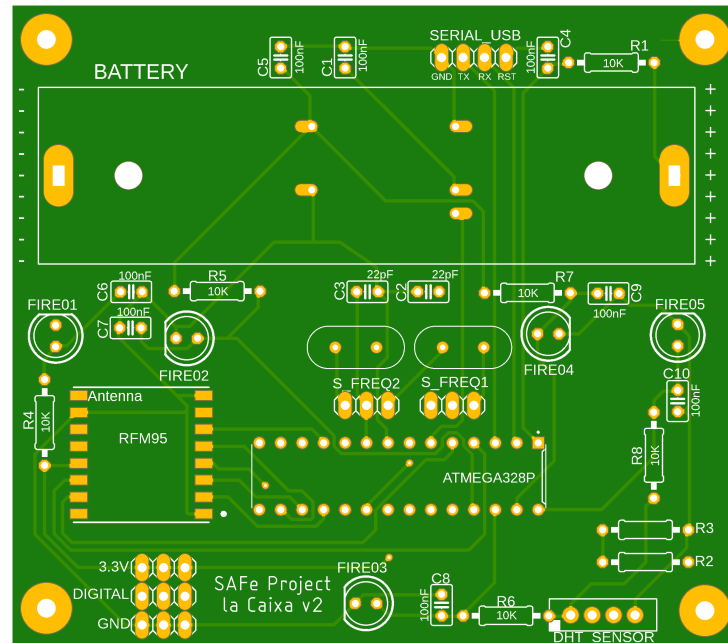


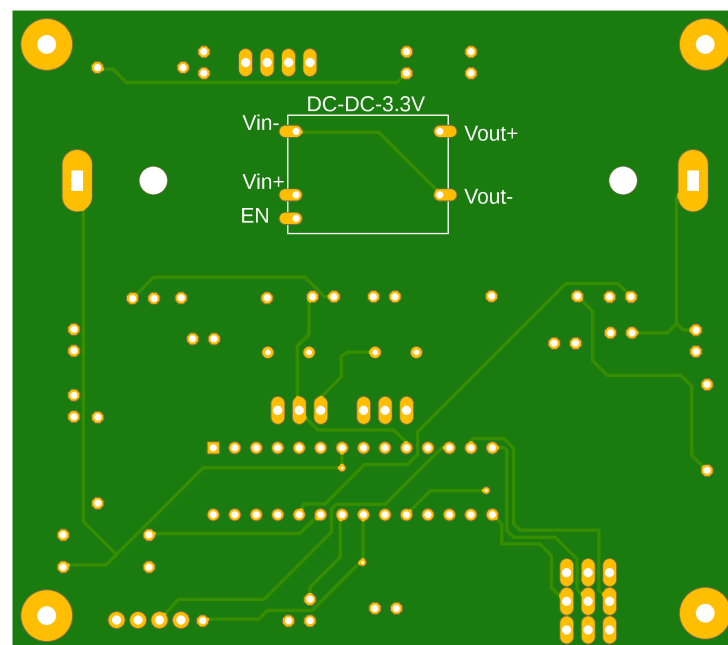
Figure A.2: PCB layout of the wireless module.

A.3 Wireless Module - Gerber files view

The Gerber files view included in this subsection and available in Figure A.3 represent the top (Figure A.3a), bottom (Figure A.3b), and drill layers (Figure A.3c) of the PCB. These files are essential for printed circuit manufacturing, detailing the precise placement of tracks, component pads, and lines.

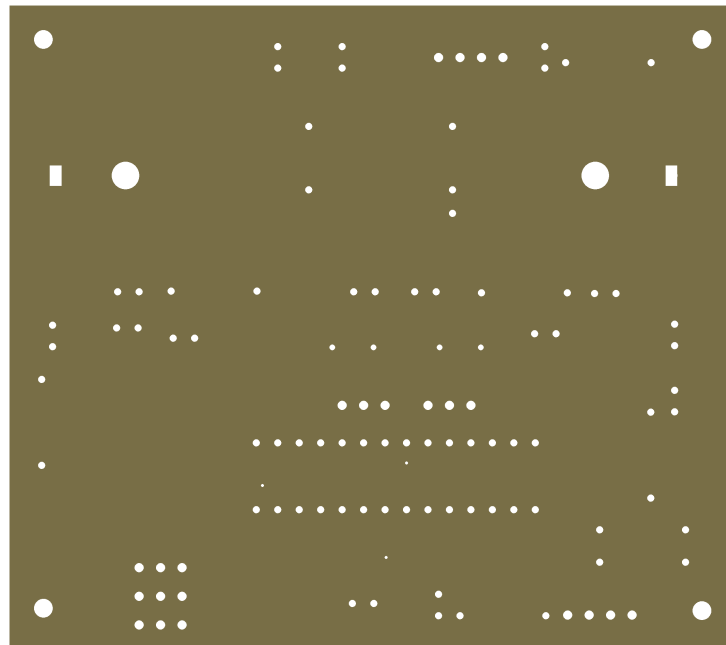


(a) Top view.



(b) Bottom view.

Figure A.3: Gerber files view of the Wireless Module.



(c) Drill layer view.

Figure A.3: Gerber files view of the Wireless Module.

Appendix B

Wireless Module Enclosure

This appendix contains the measurements of the enclosure used in the project, whose model is RND 455-01087 and is produced by RND Components. Therefore, following Figure B.1, it is possible to obtain all the distances for each part of the box.

Appendix C

Source Code Used in System Development

This appendix contains the source code referenced in the system development chapter. Each section includes code snippets and detailed explanations of their functionalities within the system. The provided code demonstrates key aspects of the system's architecture, including data collection, processing, and integration of various technologies. This detailed presentation aims to offer a clear understanding of the technical implementations and facilitate replication or further development by other researchers or developers.

C.1 Code Implementations for TTN

This section presents the source code utilized in integrating the system with **TTN**. The provided code demonstrates the configuration and management of LoRaWAN devices and data decoder to data transmission using the TTN infrastructure.

C.1.1 Section 4.3.1: LoRaWAN Data Decoding on TTN

An algorithm designed to decode LoRaWAN data within **TTN** platform is presented (mentioned in Section 4.4.1). The Algorithm C.1 processes incoming payloads from LoRaWAN devices, translating the raw data into meaningful information that can be utilized for various applications. By utilizing this decoding algorithm, it is possible to transform data from bytes into JSON format. This transformation ensures that the data is easily readable and can be further processed or analyzed for various applications.

Listing C.1: JavaScript Decoder Function

```
1 function Decoder(bytes, port) {  
2   // Decode an uplink message from a buffer  
3   var decoded = {};  
4   var S = [0x0000, 0x0000, 0x0000, 0x0000, 0x0000];  
5   var i;  
6   for (i = 0; i < 5; i++) {
```

```

7     S[i] = (bytes[2 * i + 1] << 8) | bytes[2 * i];
8 }
9 decoded.tem = ((S[0] & 0xFC00) >> 10) | ((S[1] & 0x0C00) >> 4);
10 decoded.h = ((S[1] & 0xF000) >> 12) | ((S[2] & 0x3C00) >> 6);
11 decoded.sen1 = ((S[2] & 0xC000) >> 14) | ((S[3] & 0xFC00) >> 8) | ((S[4] & 0x0C00) >> 2);
12 decoded.AUX = ((S[4] & 0xF000) >> 12);
13 decoded.b = S[0] & 0x03FF;
14 decoded.sen2 = S[1] & 0x03FF;
15 decoded.sen3 = S[2] & 0x03FF;
16 decoded.sen4 = S[3] & 0x03FF;
17 decoded.sen5 = S[4] & 0x03FF;
18 return decoded;
19 }

```

C.1.2 Section 4.4.1: TTN Parameter Definitions for Firmware

This subsection provides an example of defining parameters for **TTN** in the firmware (mentioned in Section 4.4.1). The Algorithm C.2 shows how to set the device address, application session key, and network session key. These parameters are crucial for establishing secure communication between the LoRaWAN device and the TTN infrastructure using **ABP** proper authentication and data integrity.

Listing C.2: Example of TTN parameter definitions for the firmware.

```

1 static const u4_t DEVADDR = { DevAddr_TTN };
2 static const u1_t PROGMEM APPSKEY[16] = { AppSKey_TTN };
3 static const PROGMEM u1_t NWKSKEY[16] = { NwkSKey_TTN };

```

C.2 InfluxDB Codes for Read and Write Queries

This section provides the source code used for performing read and write queries in InfluxDB. The included code snippets demonstrate how to interact with the InfluxDB database to store and retrieve time series data efficiently. These examples cover basic operations such as writing data points, querying data, and handling responses. The provided code ensures optimal use of InfluxDB's capabilities for real-time data monitoring and analysis, facilitating seamless integration within the overall system architecture.

C.2.1 Section 4.3.2.2: InfluxDB Write Query Example

This subsection provides an example of how to perform a write query in InfluxDB (mentioned in Section 4.3.2.2). The Algorithm C.3 demonstrates how to structure and send data to an InfluxDB database, including setting measurement names, tags, fields, and timestamps. This example is essential for understanding how to store sensor data efficiently in a time series database.

Listing C.3: Example of structure to send data to InfluxDB database.

```

1 msg.payload = {
2     measurement: "sensor_data",

```

```

3     tags: {
4         sensor_id: msg.sensor_id,
5         location: msg.location
6     },
7     fields: {
8         temperature: msg.temperature,
9         humidity: msg.humidity,
10        infrared_intensity: msg.infrared_intensity
11    },
12    timestamp: new Date(msg.timestamp).getTime()
13 };
14 return msg;

```

C.2.2 Section 4.3.2.2: InfluxDB Read Query Example

This subsection provides an example of how to perform a read query using the InfluxDB API (mentioned in Section 4.3.2.2). The Algorithm C.4 demonstrates how to construct a query to retrieve data from an InfluxDB database based on specified fields, tags, and time range. This example is crucial for understanding how to efficiently access and analyze time series data stored in InfluxDB.

Listing C.4: Example of read query for InfluxDB API.

```

1 def read_data(database, fields, tags, start_time, end_time):
2     tag_string = ' AND '.join([f'{key}={value}' for key, value in tags.items()])
3     query = f"SELECT * FROM {database}.{fields} WHERE {tag_string} AND time >= '{start_time}'
4           AND time <= '{end_time}'"
5     result = influxdb_client.query(query)
6     return result

```

C.3 Node-RED flows

This section contains the flows in JSON format used in the project, as represented in the figures of the blocks connected in Node-RED presented in the main body of the thesis. Each section of this section corresponds to a specific figure and details the associated JSON flux.

C.3.1 Figure 4.3: Node-RED flow for processing data from the TTN MQTT topic.

Figure 4.3 in the main body of the thesis presents the read flow of the TTN MQTT topic. The flux in JSON format is shown below in Listing C.5:

Listing C.5: JSON flux for read the TTN MQTT topic

```

1 [{"id": "18144534214b7c14", "type": "debug", "z": "e4f8d7bb4458fcde", "name": "debug TTN MQTT", "
  active": true, "tosidebar": true, "console": false, "tostatus": false, "complete": "payload", "
  targetType": "msg", "x": 270, "y": 340, "wires": []}]
2 [{"id": "a20bce7660ed9b80", "type": "mqtt in", "z": "e4f8d7bb4458fcde", "name": "TTN MQTT topic", "
  topic": "#", "qos": 2, "datatype": "json", "broker": "75fdd34ea32724b5", "nl": false, "rap":
  false, "inputs": 0, "x": 260, "y": 380, "wires": [[{"id": "18144534214b7c14", "x": 270, "y": 340}]]}]

```

```

3  { "id": "49cd3ceb5c493740", "type": "function", "z": "e4f8d7bb4458fcde", "name": "Sensor Data
    Filter", "func": "var dev = msg.payload.end_device_ids.device_id;\npayload = msg.payload
    .uplink_message.decoded_payload;\npayload.t = msg.payload.uplink_message.received_at;\n
    msg.payload= payload;\nmsg.measurement= dev;\nreturn msg;", "outputs": 1, "noerr": 0, "
    initialize": "", "finalize": "", "libs": [], "x": 510, "y": 380, "wires": [ [ "e264c9eb957e9863", "
    d0aee899fac8c901" ] ] },
4  { "id": "e264c9eb957e9863", "type": "debug", "z": "e4f8d7bb4458fcde", "name": "debug function", "
    active": true, "tosidebar": true, "console": false, "tostatus": false, "complete": "true", "
    targetType": "full", "statusVal": "", "statusType": "auto", "x": 500, "y": 340, "wires": [ ] },
5  { "id": "d0aee899fac8c901", "type": "influxdb out", "z": "e4f8d7bb4458fcde", "influxdb": "0
    fa5ca6b552e30a0", "name": "InfluxDB VM", "measurement": "", "precision": "", "retentionPolicy":
    "", "database": "", "precisionV18FluxV20": "ms", "retentionPolicyV18Flux": "", "org": "CeDRI
    ", "bucket": "SAFe_project", "x": 730, "y": 380, "wires": [ ] }
6  ]

```

C.3.2 Figure 4.4: Node-RED flow for processing and storing signal quality data.

Figure 4.4 in the main body of the thesis presents the flow for processing and storing signal quality data. The flux in JSON format is shown below in Listing C.6:

Listing C.6: JSON flux to save signal quality data.

```

1  [ { "id": "d7753e0f12796536", "type": "function", "z": "e4f8d7bb4458fcde", "name": "Signal filter",
    "func": "var Frequency = Number(msg.payload.uplink_message.settings.frequency);\nvar
    Bandwidth = msg.payload.uplink_message.settings.data_rate.lora.bandwidth;\nvar SF =
    msg.payload.uplink_message.settings.data_rate.lora.spreading_factor;\nvar dev = msg.
    payload.end_device_ids.device_id;\nvar influx=[];\nfor (let i = 0; i < msg.payload.
    uplink_message.rx_metadata.length; i++){ \n var Channel_ID = 0;\n var GatewayPosition =
    0;\n var GatewayName = msg.payload.uplink_message.rx_metadata[i].gateway_ids.
    gateway_id;\n var RSSI = msg.payload.uplink_message.rx_metadata[i].rssi;\n var SNR =
    msg.payload.uplink_message.rx_metadata[i].snr;\n if (msg.payload.uplink_message.
    rx_metadata[i].channel_index != null){ \n Channel_ID = msg.payload.uplink_message.
    rx_metadata[i].channel_index;\n } \n if (msg.payload.uplink_message.rx_metadata[i].
    location != null){ \n GatewayPosition = msg.payload.uplink_message.rx_metadata[i].
    location\n } \n \n influx[i] = { \n bucket: 'SAFe_metadata', \n data: [ { \n measurement:
    GatewayName, \n tags: { DeviceName: dev }, \n fields: { \n RSSI: RSSI, \n SNR: SNR, \n
    Frequency: Frequency, \n Bandwidth: Bandwidth, \n SF: SF, \n Channel_ID: Channel_ID\n }, \n
    \n }, \n \n ]; \n \n msg.payload=0;\nmsg.payload=influx;\nreturn msg;\n", "outputs": 1, "
    noerr": 0, "initialize": "", "finalize": "", "libs": [], "x": 490, "y": 900, "wires": [ [ "26
    bab17b9aee8113", "a8c962d46b444fa1" ] ] },
2  { "id": "26bab17b9aee8113", "type": "debug", "z": "e4f8d7bb4458fcde", "name": "debug function",
    "active": true, "tosidebar": true, "console": false, "tostatus": false, "complete": "payload
    ", "targetType": "msg", "statusVal": "", "statusType": "auto", "x": 500, "y": 860, "wires"
    : [ ] },
3  { "id": "a8c962d46b444fa1", "type": "split", "z": "e4f8d7bb4458fcde", "name": "Splitting
    Gateways ID", "spl": "\n", "splType": "str", "arraySpl": 1, "arraySplType": "len", "
    stream": false, "addname": "", "x": 740, "y": 900, "wires": [ [ "d919814b7d351a63", "34
    fc32473d2226cf" ] ] },
4  { "id": "d919814b7d351a63", "type": "debug", "z": "e4f8d7bb4458fcde", "name": "debug split
    metadata", "active": true, "tosidebar": true, "console": false, "tostatus": false, "complete
    ": "payload", "targetType": "msg", "statusVal": "", "statusType": "auto", "x": 740, "y": 860, "
    wires": [ ] },
5  { "id": "34fc32473d2226cf", "type": "Stackhero-InfluxDB-v2-write", "z": "e4f8d7bb4458fcde", "
    server": "4566266067525448", "name": "InfluxDB VM", "x": 970, "y": 900, "wires": [ [ "1
    dfee4c750db1a58" ] ] },

```

```

6    {"id":"1dfee4c750db1a58","type":"debug","z":"e4f8d7bb4458fcde","name":"debug InfluxDB
    VM","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"
    payload","targetType":"msg","statusVal":"","statusType":"auto","x":990,"y":860,"
    wires":[]},
7    {"id":"4db6c31970262357","type":"debug","z":"e4f8d7bb4458fcde","name":"debug TTN MQTT ",
    "active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload
    ","targetType":"msg","statusVal":"","statusType":"auto","x":270,"y":860,"wires"
    :[]},
8    {"id":"bfeb81f96ca0fe2e","type":"mqtt in","z":"e4f8d7bb4458fcde","name":"TTN MQTT topic
    ","topic":"#", "qos":"2","datatype":"json","broker":"75fdd34ea32724b5","nl":false,"
    rap":false,"inputs":0,"x":260,"y":900,"wires":[["4db6c31970262357","
    d7753e0f12796536"]]},
9    {"id":"4566266067525448","type":"Stackhero-InfluxDB-v2-Server","name":"","host":"
    localhost","port":"8086","tls":true},
10   {"id":"75fdd34ea32724b5","type":"mqtt-broker","name":"TTN_v3","broker":"eul.cloud.
    thethings.network","port":"1883","clientId":"","autoConnect":true,"usetls":false,"
    protocolVersion":"4","keepalive":"60","cleansession":true,"birthTopic":"","birthQos
    ":"0","birthPayload":"","birthMsg":{"},"closeTopic":"","closeQos":"0","closePayload"
    :":"","closeMsg":{"},"willTopic":"","willQos":"0","willPayload":"","willMsg":{"},"
    sessionExpiry":""}
11 ]

```

C.3.3 Figure 4.5: Node-RED flow to get data from TTN storage

Figure 4.5 in the main body of the thesis presents the flow to obtain data from TTN storage. The flux in JSON format is shown below in Listing C.7:

Listing C.7: JSON flow to get data from TTN storage

```

1    [{"id":"8e0df60f53f315e2","type":"json","z":"e4f8d7bb4458fcde","name":"Transf. Msg. JSON",
    "property":"payload","action":"","pretty":false,"x":310,"y":1400,"wires":[["5
    a26088b21e70603","0380ccde46561d62"]]},
2    {"id":"5a26088b21e70603","type":"debug","z":"e4f8d7bb4458fcde","name":"debug Msgs. JSON",
    "active":false,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload",
    "targetType":"msg","statusVal":"","statusType":"auto","x":310,"y":1440,"wires":[]},
3    {"id":"ee9fd19f1ab70899","type":"http request","z":"e4f8d7bb4458fcde","name":"HTTP TTN
    Request","method":"use","ret":"txt","paytoqs":"ignore","url":"","tls":"","persist":
    false,"proxy":"","authType":"bearer","sender":false,"x":550,"y":1340,"wires":[["2
    cbc418e64b43000","de1cf97218a4621a"]]},
4    {"id":"de1cf97218a4621a","type":"debug","z":"e4f8d7bb4458fcde","name":"debug TTN request",
    "active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload",
    "targetType":"msg","statusVal":"","statusType":"auto","x":550,"y":1300,"wires":[]},
5    {"id":"2cbc418e64b43000","type":"split","z":"e4f8d7bb4458fcde","name":"Split Mult. Msgs.",
    "spl": "\\n","splType":"str","arraySpl":1,"arraySplType":"len","stream":false,"
    addname":"","x":780,"y":1340,"wires":[["8353def11ce5adb3","fe40b84a1815e6f1"]]},
6    {"id":"8353def11ce5adb3","type":"debug","z":"e4f8d7bb4458fcde","name":"debug Mult. Msgs.",
    "active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload",
    "targetType":"msg","statusVal":"","statusType":"auto","x":790,"y":1300,"wires":[]},
7    {"id":"efb9ad9254a3c564","type":"function","z":"e4f8d7bb4458fcde","name":"Prep. GET query",
    "func":"var url = 'https://eul.cloud.thethings.network/api/v3/as/applications/rf95safe
    /devices/' + msg.payload + '/packages/storage/uplink_message'\nvar method = 'get'\nmsg.
    method = method\nmsg.url = url\nreturn msg;","outputs":1,"noerr":0,"initialize":"","
    finalize":"","libs":[],"x":930,"y":1240,"wires":[["8b3c23f671dc995d","329e0804b4ab0a81
    "]]},

```

```

8   {"id":"9a955ba61649d7b9","type":"function","z":"e4f8d7bb4458fcde","name":"ID filter","func
   ": "for (let i = 0; i < msg.payload.length; i++) {\n msg.payload[i] = msg.payload[i].
   _value;\n}\nreturn msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs"
   ": [], "x":500,"y":1240,"wires": [[ "76a30434b1533ab1", "3b3e263989ec4f3a" ] ] },
9   {"id":"a1bf73fb4e847657","type":"influxdb in","z":"e4f8d7bb4458fcde","influxdb":"2cec599c.
   b84df6","name":"InfluxDB VM","query":"import `influxdata/influxdb/schema` schema.
   measurements(bucket: `SAFe_project`)","rawOutput":false,"precision":"","
   retentionPolicy":"","org":"CeDRI","x":290,"y":1240,"wires": [[ "9a955ba61649d7b9", "673
   a66a8e8c20429" ] ] }, {"id":"673a66a8e8c20429","type":"debug","z":"e4f8d7bb4458fcde","name
   ": "debug Dev. List","active":true,"tosidebar":true,"console":false,"tostatus":false,"
   complete":"payload","targetType":"msg","statusVal":"","statusType":"auto","x":300,"y"
   ":1200,"wires": [] },
10  {"id":"b42446245e816ba7","type":"inject","z":"e4f8d7bb4458fcde","name":"Loop 12h","props"
   ": [{ "p":"payload" }, { "p":"topic", "vt":"str" } ], "repeat":"","crontab":"00 12 * * *", "once"
   ": false, "onceDelay":0.1, "topic":"","payloadType":"date", "x":110,"y":1240,"wires": [[
   "a1bf73fb4e847657" ] ] },
11  {"id":"76a30434b1533ab1","type":"debug","z":"e4f8d7bb4458fcde","name":"debug ID","active":
   false,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload", "
   targetType":"msg","statusVal":"","statusType":"auto","x":500,"y":1200,"wires": [] },
12  {"id":"8b3c23f671dc995d","type":"debug","z":"e4f8d7bb4458fcde","name":"debug GETquery","
   active":false,"tosidebar":true,"console":false,"tostatus":false,"complete":true,"
   targetType":"full","statusVal":"","statusType":"auto","x":930,"y":1200,"wires": [] },
13  {"id":"0380ccde46561d62","type":"function","z":"e4f8d7bb4458fcde","name":"Prep. Data", "
   func": "if (msg.payload.result && msg.payload.result.end_device_ids && msg.payload.
   result.end_device_ids.device_id != null) {\n var dev = msg.payload.result.
   end_device_ids.device_id;\n var payload = msg.payload.result.uplink_message.
   decoded_payload;\n var isoDatetime = msg.payload.result.uplink_message.received_at;\n
   var date = new Date(isoDatetime);\n var timestamp = date.getTime();\n payload.time =
   timestamp;\n msg.payload = payload;\n msg.measurement = dev;\n return msg;\n}\n", "
   outputs":1,"noerr":0,"initialize":"","finalize":"","libs": [], "x":530,"y":1400,"wires"
   ": [[ "727c731331a04b01", "c408b652c65c1836" ] ] },
14  {"id":"727c731331a04b01","type":"debug","z":"e4f8d7bb4458fcde","name":"debug Prep. Data", "
   active":false,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload", "
   targetType":"msg","statusVal":"","statusType":"auto","x":550,"y":1440,"wires": [] },
15  {"id":"54ee094044c3598e","type":"influxdb out","z":"e4f8d7bb4458fcde","influxdb":"2cec599c.
   b84df6","name":"InfluxDB VM","measurement":"","precision":"","retentionPolicy":"","
   database":"database","precisionV18FluxV20":"ms","retentionPolicyV18Flux":"","org":
   "CeDRI","bucket":"SAFe_storageTTN","x":1010,"y":1400,"wires": [] },
16  {"id":"3b3e263989ec4f3a","type":"split","z":"e4f8d7bb4458fcde","name":"Splitting Mult. Dev.
   ", "splt":"\\n", "spltType":"str", "arraySplt":1, "arraySpltType":"len", "stream":false, "
   addname":"","x":710,"y":1240,"wires": [[ "efb9ad9254a3c564", "03402ad55d2a543f" ] ] },
17  {"id":"03402ad55d2a543f","type":"debug","z":"e4f8d7bb4458fcde","name":"debug Split Dev.", "
   active":false,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload", "
   targetType":"msg","statusVal":"","statusType":"auto","x":700,"y":1200,"wires": [] },
18  {"id":"fe40b84a1815e6f1","type":"function","z":"e4f8d7bb4458fcde","name":"Null Msgs.
   filter","func": "if (msg.payload.length > 0) {\n return msg;\n}", "outputs":1,"noerr":0, "
   initialize":"","finalize":"","libs": [], "x":1020,"y":1340,"wires": [[ "8e0df60f53f315e2",
   "d85aacb71e6d6d2b" ] ] },
19  {"id":"d85aacb71e6d6d2b","type":"debug","z":"e4f8d7bb4458fcde","name":"debug Check Null", "
   active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload", "
   targetType":"msg","statusVal":"","statusType":"auto","x":1030,"y":1300,"wires": [] },
20  {"id":"c408b652c65c1836","type":"delay","z":"e4f8d7bb4458fcde","name":"","pauseType":"rate
   ", "timeout":3, "timeoutUnits":"seconds", "rate":500, "nbRateUnits":1, "rateUnits":
   "second", "randomFirst":1, "randomLast":5, "randomUnits":"seconds", "drop":false, "
   allowrate":false, "outputs":1, "x":780,"y":1400,"wires": [[ "28ba303aef7c39c2", "54
   ee094044c3598e" ] ] },

```

```

21 { "id": "28ba303aef7c39c2", "type": "debug", "z": "e4f8d7bb4458fcde", "name": "debug Buffer", "
    active": true, "tosidebar": true, "console": false, "tostatus": false, "complete": "payload", "
    targetType": "msg", "statusVal": "", "statusType": "auto", "x": 770, "y": 1440, "wires": [] },
22 { "id": "329e0804b4ab0a81", "type": "delay", "z": "e4f8d7bb4458fcde", "name": "Queue 1msg./2sec.",
    "pauseType": "rate", "timeout": "5", "timeoutUnits": "seconds", "rate": "1", "nbRateUnits": "2"
    , "rateUnits": "second", "randomFirst": "1", "randomLast": "5", "randomUnits": "seconds", "drop
    ": false, "allowrate": false, "outputs": 1, "x": 310, "y": 1300, "wires": [ [ "ee9fd19flab70899"
    ] ] },
23 { "id": "2cec599c.b84df6", "type": "influxdb", "hostname": "127.0.0.1", "port": "8086", "protocol":
    "http", "database": "SAFe_project", "name": "SAFe_project", "usetls": false, "tls": "", "
    influxdbVersion": "2.0", "url": "http://localhost:8086", "rejectUnauthorized": true }
24 ]

```

C.3.4 Figure 4.6: Node-RED flow for periodic InfluxDB backup

Figure 4.6 in the main body of the thesis presents the flow to backup the InfluxDB periodically. The flux in JSON format is shown below in Listing C.8:

Listing C.8: JSON flux for periodic InfluxDB backup

```

1 [ { "id": "90cd0b9d4a28cf15", "type": "inject", "z": "e4f8d7bb4458fcde", "name": "Loop 24h", "props"
    : [ { "p": "payload" }, { "p": "topic", "vt": "str" } ], "repeat": "", "crontab": "00 01 * * *", "once"
    : false, "onceDelay": 0.1, "topic": "", "payloadType": "date", "x": 120, "y": 1850, "wires": [ [ "0
    d2f82f1c1325222" ] ] },
2 { "id": "0d2f82f1c1325222", "type": "exec", "z": "e4f8d7bb4458fcde", "command": "influx backup /
    mnt/nfs/BackupsInflux/backup_$(date '+%Y-%m-%d_%H-%M') -t
    eyJrIjo1ZmFrZXRva2VuIiwidCI6IkZBS0VUT0tFTiIsImMiOiJNSVRVIn0=", "addpay": "", "append": ""
    , "useSpawn": "false", "timer": "", "winHide": false, "oldrc": false, "name": "BKP Query", "x"
    : 320, "y": 1850, "wires": [ [ "316046c2adb6b9b5" ], [ "477258b9c9b79ebe" ], [ "c0d4dcc4af8f0754"
    ] ] },
3 { "id": "316046c2adb6b9b5", "type": "debug", "z": "e4f8d7bb4458fcde", "name": "debug out 1 BKP", "
    active": true, "tosidebar": true, "console": false, "tostatus": false, "complete": "payload", "
    targetType": "msg", "statusVal": "", "statusType": "auto", "x": 540, "y": 1800, "wires": [] },
4 { "id": "477258b9c9b79ebe", "type": "debug", "z": "e4f8d7bb4458fcde", "name": "debug out 2 BKP", "
    active": true, "tosidebar": true, "console": false, "tostatus": false, "complete": "payload", "
    targetType": "msg", "statusVal": "", "statusType": "auto", "x": 540, "y": 1850, "wires": [] },
5 { "id": "c0d4dcc4af8f0754", "type": "debug", "z": "e4f8d7bb4458fcde", "name": "debug out 3 BKP", "
    active": true, "tosidebar": true, "console": false, "tostatus": false, "complete": "payload", "
    targetType": "msg", "statusVal": "", "statusType": "auto", "x": 540, "y": 1900, "wires": [] } ]

```

C.3.5 Figure 4.7: A GUI to assist in the identification of modules' locations

Figure 4.7 in the main body of the thesis presents the flow to backup the InfluxDB periodically. The flux in JSON format is shown below in Listing C.9:

Listing C.9: JSON flux to assist in the identification of modules' locations

```

1 [ { "id": "ecb02d10b65b7cf1", "type": "debug", "z": "79af94607124528b", "name": "debug Dev. Sel.", "
    active": true, "tosidebar": true, "console": false, "tostatus": false, "complete": "payload", "
    targetType": "msg", "statusVal": "", "statusType": "auto", "x": 400, "y": 260, "wires": [] },

```



```

2   {"id":"804da706bc7a2334","type":"ui_dropdown","z":"79af94607124528b","name":"Measurement
    List","label":"Measurement List","tooltip":"","place":"Select option","group":"85
    cff256.eb875","order":2,"width":0,"height":0,"passthru":true,"multiple":false,"options
    ":[],"payload":"","topic":"payload","topicType":"msg","className":"","x":410,"y":300,"
    wires":[["ecb02d10b65b7cf1","de0f198505815bbf"]]},
3   {"id":"41697a4ff8eedb9a","type":"ui_form","z":"79af94607124528b","name":"Manual Coordinate
    ","label":"Manual Coordinate","group":"85cff256.eb875","order":3,"width":0,"height":0,
    "options":[{"label":"LATITUDE","value":"Lat","type":"number","required":true,"rows":
    null},{label":"LONGITUDE","value":"Lon","type":"number","required":true,"rows":null
    },{"label":"ALTITUDE","value":"Alt","type":"number","required":true,"rows":null}],
    "formValue":{"Lat":"","Lon":"","Alt":""},"payload":"","submit":"Submit","cancel":"
    Cancel","topic":"payload","topicType":"msg","splitLayout":true,"className":"","x":170,
    "y":420,"wires":[["251618f03b60bbcc","dbfd69cdc59d496b"]]},
4   {"id":"de2a09272a207494","type":"change","z":"79af94607124528b","name":"msg Options","
    rules":[{"t":"set","p":"options","pt":"msg","to":"payload","tot":"msg"}],"action":"","
    property":"","from":"","to":"","reg":false,"x":150,"y":280,"wires":[["804da706bc7a2334
    "]]},
5   {"id":"0b6347cb38cb41ad","type":"influxdb in","z":"79af94607124528b","influxdb":"4
    c2ac6b0dd1b1d54","name":"","query":"import `influxdata/influxdb/schema` schema.
    measurements(bucket: `name`)", "rawOutput":false,"precision":"","retentionPolicy":"","
    "org":"CeDRI","x":610,"y":180,"wires":[["d2d93695f7c3c9de","7712cb52075f1c6c"]]},
6   {"id":"d2d93695f7c3c9de","type":"debug","z":"79af94607124528b","name":"debug Query Bucket",
    "active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","
    targetType":"msg","statusVal":"","statusType":"auto","x":600,"y":140,"wires":[]},
7   {"id":"7ec04fb0831cf821","type":"ui_button","z":"79af94607124528b","name":"Refresh
    Databases","group":"85cff256.eb875","order":4,"width":0,"height":0,"passthru":false,"
    label":"Refresh Databases","tooltip":"","color":"","bgcolor":"","className":"","icon":
    "", "payload":"","payloadType":"date","topic":"payload","topicType":"msg","x":170,"y"
    :220,"wires":[["014e3bad4a9c2c73"]]},
8   {"id":"7712cb52075f1c6c","type":"function","z":"79af94607124528b","name":"Splitting Mult.
    Dev.","func":"for (let i = 0; i < msg.payload.length; i++){
    \n msg.payload[i] = msg.
    payload[i]._value;\n }
    \nreturn msg;","outputs":1,"noerr":0,"initialize":"","finalize":
    "", "libs":[],"x":850,"y":180,"wires":[["de2a09272a207494","4df8378802bae470"]]},
9   {"id":"014e3bad4a9c2c73","type":"ui_dropdown","z":"79af94607124528b","name":"Database List
    ","label":"Database List","tooltip":"","place":"Select option","group":"85cff256.eb875
    ","order":1,"width":0,"height":0,"passthru":false,"multiple":false,"options":[{"label"
    : "SAFe_project","value":"SAFe_project","type":"str"}, {"label":"SAFe_cubecell","value":
    "SAFe_cubecell","type":"str"}],"payload":"","topic":"payload","topicType":"msg","
    className":"","x":160,"y":180,"wires":[["8c1d24d4dcb9a997","b850cd87a35f7f93"]]},
10  {"id":"8c1d24d4dcb9a997","type":"debug","z":"79af94607124528b","name":"debug Database
    Selected","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"
    payload","targetType":"msg","statusVal":"","statusType":"auto","x":190,"y":140,"wires"
    :[]},
11  {"id":"b850cd87a35f7f93","type":"function","z":"79af94607124528b","name":"msg Query","func
    ":"msg.query = `import `influxdata/influxdb/schema` schema.measurements(bucket: `"+
    msg.payload+"`);
    \n\nreturn msg;","outputs":1,"noerr":0,"initialize":"","finalize":
    "", "libs":[],"x":390,"y":180,"wires":[["0b6347cb38cb41ad"]]},
12  {"id":"3c2faa47ac3eadee","type":"ui_button","z":"79af94607124528b","name":"Refresh
    Measurements","group":"85cff256.eb875","order":5,"width":0,"height":0,"passthru":false
    ,"label":"Refresh Measurements","tooltip":"","color":"","bgcolor":"","className":"","
    icon":"","payload":"","payloadType":"date","topic":"payload","topicType":"msg","x"
    :430,"y":340,"wires":[["804da706bc7a2334"]]},
13  {"id":"251618f03b60bbcc","type":"debug","z":"79af94607124528b","name":"debug Coordinate",
    "active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload",
    "targetType":"msg","statusVal":"","statusType":"auto","x":170,"y":380,"wires":[]},
14  {"id":"1f1f98483678b05d","type":"influxdb out","z":"79af94607124528b","influxdb":"67d03b9f
    .608cf4","name":"InfluxDB VM","measurement":"","precision":"","retentionPolicy":"","
  
```

```

    database:"", "precisionV18FluxV20":"ms", "retentionPolicyV18Flux":"","org":"CeDRI", "
    bucket":"SAFE_positions", "x":950, "y":360, "wires":[[]],
15  {"id":"eeb9d2e259f96574", "type":"function", "z":"79af94607124528b", "name":"Prep. payload", "
    func":"msg.measurement= msg.payload[0];\n\nvar Latitude = msg.payload[1].Lat;\nvar
    Longitude = msg.payload[1].Lon;\nvar Altitude = msg.payload[1].Alt;\nvar Name = msg.
    measurement;\n\npayload={ 'Latitude':Latitude, 'Longitude':Longitude, 'Altitude':Altitude
    , 'Name': Name};\n\nmsg.payload=payload;\n\nreturn msg;\n", "outputs":1, "noerr":0, "
    initialize":"","finalize":"","libs":[], "x":720, "y":360, "wires":[["d9d2378699800b1f", "1
    f1f98483678b05d"]]},
16  {"id":"de0f198505815bbf", "type":"join", "z":"79af94607124528b", "name":"Merge Dev. Coord. ",
    "mode":"custom", "build":"array", "property":"payload", "propertyType":"msg", "key":"topic
    ", "joiner":"\n", "joinerType":"str", "accumulate":false, "timeout":"","count":"","
    reduceRight":false, "reduceExp":"","reduceInit":"","reduceInitType":"","reduceFixup":"","
    , "x":730, "y":300, "wires":[["33b56daed1032a90", "eeb9d2e259f96574"]]},
17  {"id":"dbfd69cdc59d496b", "type":"function", "z":"79af94607124528b", "name":"Trigger Null
    filter", "func":"if(msg.payload !=null){\n msg.complete=true;\n return msg;\n}", "
    outputs":1, "noerr":0, "initialize":"","finalize":"","libs":[], "x":400, "y":420, "wires"
    :[[["de0f198505815bbf"]]},
18  {"id":"33b56daed1032a90", "type":"debug", "z":"79af94607124528b", "name":"debug Coord. Store",
    "active":true, "tosidebar":true, "console":false, "tostatus":false, "complete":"payload", "
    targetType":"msg", "statusVal":"","statusType":"auto", "x":730, "y":260, "wires":[[]],
19  {"id":"4df8378802bae470", "type":"debug", "z":"79af94607124528b", "name":"debug Mult. Dev.",
    "active":true, "tosidebar":true, "console":false, "tostatus":false, "complete":"payload", "
    targetType":"msg", "statusVal":"","statusType":"auto", "x":850, "y":140, "wires":[[]],
20  {"id":"d9d2378699800b1f", "type":"debug", "z":"79af94607124528b", "name":"debug Payload", "
    active":true, "tosidebar":true, "console":false, "tostatus":false, "complete":"payload", "
    targetType":"msg", "statusVal":"","statusType":"auto", "x":720, "y":400, "wires":[[]],
21  {"id":"85cff256.eb875", "type":"ui_group", "name":"Insert Positions", "tab":"3aed6lad.d9faae",
    "order":2, "disp":true, "width":30, "collapse":false, "className":""},
22  {"id":"4c2ac6b0dd1b1d54", "type":"influxdb", "hostname":"http://localhost:8086", "port":"8086
    ", "protocol":"http", "database":"database", "name":"","usetls":false, "tls":"","
    influxdbVersion":"2.0", "url":"http://localhost:8086", "rejectUnauthorized":true},
23  {"id":"67d03b9f.608cf4", "type":"influxdb", "hostname":"http://localhost:8086", "port":"8086",
    "protocol":"http", "database":"SAFE_positions", "name":"SAFE_positions", "usetls":false,
    "tls":"","influxdbVersion":"1.8-flux", "url":"http://localhost:8086", "rejectUnauthorized
    ":true, "credentials":{}},
24  {"id":"3aed6lad.d9faae", "type":"ui_tab", "name":"Positions", "icon":"dashboard", "order":3, "
    disabled":false, "hidden":false}]

```

C.3.6 Figure 4.9: Example of Node-RED flow for real-time data processing and analysis.

Figure 4.9 in the main body of the thesis presents the flow to for real-time data processing and analysis. The flux in JSON format is shown below in Listing C.10:

Listing C.10: JSON flux for real-time data processing and analysis

```

1  [{"id":"763bb56a294e47a4", "type":"inject", "z":"dcd27188.eed89", "name":"","props":[{"p":"
    payload"}, {"p":"topic", "vt":"str"}], "repeat":"","crontab":"","once":false, "onceDelay"
    :0.1, "topic":"","payload":"","payloadType":"date", "x":810, "y":900, "wires":[["6
    b6cf378cbd95a2c"]]},
2  {"id":"efb4e4ce0d1b1c5f", "type":"debug", "z":"dcd27188.eed89", "name":"","active":true, "
    tosidebar":true, "console":false, "tostatus":false, "complete":false, "statusVal":"","
    statusType":"auto", "x":1220, "y":850, "wires":[[]],

```

```

3 { "id": "6b6cf378cbd95a2c", "type": "exec", "z": "dcd27188.eed89", "command": "python3 /home/brito
  /textnodered.py", "addpay": "", "append": "", "useSpawn": "false", "timer": "", "winHide": false
  , "oldrc": false, "name": "Execution block", "x": 1010, "y": 900, "wires": [[ "efb4e4ce0d1b1c5f"
  ], [], [] ] }

```

C.3.7 Accessing Data from InfluxDB - Bucket SAFe_project

This subsection presents the algorithm designed to access data from bucket SAFe_project in InfluxDB (mentioned in Section 4.4.3). The process of accessing InfluxDB data by Algorithm 1 is described next:

Algorithm 1 Accessing Data from InfluxDB.

```

Require: start, stop, module
Ensure: df
  token ← "e5CZFu8YYOQU64cEL2GH"           ▷ Example of parameters
  host ← "localhost"
  org ← "CeDRI"
  database ← "SAFe_project"
  function DATA_GET(start, stop, module)
    client ← InfluxDBClient(host, token, org, database)
    query_api ← client.query_api()           ▷ Querying data from InfluxDB
    query ← 'SELECT * FROM "module" WHERE time > "start" AND time < "stop"'
    data_frame_influx ← query_api.query_data_frame(query)
    client.close()
    print "Influx Accessed"
    if data_frame_influx.empty then
      print "No data"
      df ← pd.DataFrame()
    else
      df ← data_frame_influx
    end if
    return df
  end function

```

- **Configuration Parameters:** The parameters required to access the database are defined, including the authentication token, host, organization, and database name. In this example, the token is fictitious and used for illustrative purposes only.
- **Client Configuration:** The function configures the InfluxDB client using the parameters provided. The client instance is created using the specified host, token, organization, and database.
- **Query Execution:** The query API (*query_api*) is used to build and execute a query in InfluxDB. The query is defined to select all data from a specific module within a time interval (*start* and *stop*).
- **Client Closure:** After query execution, the InfluxDB client closes to free up resources.
- **Data Checking:** The data returned by the query is stored in a Pandas DataFrame. If the *data_frame_influx* is empty (no data found for the specified range and module), a No

data message is printed, and an empty DataFrame is returned. Otherwise, the data frame containing the queried data is returned.

C.3.8 Writing Data in InfluxDB - Bucket SAFe_project

This subsection presents the algorithm designed to write data in bucket `SAFe_analysis` in InfluxDB (mentioned in Section 4.4.3). The process of accessing InfluxDB data by Algorithm 2 is described next:

Algorithm 2 Writing Data to InfluxDB.

```

Require: data_to_write, module
Ensure: Data written to InfluxDB
bucket ← "SAFe_analysis"                                ▷ Example of parameters
org ← "CeDRI"
token ← "YviYvAYmrtNgC0vDTEQIQ9"
url ← "localhost"
function DATA_WRITE(data_to_write, module)              ▷ Querying data into InfluxDB
    client ← InfluxDBClient(url, token, org, timeout = 180000)
    df ← data_write
    write_api ← client.write_api(write_options = SYNCHRONOUS)
    write_api.write(bucket, df, module, df_tag_columns, df_timestamp_column)
    client.close()
    print "Data Stored"
end function

```

- **Configuration Parameters:** The parameters required to access the database are defined, including the bucket, authentication token, organization, and server URL. In this example, the token is fictitious and used for illustrative purposes only.
- **Client Configuration:** The client instance is created using the URL, token, organization, and a specified timeout with the provided parameters.
- **Write Query Execution:** The write API (`write_api`) is used to build and execute a query that inserts data into InfluxDB. The data to be written is a Pandas DataFrame. This function inserts the data into the specified bucket using the module and columns for tags (if needed) and timestamps.
- **Client Closing:** After executing the write query, the InfluxDB client is closed to free up resources.
- **Function Result:** The `data_write` function prints a "Data Stored" message to indicate that the data has been successfully stored in InfluxDB.

C.3.9 Figure 4.14: System Architecture Containerization with Docker implementation.

Figure 4.14 in the main body of the thesis presents the system architecture containerization with docker implementation. The YAML format is shown below in Listing C.11:

Listing C.11: Docker Compose Configuration for Node-RED, InfluxDB, and Grafana.

```

1  version: "3.8"
2
3  services:
4    nodered:
5      build:
6        context: .
7        dockerfile: Dockerfile
8      restart: unless-stopped
9      ports:
10     - "1880:1880"
11     volumes:
12     - ./nodered_data:/data
13     environment:
14     - NODE_RED_ENABLE_SAFE_MODE=false
15     - NODE_RED_ENABLE_PROJECTS=true
16     - NODE_RED_ENABLE_TOURS=true
17     - NODE_RED_ENABLE_HTTP_ADMIN=true
18     - NODE_RED_ENABLE_HTTP_NODE=true
19     - NODE_RED_ENABLE_HTTPS=false
20     - INFLUXDB_TOKEN=my-secret-token
21
22    influxdb:
23      image: influxdb:latest
24      restart: unless-stopped
25      ports:
26     - "8086:8086"
27     volumes:
28     - ./influxdb_data:/var/lib/influxdb
29     - ./init-influxdb.sh:/docker-entrypoint-initdb.d/init-influxdb.sh
30     environment:
31     - INFLUXDB_DB=mydb
32     - INFLUXDB_ADMIN_USER=admin
33     - INFLUXDB_ADMIN_PASSWORD=admin
34     - INFLUXDB_HTTP_AUTH_ENABLED=true
35     - DOCKER_INFLUXDB_INIT_MODE=setup
36     - DOCKER_INFLUXDB_INIT_USERNAME=admin
37     - DOCKER_INFLUXDB_INIT_PASSWORD=admin123
38     - DOCKER_INFLUXDB_INIT_ORG=CeDRI
39     - DOCKER_INFLUXDB_INIT_BUCKET=Safe_project
40     - DOCKER_INFLUXDB_INIT_RETENTION=0
41     - DOCKER_INFLUXDB_INIT_ADMIN_TOKEN=my-secret-token
42
43    grafana:
44      image: grafana/grafana:latest
45      restart: unless-stopped
46      ports:
47     - "3000:3000"
48     environment:
49     - GF_SECURITY_ADMIN_USER=admin
50     - GF_SECURITY_ADMIN_PASSWORD=admin
51     - INFLUXDB_TOKEN=my-secret-token
52     volumes:
53     - ./grafana_data:/var/lib/grafana
54     - ./provisioning:/etc/grafana/provisioning
55
56  volumes:

```

```
57  nodered_data:  
58  influxdb_data:  
59  grafana_data:
```

Appendix D

Supplementary Data Analysis

This appendix contains additional plots and visualizations referenced in the data analysis chapters. These supplementary materials include detailed graphs, such as distribution plots and time-series trends, among others. The inclusion of these plots aims to support the discussions presented in the main text while avoiding overloading the respective sections. These visualizations serve as a comprehensive reference for the reader and facilitate further understanding of the analyzed data's patterns, trends, and characteristics.

D.1 Trends Analysis for Dataset 1

This section extends the trends analysis presented in the main text for Dataset 1 (Section 5.4.1), focusing on the variables `Humidity` and `IR1-IR5`. In the main body of the text, the Temperature trends (Figure 5.19) over 6-hour, 12-hour, and 24-hour intervals were analyzed using the `STL`. Here, similar analyses are conducted for the additional variables to explore their seasonal components, trends, and residuals over the same time intervals.

The Humidity trends in Dataset 1 are exhibit in Figure D.1. The 6-hour trend captures detailed, short-term fluctuations in humidity levels, highlighting the immediate response of the environment to shifts in temperature and other external conditions. The 12-hour trend provides a balanced view of daytime and nighttime humidity cycles, reducing the impact of short-lived variations and revealing broader, sustained changes. This trend reflects the typical cycle of lower humidity during the day when temperatures rise, followed by an increase in humidity at night. The 24-hour trend smooths out shorter fluctuations entirely, capturing the general humidity pattern across full days. This view helps in identifying any overall trends in humidity levels, offering a baseline against which deviations, such as those caused by specific events, can be compared.

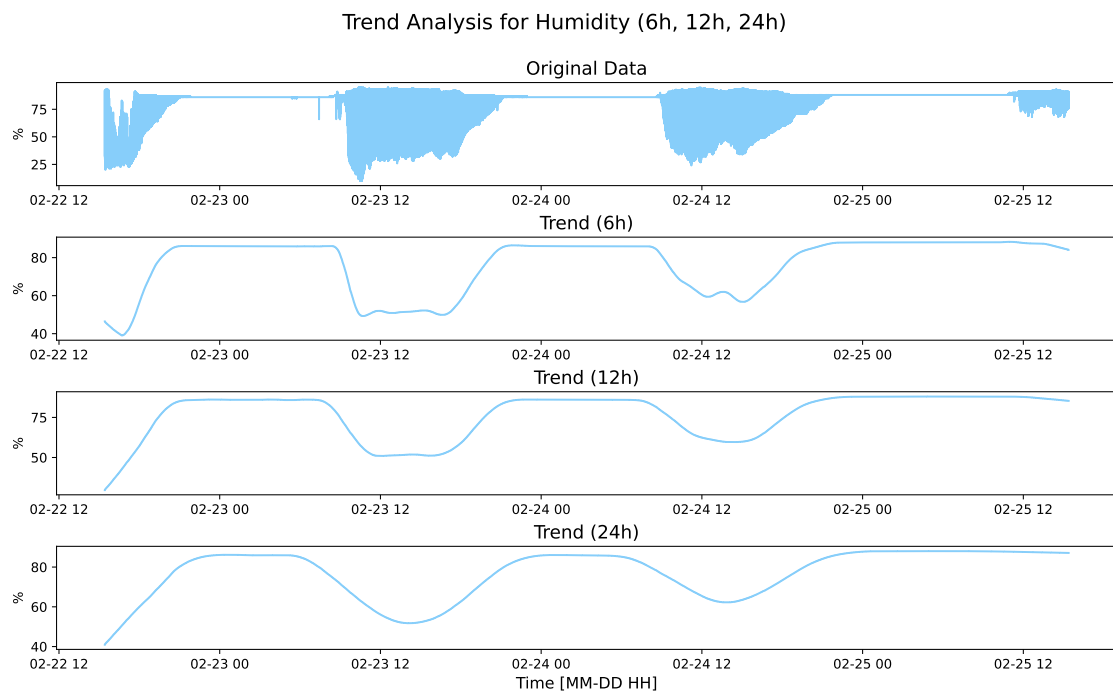


Figure D.1: Humidity trends in Dataset 1 over 6-hour, 12-hour, and 24-hour.

The trend analysis for IR1 sensor readings, as presented in Figure D.2, indicates the data across 6-hour, 12-hour, and 24-hour intervals. The 6-hour trend offers a detailed view of rapid changes in the IR1 readings, which are highly sensitive to sudden environmental shifts, including brief fire events or light variations. Moving to the 12-hour trend, the analysis reveals more consistent variations between day and night, removing some of the brief fluctuations while still highlighting persistent changes. The 24-hour trend smooths the data further, capturing the general pattern of IR1 sensor activity over complete days. This longer trend provides a baseline view, illustrating the daily IR1 levels and supporting the detection of any sustained changes that deviate from typical day-to-day behavior.

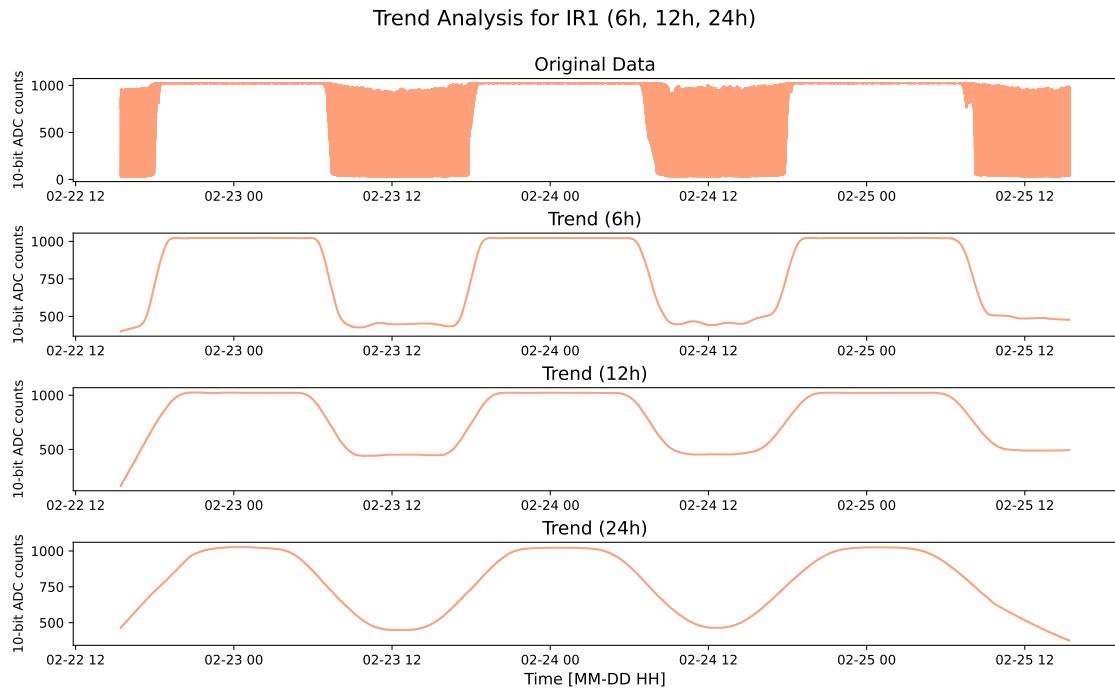


Figure D.2: IR1 sensor trends in Dataset 1 over 6-hour, 12-hour, and 24-hour.

The trend analysis of IR2, displayed in Figure D.3, emphasize the sensor's behavior across three periods. In the 6-hour trend, rapid, short-term variations are clearly visible, capturing IR2's responsiveness to immediate changes in IR intensity. These fluctuations may result from quick environmental shifts or sensor interactions with nearby sources of IR. With the 12-hour trend, the data smooths out, allowing for the observation of broader day-night patterns. This intermediate interval removes brief irregularities, clarifying any recurring IR patterns over the half-day period, which are often linked to the influence of sunlight or environmental IR sources during specific times of day. The 24-hour trend of IR2 indicates a more stabilized pattern over daily cycles, presenting the overall behavior without interference from shorter-term variations. This daily trend reinforces the general IR levels for each full day, providing a baseline that aids in distinguishing persistent shifts from typical diurnal changes.

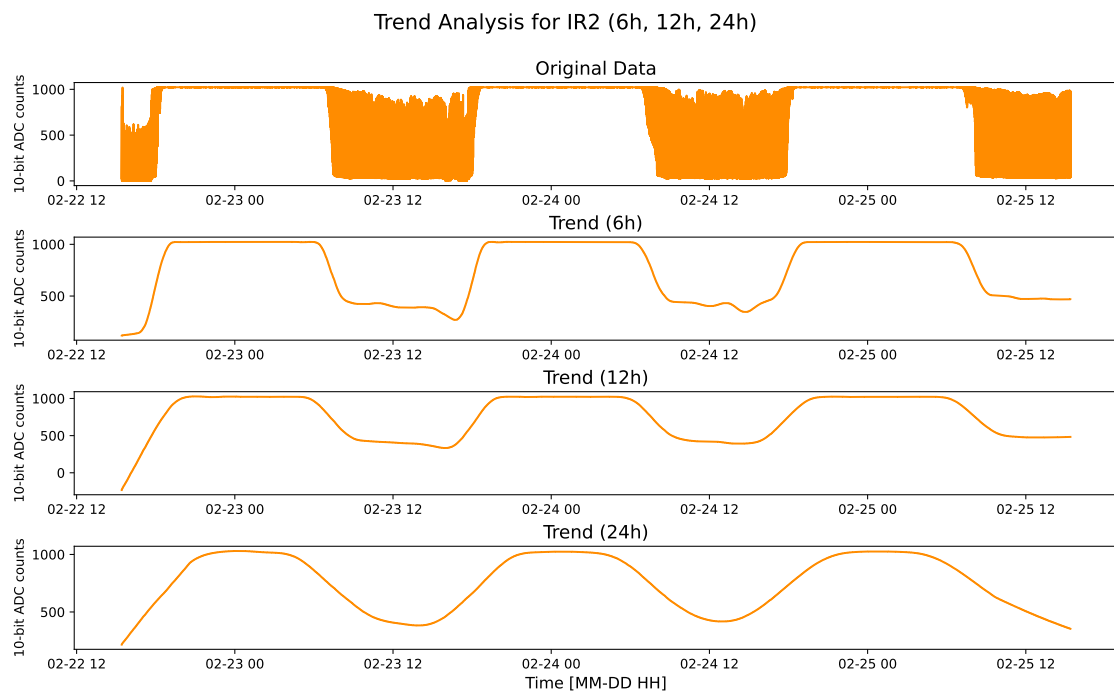


Figure D.3: IR2 sensor trends in Dataset 1 over 6-hour, 12-hour, and 24-hour.

Figure D.4 presents the trend analysis for IR3, illustrating its behavior across the 6-hour, 12-hour, and 24-hour intervals. In the 6-hour trend, short-term fluctuations are observed, highlighting IR3's responsiveness to rapid shifts in IR readings, potentially reflecting transient environmental changes or sensor interactions with nearby IR sources. The 12-hour trend plot smooths these variations, which provides a more apparent separation between day and night IR behaviors. In the 24-hour trend, IR3's daily baseline becomes prominent, offering a high-level view of IR behavior across entire days. This long-term trend captures the general diurnal IR cycle, aiding in understanding overall sensor behavior and identifying any sustained changes in IR levels across the dataset.

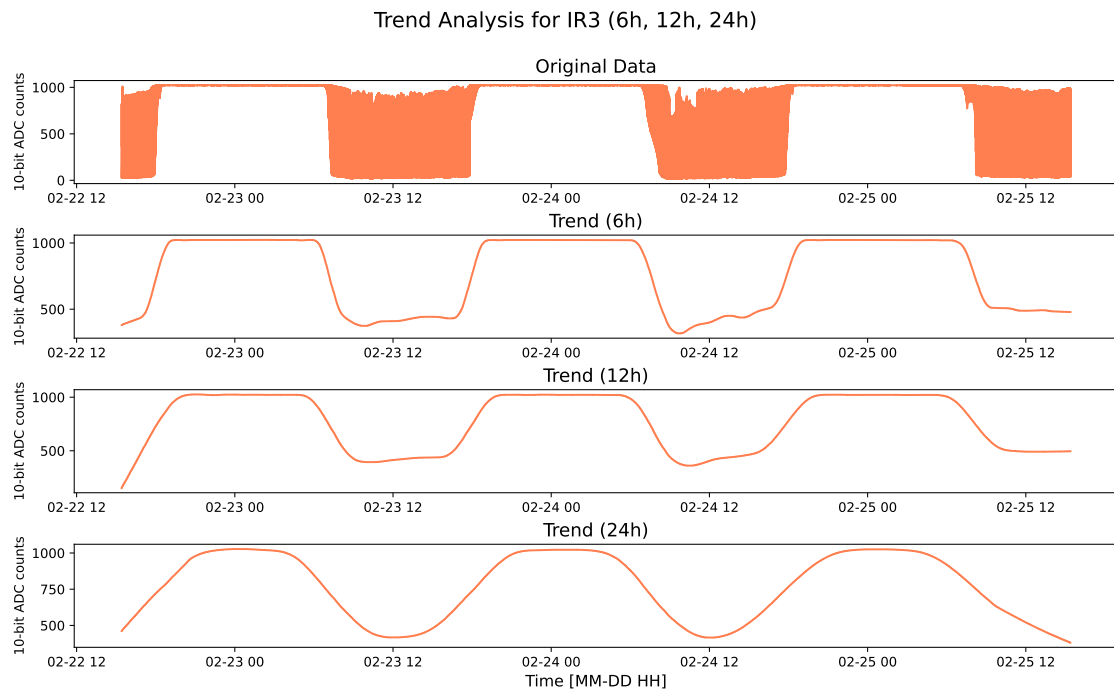


Figure D.4: IR3 sensor trends in Dataset 1 over 6-hour, 12-hour, and 24-hour.

In Figure D.5, the trend analysis for IR4 across 6-hour, 12-hour, and 24-hour intervals can be visualized. The 6-hour trend captures detailed, short-term fluctuations in IR4 readings, reflecting rapid variations due to brief environmental shifts or sensor responses to immediate **IR** sources. The 12-hour trend smooths out these short-term variations, helping to distinguish more persistent half-day patterns. This interval provides a balanced view, revealing the day-to-night differences in IR4's readings, which may be influenced by consistent **IR** sources such as sunlight. In the 24-hour trend, a broader daily pattern in IR4's readings becomes evident, providing a clear view of daily cycles. This longer trend emphasizes the baseline **IR** levels across entire days, supporting an understanding of IR4's overall behavior and highlighting any prolonged changes or irregularities across the observation period.

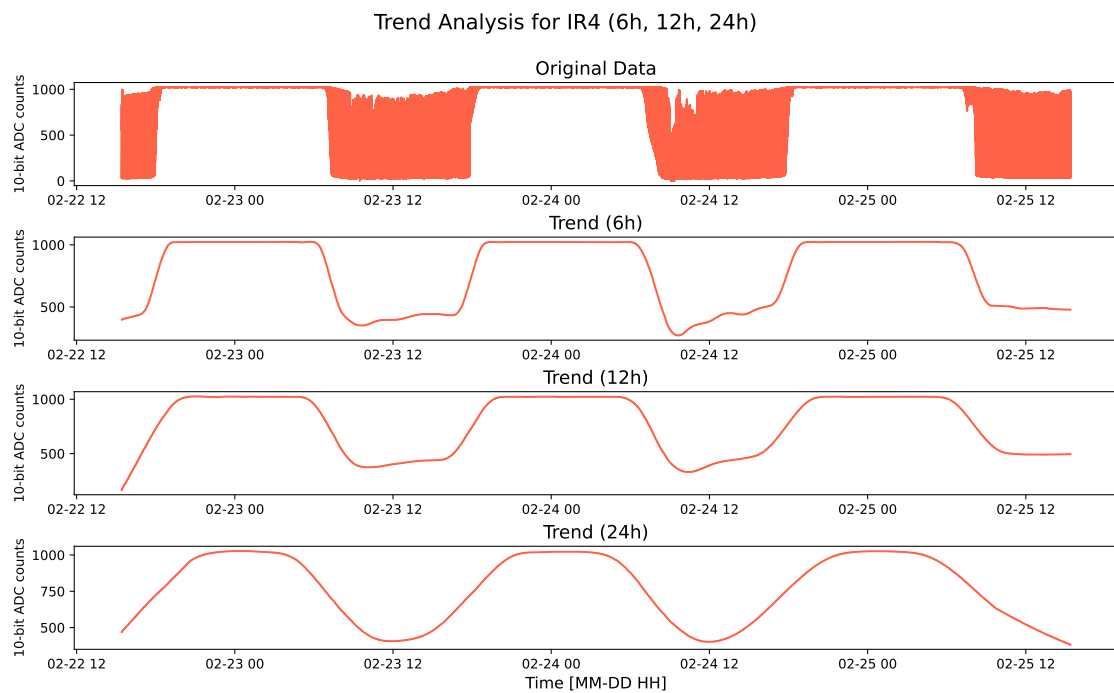


Figure D.5: IR4 sensor trends in Dataset 1 over 6-hour, 12-hour, and 24-hour.

The 6-hour, 12-hour, and 24-hour trends for IR5 data are plotted in Figure D.6. The 6-hour interval reveals short-term fluctuations, capturing immediate changes in IR5 readings that might correspond to brief environmental shifts or specific IR events. The 12-hour trend shows a smoothed view of IR5 values, which helps reveal broader day-to-night patterns. This interval provides an intermediate perspective, showing how IR levels vary between daytime and nighttime. The 24-hour trend presents the overall daily cycle in IR5 readings, highlighting consistent diurnal changes. This longer interval offers a clear view of baseline IR5 behavior across full days, indicating any sustained patterns or significant deviations throughout the monitoring period.

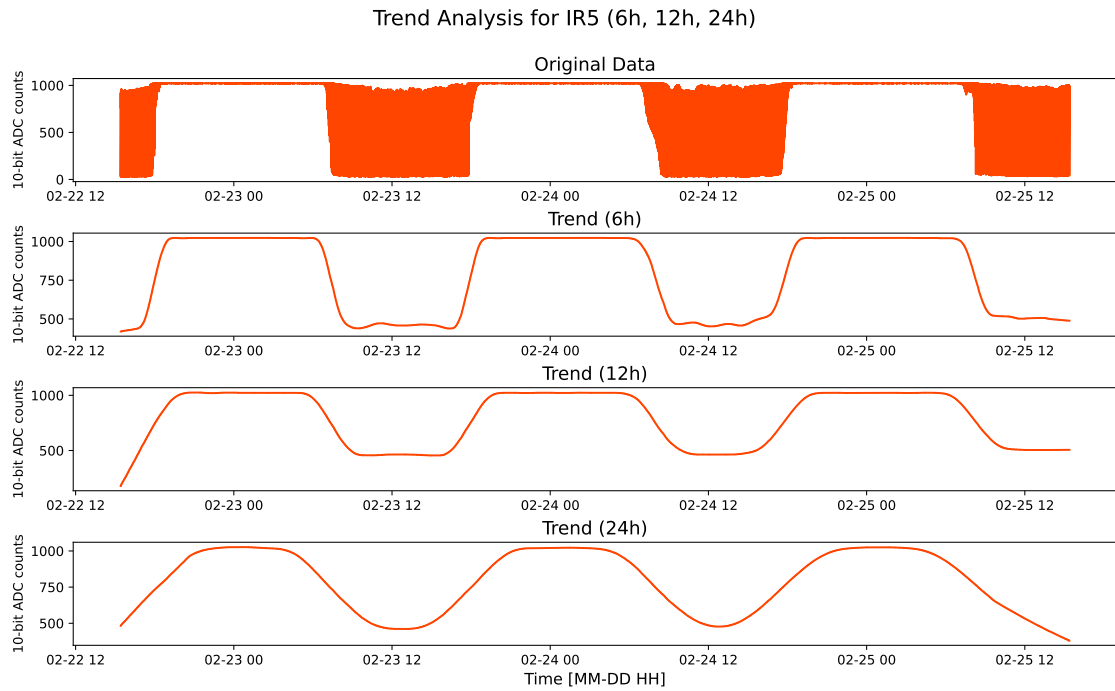


Figure D.6: IR5 trends in Dataset 1 over 6-hour, 12-hour, and 24-hour.

D.2 Trends Analysis for Dataset 2

This section extends the trends analysis presented in the main text for Dataset 2 (Section 5.4.2), focusing on the variables `Humidity` and `IR1–IR5`. In the main body of the text, the temperature trends (Figure 5.27) over 6-hour, 12-hour, and 24-hour intervals were analyzed using the **STL**. Here, similar analyses are conducted for the additional variables to explore their seasonal components, trends, and residuals over the same time intervals.

The trend analysis for `Humidity`, displayed in Figure D.7, shows the original data and 6-hour, 12-hour, and 24-hour trends. The 6-hour trend reveals granular variations, capturing short-term shifts in humidity likely influenced by immediate environmental changes, such as temperature fluctuations or weather patterns. Moving to the 12-hour trend, these variations smooth out, emphasizing broader daytime and nighttime humidity cycles. This interval reveals a balance between short-lived fluctuations and sustained changes, providing the general humidity pattern over half-day periods. The 24-hour trend shows the entire daily cycle in humidity levels. This interval highlights the overall diurnal pattern, offering a stable view of daily humidity behavior. This longer-term trend allows for identifying any sustained deviations across days, setting a baseline against which more transient changes can be measured.

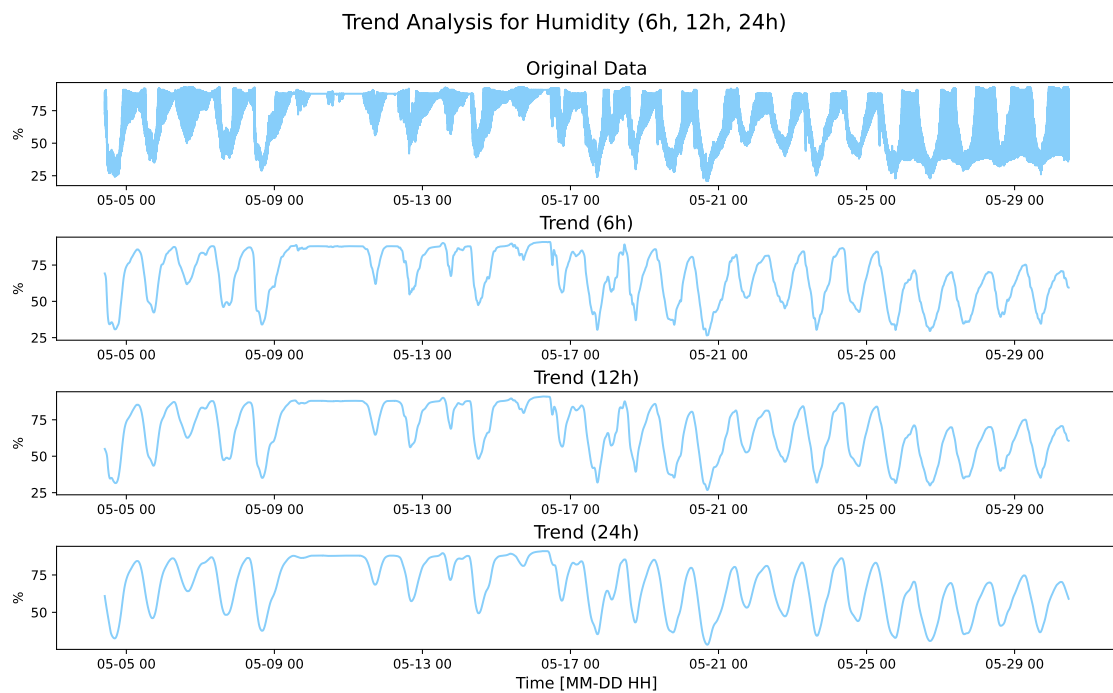


Figure D.7: Humidity trends over different time intervals (6-hour, 12-hour, and 24-hour) in Dataset 2.

The trend analysis for IR1 sensor readings, as displayed in Figure D.8, highlights the data behavior over three periods. The original data for IR1 shows pronounced fluctuations, with values consistently approaching the upper ADC count limit of 1023, indicating a frequent high response due to environmental conditions or fire events within the monitored period. The 6-hour trend captures these rapid oscillations effectively, revealing IR1's sensitivity to immediate variations in IR levels. The 12-hour trend smooths these fluctuations, focusing on a more evident pattern of daytime peaks and nighttime lows, emphasizing the sensor's periodic response to daily environmental changes. Lastly, the 24-hour trend consolidates this behavior into a stable, cyclical pattern, providing a comprehensive view of baseline IR readings for IR1.

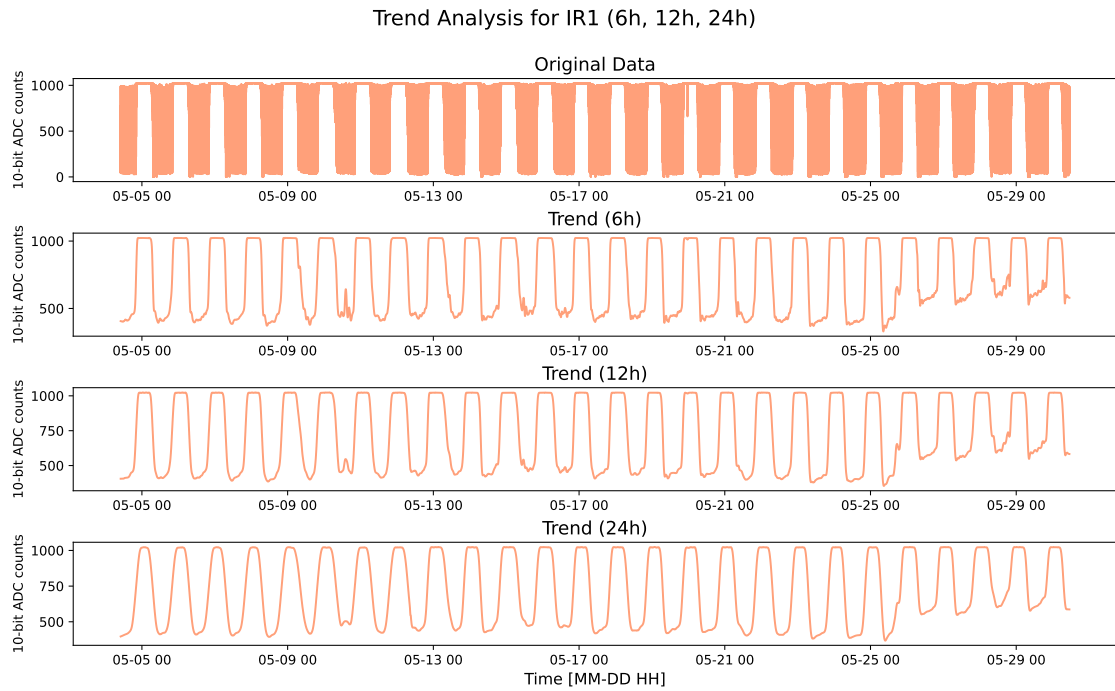


Figure D.8: IR1 sensor trends over different time intervals (6-hour, 12-hour, and 24-hour) in Dataset 2.

The trend analysis of the IR2 sensor data, shown in Figure D.9, highlights the behavior of IR2 across different time intervals. The original IR2 data shows rapid fluctuations, capturing the sensor's high sensitivity to variations in environmental IR levels. The 6-hour trend line smooths out some of these fluctuations while retaining the sharp changes, influenced by short-term environmental shifts or nearby IR sources. The 12-hour trend provides a more balanced view, emphasizing the sensor's general responsiveness across day and night cycles while reducing the effect of shorter, temporary fluctuations. This trend showcases recurring peaks that suggest the impact of daily cycles on IR readings. The 24-hour trend further smooths the data, emphasizing a stable, repeating pattern over full-day intervals, which helps highlight the overall baseline of IR2 sensor behavior across the testing period. This long-term trend provides a valuable reference for identifying deviations from typical IR patterns under high TCD.

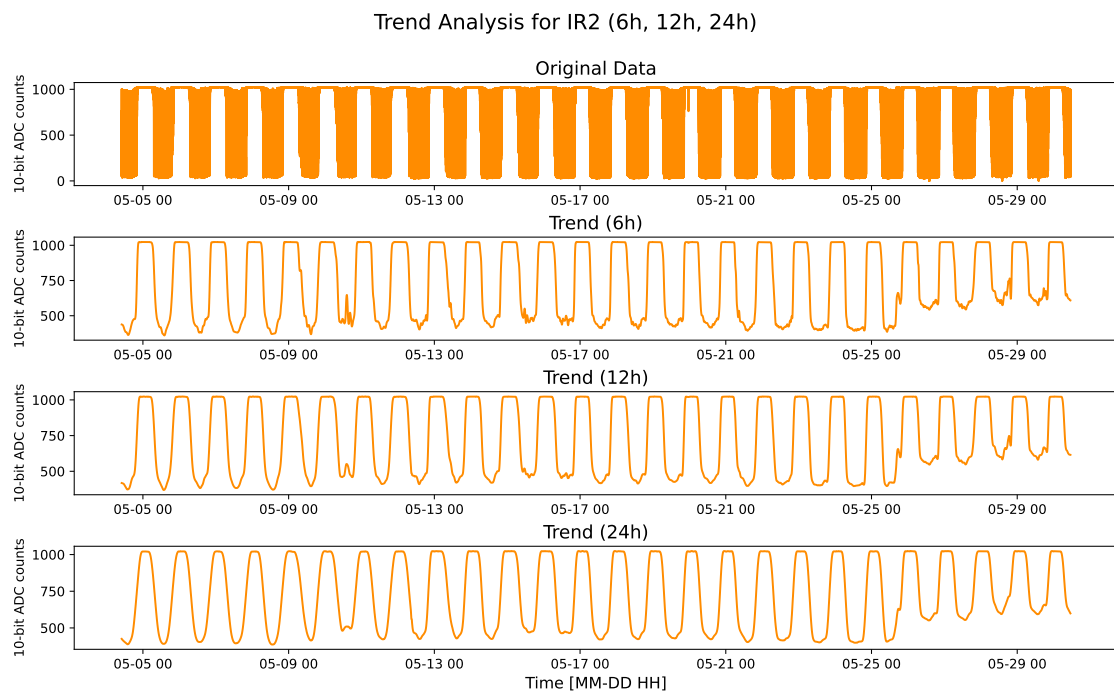


Figure D.9: IR2 sensor trends over different time intervals (6-hour, 12-hour, and 24-hour) in Dataset 2.

The trend analysis for IR3 is shown across the 6-hour, 12-hour, and 24-hour intervals, providing the short-term and longer-term patterns in sensor readings (Figure D.10). The 6-hour trend captures immediate, rapid fluctuations, highlighting IR3's sensitivity to short-lived events. These fluctuations are likely associated with brief environmental shifts or interactions with nearby IR sources, as IR3 registers high-frequency changes in ADC counts. The 12-hour trend smooths out these fluctuations, revealing more sustained daily cycles, which become evident as a shift between daytime and nighttime readings. This interval provides a balanced view of IR3 activity, emphasizing recurring patterns over half-day cycles. The 24-hour trend further simplifies the pattern, offering a high-level view of IR3's behavior across entire days. This daily trend highlights the overall diurnal cycle in the dataset and provides a more apparent baseline for evaluating sustained deviations in IR3 activity.

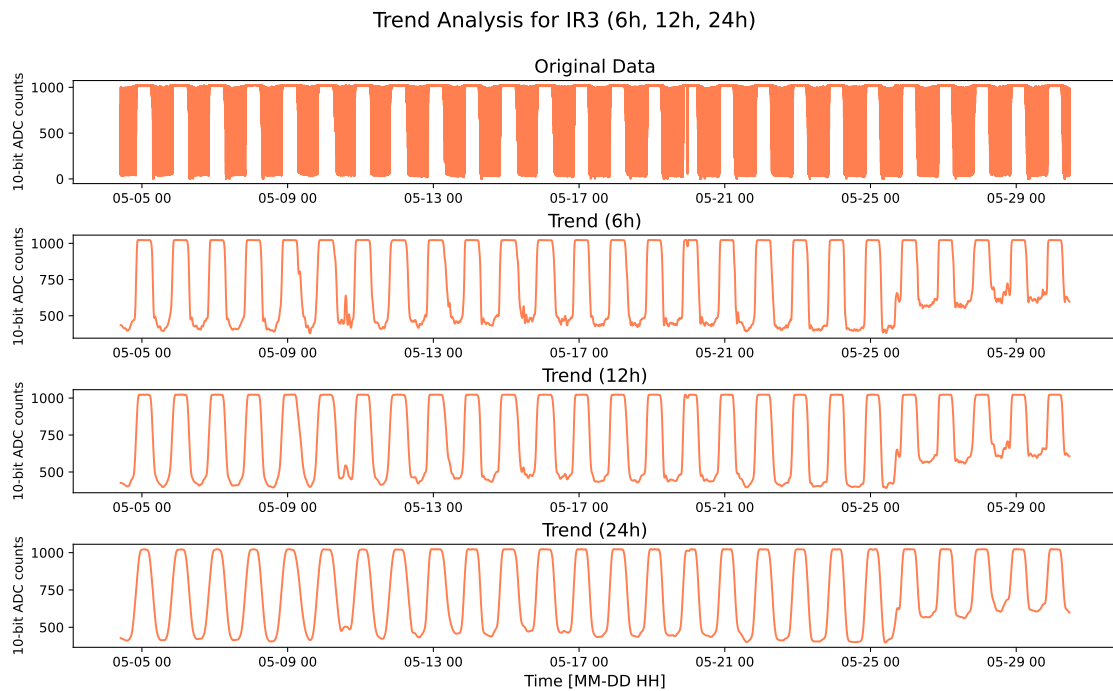


Figure D.10: IR3 sensor trends over different time intervals (6-hour, 12-hour, and 24-hour) in Dataset 2.

The IR4 sensor's trend analysis, shown in Figure D.11, presents data behavior across three temporal resolutions. The original data shows prominent values in the upper ADC range, with notable fluctuations indicative of high signal sensitivity to immediate environmental changes. In the 6-hour trend, these short-term variations are smoothed but retain the high sensitivity of the original readings, capturing frequent, moderate shifts in the IR signal. The 12-hour trend smooths the data further, highlighting an evident rising and falling trend that could correspond to daily environmental cycles or sensor interactions with nearby light sources. The 24-hour trend provides a high-level overview of IR4's daily behavior, reflecting a consistent pattern of elevated readings throughout most days.

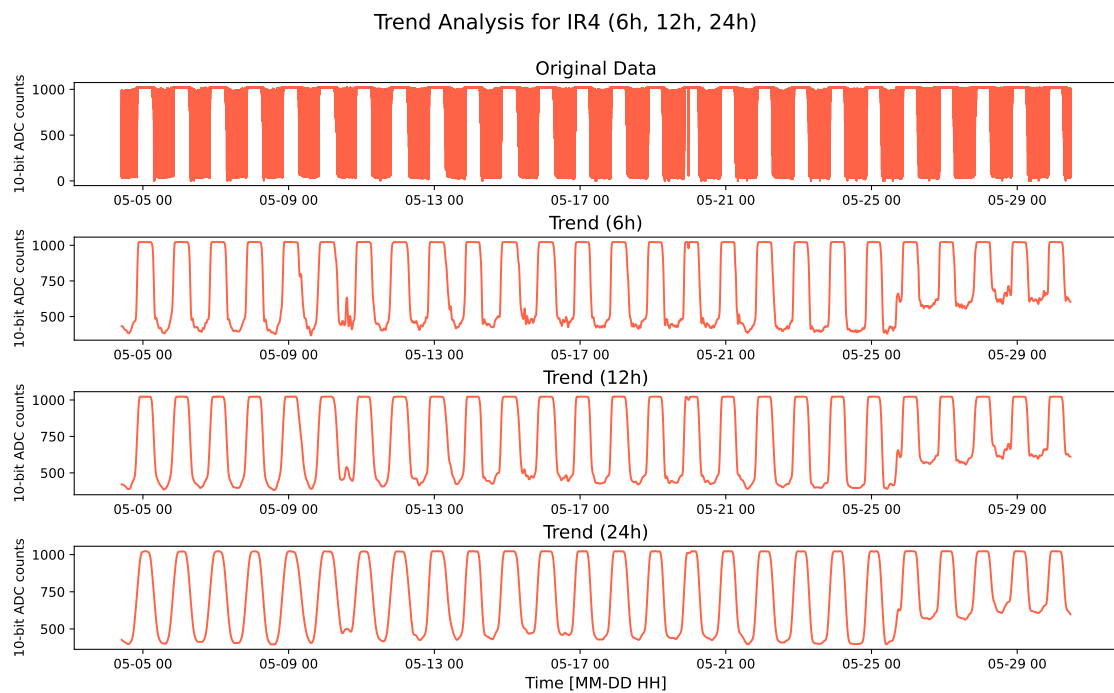


Figure D.11: IR4 sensor trends over different time intervals (6-hour, 12-hour, and 24-hour) in Dataset 2.

The trend analysis of the IR5 sensor, Figure D.12 presents the plots and reveals distinct fluctuations across multiple intervals. The 6-hour trend captures granular changes, displaying short-term variations that immediate environmental factors may influence. This interval highlights high-frequency fluctuations, possibly linked to temporary events affecting the IR5 readings. The 12-hour trend smooths out some of these rapid changes, focusing on medium-term patterns. This period emphasizes day-to-night variations, offering a clearer view of how the IR5 sensor readings shift in response to alternating light and environmental conditions. The 24-hour trend captures IR5's broader daily pattern, exposing a general baseline for each day and a notable consistency in the sensor's higher-value readings over extended periods. This view provides an overview of typical IR5 behavior across full days, helping to identify sustained patterns and potential deviations throughout the month.

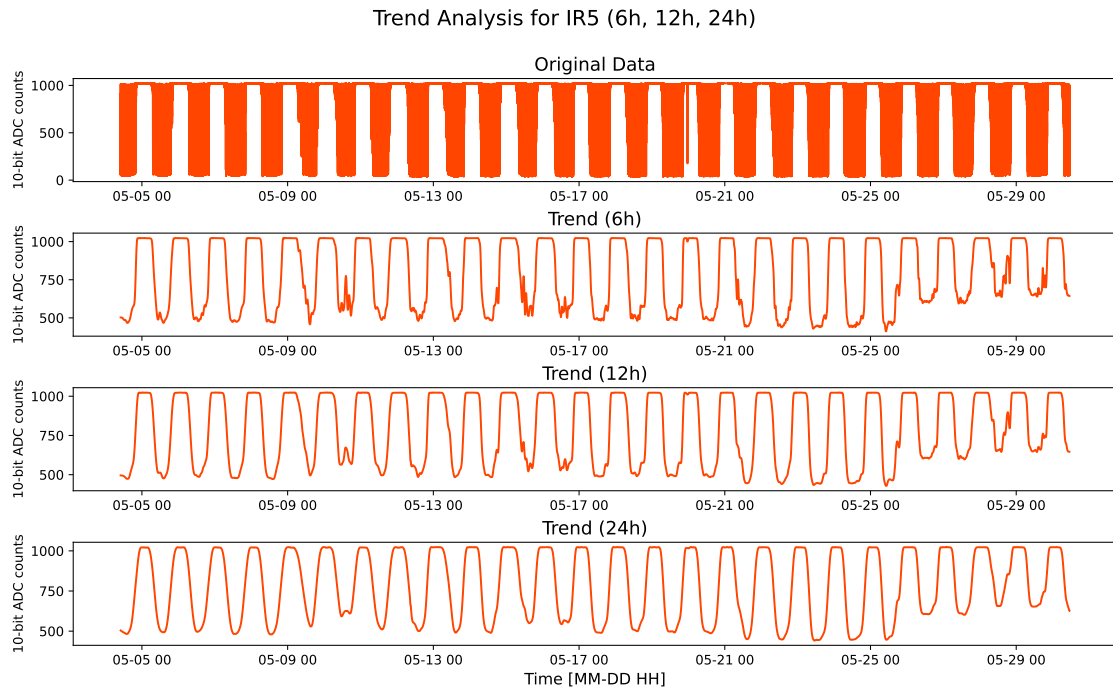


Figure D.12: IR5 sensor trends over different time intervals (6-hour, 12-hour, and 24-hour) in Dataset 2.

D.3 Linear Regression Approach

The content of this section is complementary to the methodology described in Section 5.6.1. The entire dataset was initially used for **LR** modeling to evaluate its potential for capturing patterns and predicting data behavior. However, this approach revealed significant limitations, particularly in adapting to dynamic data patterns over time.

Dataset 1 represents data collected in a low **TCD** scenario, while Dataset 2 corresponds to a high **TCD** environment. These two contrasting environmental contexts provide an opportunity to examine how sensors' data relationships differ with **TCD**. In this context, it was observed during **EDA** that **TCD** affects variables such as intensity of **IR**, temperature, and humidity. Therefore, applying a model that can interpret and quantify the direct and indirect impacts between variables within each **TCD** scenario is essential.

The choice of **LR** method in both datasets is motivated by two main observations: (1) the correlation matrix results and (2) the temporal dependencies identified in the **ACF** plots. The correlation matrices reveal strong relationships between certain groups of variables, suggesting that **LR** could effectively model the influence of one variable on another in these contexts. This is evident for variables with significant interdependence, such as the **IR** sensors (IR1 to IR5) and environmental factors such as Temperature and Humidity, indicating inter-variable correlations.

The **ACF** analysis further supports the potential for **LR**, showing consistent temporal correlations within individual variables. Although these temporal patterns indicate autocorrelation and point to

trends, they also suggest that a model based on **LR** can provide valuable outputs from each dataset. In this sense, one of the possible methods to achieve this is to create a model with the features derived from lagged values and transformations. This approach uses derivatives and transformations of the variables themselves, rather than incorporating external variables, to ensure that the model is specific to each **TCD** context while focusing on sensors' data characteristics.

Before modeling the **LR**, a function named `derive_nth_feature` was applied to each dataset. It creates new features by deriving the value of a variable from previous time steps, according to the following Equation D.1. Therefore, for each variable X_i (where i ranges from 1 to 7), it's possible to obtain lagged dataset to be used for modeling.

$$X_{i,k}[n] = \begin{cases} \text{null}, & \text{if } n \leq k \\ X_i[n-k], & \text{if } n > k \end{cases} \quad (\text{D.1})$$

Where $X_{i,k}$ represents the lagged data given a variable, n is the time index, and k denotes the lag for each feature. This lag shifts the original time series by k observations, introducing prior information for prediction tasks. Consequently, increasing k generates multiple lagged features, capturing dependencies at time steps. Figure D.13 provides a visual representation of this process using synthetic data modeled as a sinusoidal wave. From the graph, it is evident that at a given moment in time, only the original value of the data is present. However, as time progresses, it becomes possible to observe the original value along with up to four previous values, corresponding to the lagged features derived from past time steps.

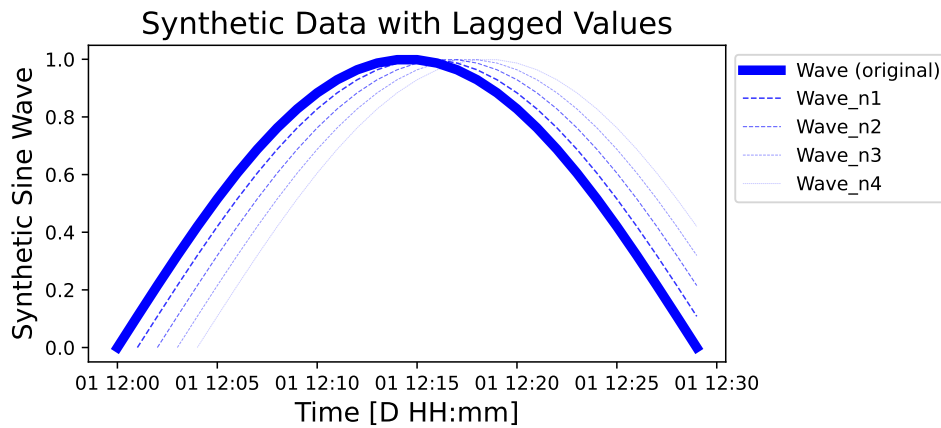


Figure D.13: Lagged feature generation using synthetic sinusoidal data, showing how past values are incorporated over time

D.3.1 LR models for WSN in Low TCD

The initial step in applying **LR** in Dataset 1 is defining the best k value to derivate each variable. Therefore, according to the **EDA** (**ACF** and **PACF** plots in Figure 5.20), the optimal lag ranges from 4 to 6 for the Temperature variable and 10 to 15 for other variables, including Humidity, and IR1 to

IR5. Based on this, the maximum k values were chosen as 15 to build all **LR** models. In this sense, the **LR** model setting was:

- **Feature Derivation:** A maximum lag $k = 15$
- **Data Splitting:** The dataset was divided into training and testing sets using an 80 – 20 split, in which X_{train} and X_{test} mean derived features for each variable, and Y_{train} and Y_{test} mean the original data values of each variable.
- **Metrics for Evaluation:** The models were evaluated using the following metrics: R^2 Score, **MAE**, **MedAE**, Explained Variance, **MSE**, **RMSE**.

Table D.1 summarizes the results of the **LR** models applied to Dataset 1 for each sensor variable. Each variable was modeled using a unique k value to maximize performance. Notably, the Temperature variable achieved the highest R^2 score, indicating a solid predictive performance. In contrast, other variables, showed relatively lower R^2 values, reflecting challenges in modeling with simple **LR**.

Table D.1: Best LR Parameters and Performance Metrics for Each Variable in Dataset 1.

Variable	k	R^2	MAE	MedAE	Explained Variance	MSE	RMSE
Temperature	7	0.978592	0.511001	0.248592	0.978594	1.065944	1.032446
Humidity	12	0.795504	3.953763	0.235637	0.79572	72.31661	8.503917
IR1	12	0.71913	116.2526	7.043835	0.719152	45314.54	212.8721
IR2	12	0.771276	103.8542	5.744636	0.771289	38257.59	195.5955
IR3	12	0.735929	112.7256	6.725218	0.735932	42913.28	207.1552
IR4	12	0.741603	111.2652	6.231573	0.741608	42055.26	205.0738
IR5	12	0.712031	114.1656	6.931536	0.712034	44993.81	212.1174

To explore the potential of **LR** in identifying anomalies, the model was trained and evaluated using data before the **WSE**. Specifically, the selected timeframe was from February 22, 2021, 15h25 to February 23, 2021, 9h14. In this sense, this pre-fire period was used to establish baseline behavior for each variable. The metrics of this process are in Table D.2:

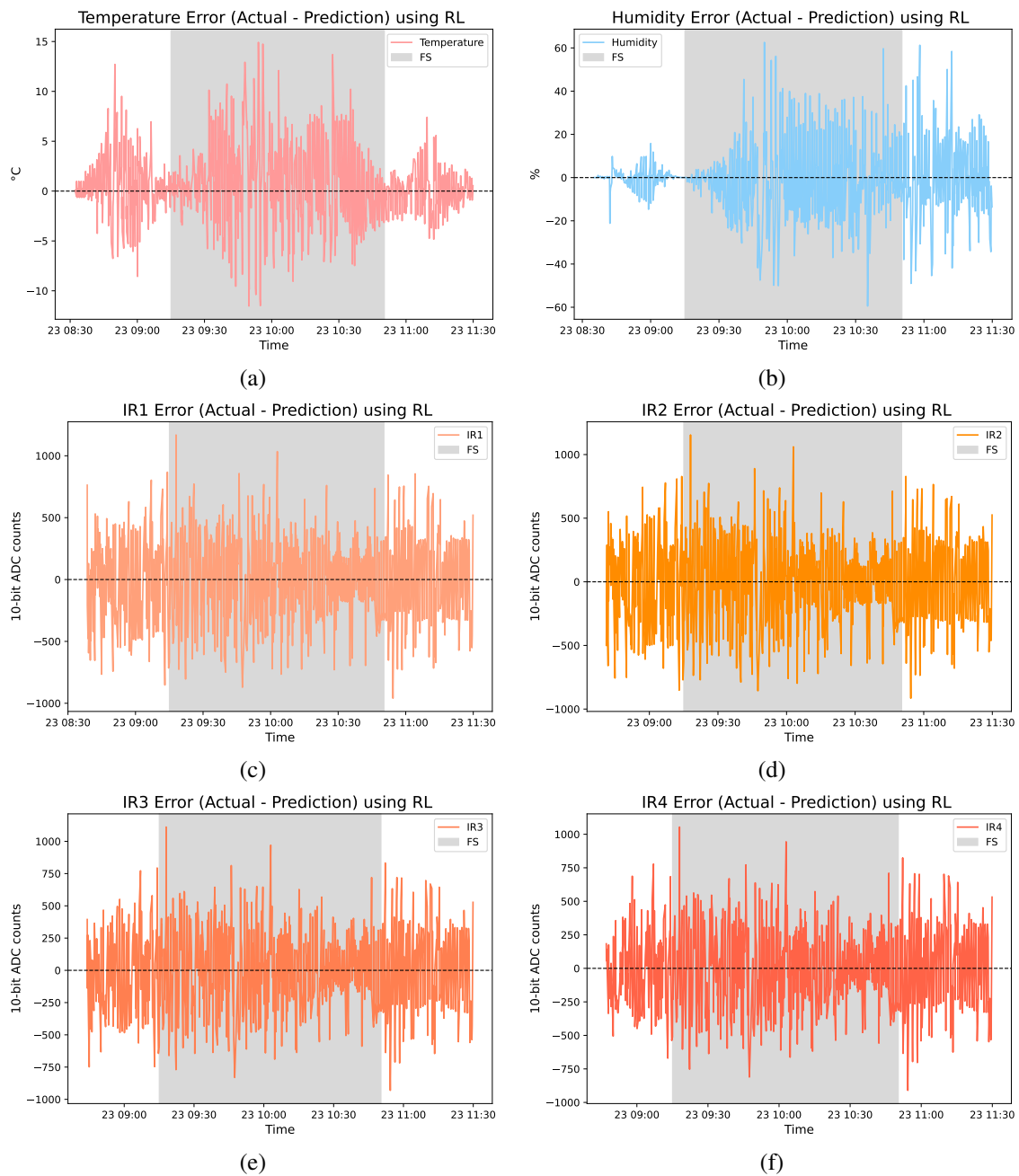
Table D.2: LR Model Performance Metrics for Pre-Fire Period in Dataset 1.

Variable	k	R^2	MAE	MedAE	Explained Variance	MSE	RMSE
Temperature	7	0.975414	0.443017	0.300745	0.975463	0.704309	0.839231
Humidity	12	0.85342	2.596632	0.020794	0.853423	46.33307	6.80684
IR1	12	0.723731	79.72409	3.551127	0.723812	31027.74	176.1469
IR2	12	0.852222	58.27614	1.640466	0.852351	20770.7	144.1205
IR3	12	0.726087	79.6005	3.52763	0.726134	31252.45	176.7836
IR4	12	0.725162	79.80903	3.286693	0.725206	31218.72	176.6882
IR5	12	0.699046	77.63869	3.26942	0.699066	31210.58	176.6652

The R^2 scores indicate different levels of predictive strength across the variables, with Temperature and Humidity achieving relatively high R^2 values of 0.975 and 0.853, respectively. Similarly,

IR2 shows a R^2 of 0.852, suggesting that the model effectively captures the variable's underlying patterns.

After modeling each variable, the data from February 23, 2021, 8h30 to February 23, 2021, 11h30 was used to forecast the values. In this sense, the Error (the difference between the predicted data from each variable model and their original data) was verified to evaluate the possibility of detecting anomalies using this approach. Figure D.14 provides a graphical representation of the prediction errors for each variable. This visualization highlights the differences between the actual data and the predicted values during the pre- and post-WSE periods.



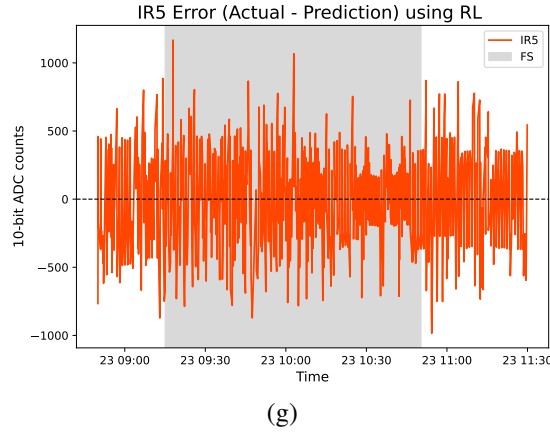


Figure D.14: Graphical representation of LR Model's prediction errors for each variable in Dataset 1. The plots display the error values for Temperature (a), Humidity (b), and IR1 through IR5 (c)-(g).

As Temperature has the highest R^2 score among all variables, it was expected to have minimal errors when comparing predicted and actual data. However, despite this favorable metric, the graphical analysis of the error did not reveal a clear anomaly pattern or a discernible threshold indicative of the **WSE**. This indicates that, even with a robust model fit, the regression approach struggled to capture sudden and significant environmental shifts, due to the dense volume of data collected over a short period.

Additionally, this behavior was consistent across the other variables, where similar error patterns were observed, lacking a clear indication of the fire-induced anomalies. It highlights the potential need for data standardization or uniformization to improve the quality of future forecasts, making the model more effective in detecting abrupt changes related to fire events.

D.3.2 LR models for WSN in High TCD

The same analysis performed on Dataset 1 in the low **TCD** scenario is extended to Dataset 2, which presents a high **TCD** environment. Using the information from the **EDA**, as well as the **ACF** and **PACF** plots in Figure 5.28, it was determined that the optimal lags for this dataset range from 10 to 20. To thoroughly explore the dependencies and relationships within this dataset, a maximum lag value of $k = 20$ was selected for modeling. Consequently, **LR** models were generated for each variable, iterating through all k values from 1 to 20.

The configurations for data splitting and the metrics used for evaluation were the same as the analysis conducted in the low **TCD** scenario. Specifically, the dataset was split into training and testing sets using an 80 – 20 split, and the models were assessed using R^2 Score, **MAE**, **MedAE**, Explained Variance, **MSE**, and **RMSE**. Table D.3 presents the performance metrics for the best **LR** models of each variable, indicating the effectiveness of this approach in this high **TCD** context.

Table D.3: Best LR Parameters and Performance Metrics for Each Variable in Dataset 2.

Variable	k	R^2	MAE	MedAE	Explained Variance	MSE	RMSE
Temperature	14	0.990238	0.328914	0.236667	0.990238	0.245686	0.495667
Humidity	18	0.876193	4.481848	2.948817	0.876193	45.99187	6.781731
IR1	15	0.522525	190.2228	108.2472	0.522534	79270.32	281.5498
IR2	15	0.517017	191.0701	108.1456	0.517026	80046.54	282.925
IR3	15	0.511115	195.892	118.524	0.511124	82858.29	287.8512
IR4	15	0.51381	195.8895	118.8162	0.513819	82886.21	287.8997
IR5	15	0.484379	174.4217	66.96699	0.484381	71281.61	266.9862

Similar to the performed before, to evaluate the potential of LR models in detecting environmental changes and anomalies related to the WSE1 in Dataset 2, a modeling test was conducted over a specific pre-fire timeframe. This test used data from May 4, 2021, at 9h00 to May 12, 2021, at 14h59. Then, it is possible to check any baseline model and assess the predictive metrics of each variable. The results of this test, with the best k values and performance metrics for each variable, are presented in Table D.4.

Table D.4: LR Model Performance Metrics for Pre-Fire (first WSE) Period in Dataset 2.

Variable	k	R^2	MAE	MedAE	Explained Variance	MSE	RMSE
Temperature	14	0.992503	0.24918	0.140502	0.992503	0.162386	0.402972
Humidity	18	0.941156	2.519419	1.392355	0.941156	17.71727	4.209189
IR1	15	0.674845	144.6699	50.98294	0.674878	54559.79	233.5804
IR2	15	0.669089	143.7815	47.01508	0.669129	54869.76	234.243
IR3	15	0.668829	146.9069	50.71287	0.668849	56208.1	237.0825
IR4	15	0.673587	145.8431	50.51317	0.673607	55508.66	235.6028
IR5	15	0.626472	136.8785	34.88773	0.626575	52084.44	228.2201

The metrics have substantial variations in model performance across different variables. The Temperature variable has an R^2 score of 0.9925, indicating an exceptionally high predictive accuracy and minimal error values, such as an MAE of 0.2492. Similarly, Humidity exhibits high predictive performance, with an R^2 of 0.9412 and relatively low error metrics.

On the other hand, the IR sensor variables (IR1 to IR5) show moderate R^2 values, ranging from 0.6265 to 0.6748, reflecting the challenges of capturing more complex patterns in the data under high TCD. These sensors also present higher MAE and RMSE values, indicating more substantial prediction errors. The graphical representation of the prediction errors for each variable in Dataset 2, showcasing the difference between the predicted and actual values, is provided in Figure D.15.

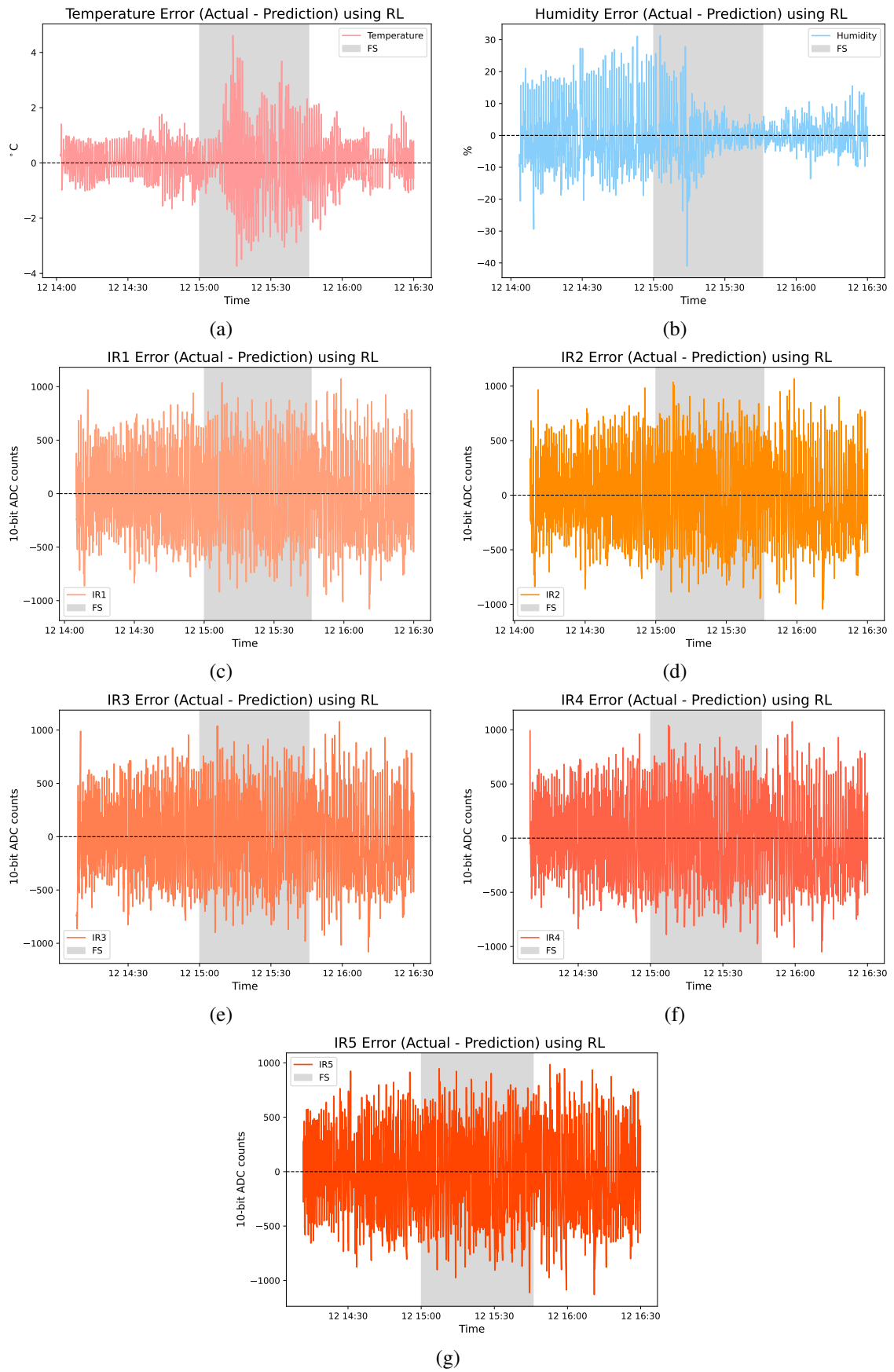


Figure D.15: Graphical Representation of RL Model's prediction errors for each variable in Dataset 2. The plots display the error values for Temperature (a), Humidity (b), and IR1 through IR5 (c)-(g).

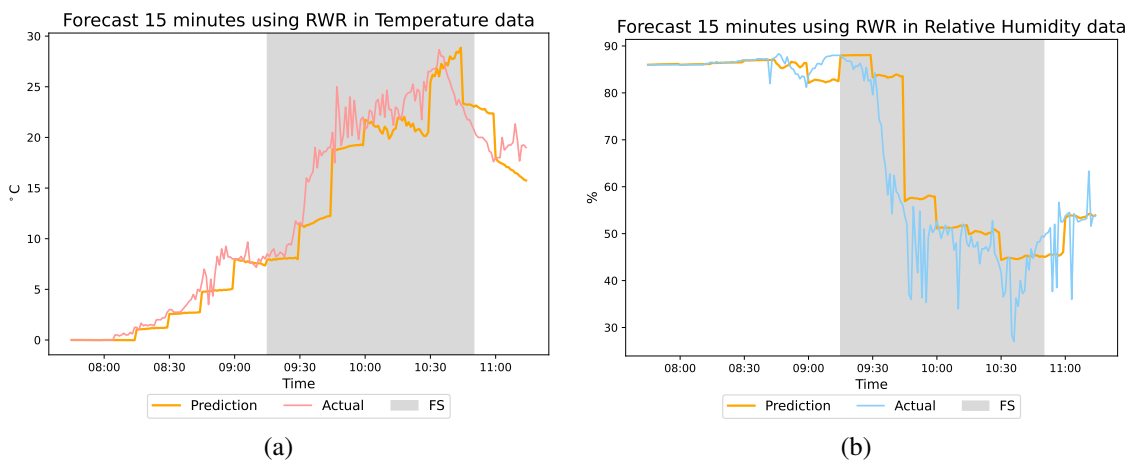
The prediction error plot for Temperature shows relatively stable errors up to the initial minutes of the **WSE**, with deviations not exceeding $\pm 2^\circ\text{C}$. Around 15h10, the errors became noticeably larger compared to previous values. This behavior, however, is not observed in the error plots for other variables, even with the high R^2 scores achieved for Humidity, the graphical analysis reveals that no distinct anomaly pattern or threshold can be identified to indicate the presence of the **WSE** clearly.

Even with robust model performance, the **LR** approach fails to capture the abrupt environmental shifts during fire events. Additional tests conducted on data from subsequent **WSEs** also presented similar results, with no discernible anomalous behavior observed during the fire events. One possible explanation for this discrepancy lies in the dataset's uniformization and high temporal resolution. The model may be overly sensitive to minor fluctuations and noise in the high-frequency data, leading to high prediction accuracy on average but failing to capture meaningful deviations during critical events, such as fire occurrences.

D.4 Visual Comparison of the RWR Predicted Values and Actual Sensor Readings for Dataset 1

This section extends the **RWR** analysis presented in the main text for Dataset 1 (Section 5.6.1.1). In the main body of the text, the **RWR** method was applied and discussed in terms of its performance metrics. Here, supplementary analyses focus on a visual comparison between the predicted and actual sensor readings.

The graphical plots in Figure D.16 aim to provide an intuitive assessment of the **RWR** model's ability to capture patterns and predict sensor values under the low **TCD** scenario. Highlighted **WSE** periods, as indicated by the **FS** column, mark the moments of fire events, offering critical context for evaluating the model's behavior during these high-variability intervals. These visualizations reinforce the observations from the main text and underscore the model's potential utility for fire anomaly detection.



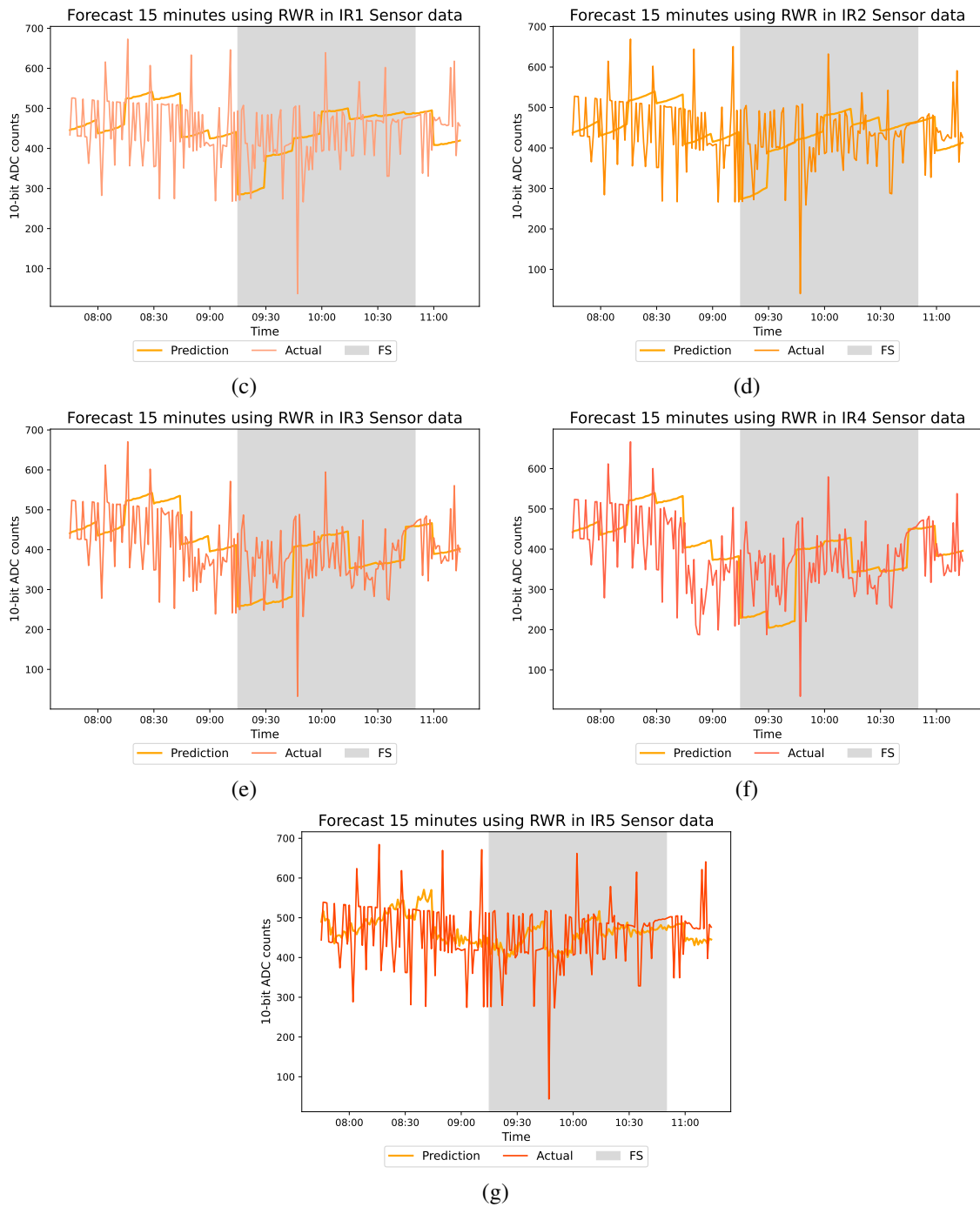


Figure D.16: Graphical representation of RWR Model's Prediction variables' values in Dataset 1. The plots display the actual versus predicted values for Temperature (a), Humidity (b), and IR1 through IR5 (c)-(g).

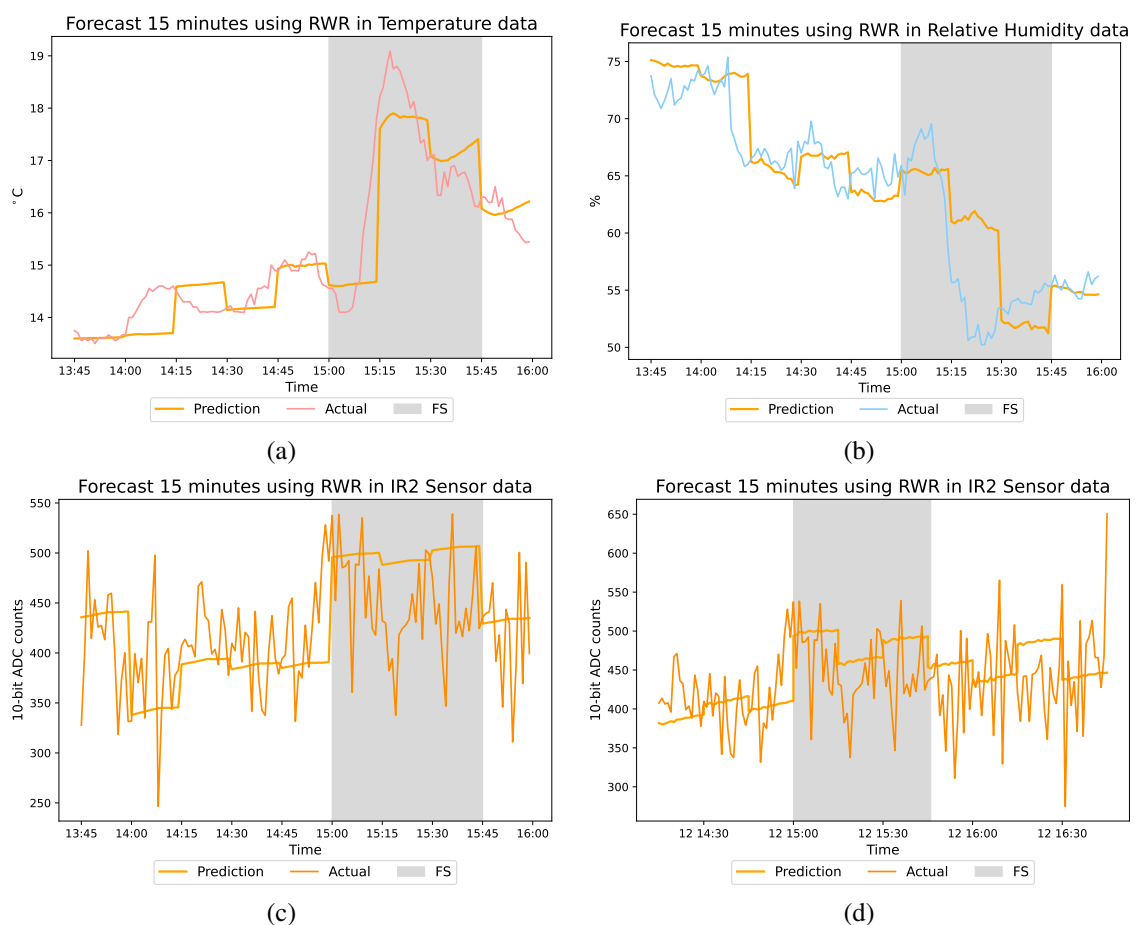
The Temperature plot shows actual and predicted values closely aligned initially. However, after the iteration at 9h30, a noticeable divergence emerges, which could indicate the onset of fire. Similarly, in the Humidity plot, the difference between actual and predicted values increases significantly around the same time, reinforcing the possibility of fire detection through these

variations. For the **IR** sensor plots, the forecast attempts to follow the actual values but tends to approximate the mean of the peaks formed by the actual data. This behavior is consistent across all **IR** sensors. Notably, around 9h45, all **IR** plots exhibit a substantial divergence between actual and predicted values, marking a clear indication of fire occurrence.

D.5 Visual Comparison of the RWR Predicted Values and Actual Sensor Readings for Dataset 2

This section extends the **RWR** analysis presented in the main text for Dataset 2 (Section 5.6.1.2). In the main body of the text, the **RWR** method was applied and discussed in terms of its performance metrics. Here, supplementary analyses focus on a visual comparison between the predicted and actual sensor readings.

The graphical plots in Figure D.17 aim to provide an intuitive assessment of the **RWR** model's ability to capture patterns and predict sensor values under the high **TCD** scenario. Highlighted **WSE** periods, as indicated by the **FS** column, mark the moments of fire events, offering critical context for evaluating the model's behavior during these high-variability intervals. These visualizations reinforce the observations from the main text and underscore the model's potential utility for fire anomaly detection.



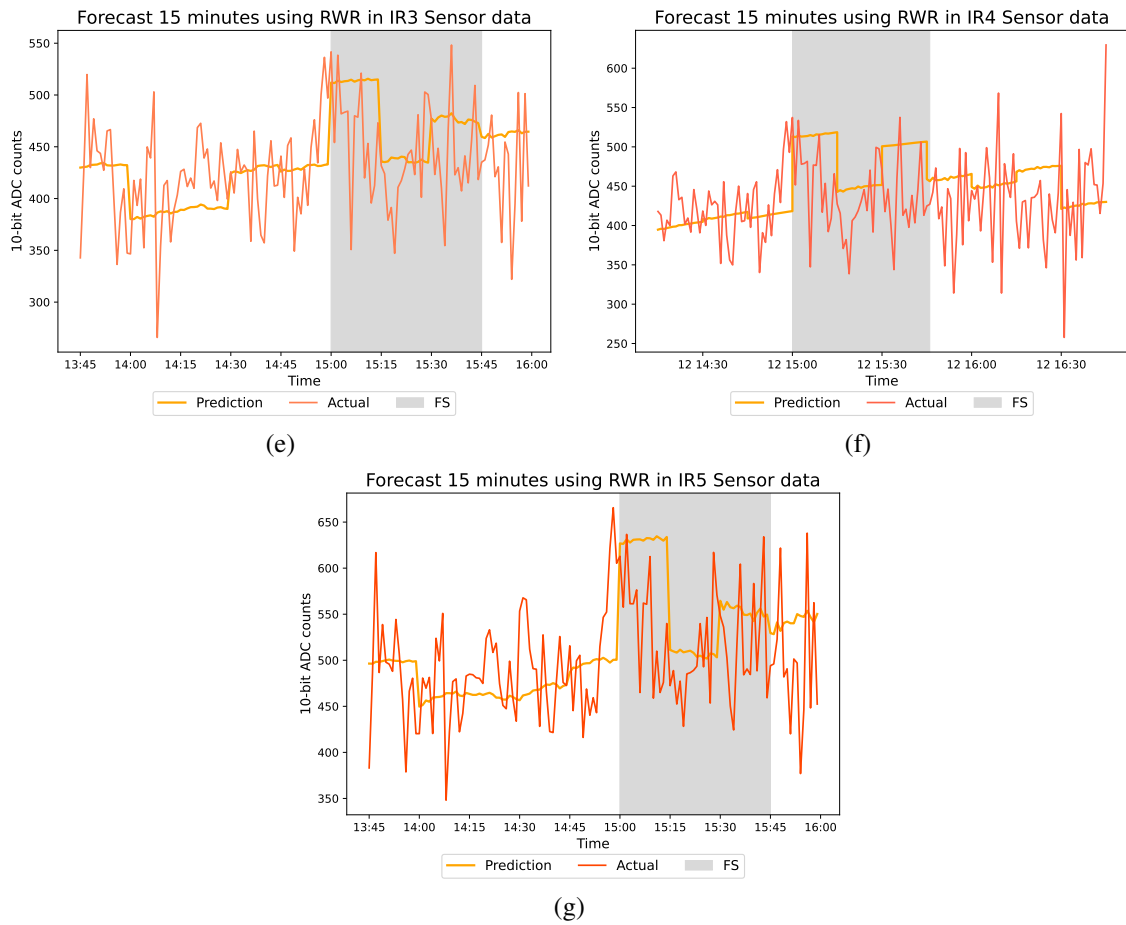


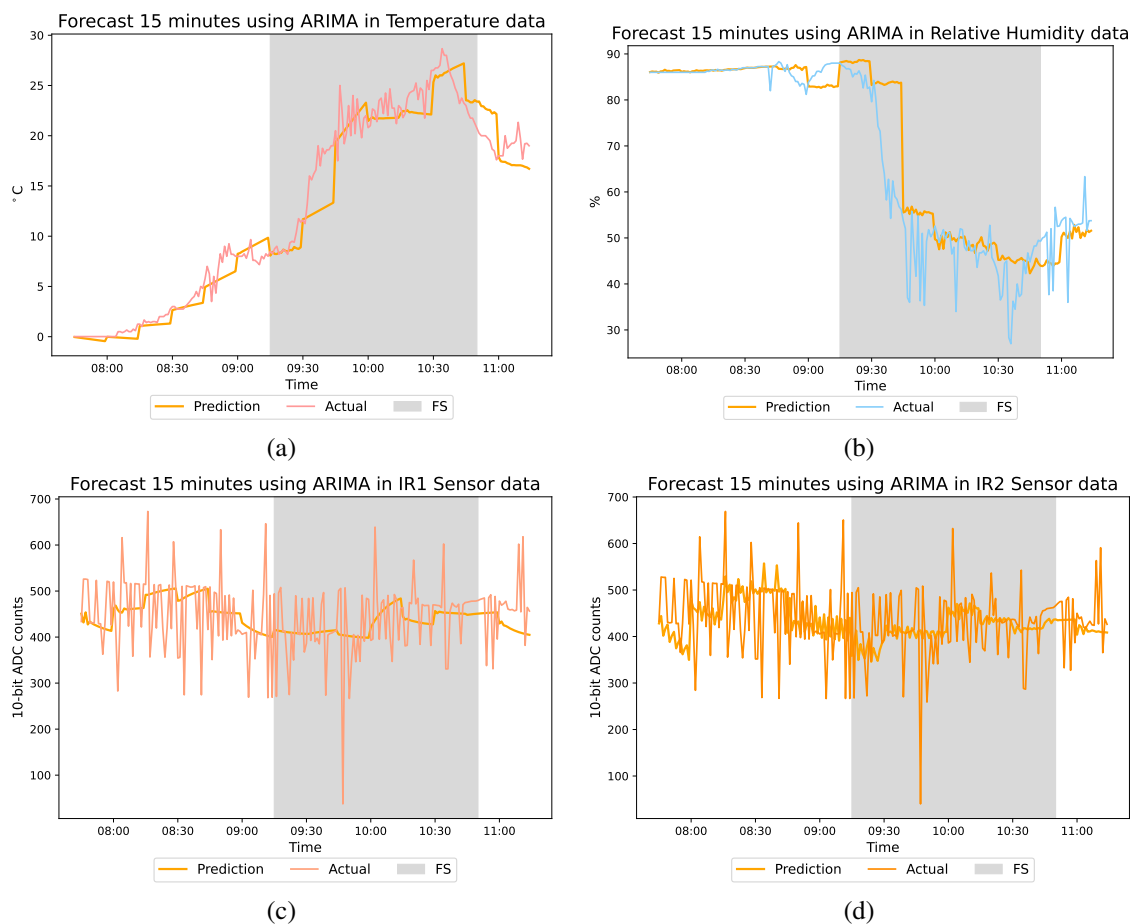
Figure D.17: Graphical representation of RWR Model's prediction variables' values in Dataset 2 during the WSE1. The plots display the actual versus predicted values for Temperature (a), Humidity (b), and IR1 through IR5 (c)-(g).

The predicted values for Temperature closely align with the actual values, showing differences of approximately $\pm 1^\circ\text{C}$ before the fire event. However, after 15h00, a noticeable divergence occurs between the forecasted and actual lines, which persists until approximately 15h30. This behavior indicates the model's difficulty in adapting to abrupt changes caused by the fire. For Humidity, the forecasted values are relatively close to the actual readings before the fire event. However, as the fire event begins, the predicted values start to deviate significantly from the actual values, mirroring the abrupt environmental changes during the event. This divergence continues until the end of the fire event. Regarding the IR sensors (IR1-IR5), the stepped behavior of the forecasted lines suggests an attempt to follow the average of the peaks formed by the actual values. However, multiple instances across all plots show a significant distance between the two lines, particularly during the fire event.

D.6 Visual Comparison of the ARIMA Predicted Values and Actual Sensor Readings for Dataset 1

This section extends the **ARIMA** analysis presented in the main text for Dataset 1 (Section 5.6.2.1). In the main body of the text, the **ARIMA** method was applied and discussed in terms of its performance metrics. Here, supplementary analyses focus on a visual comparison between the predicted and actual sensor readings.

The graphical plots in Figure D.18 aim to provide an intuitive assessment of the **ARIMA** model's ability to capture patterns and predict sensor values under the low **TCD** scenario. Highlighted **WSE** periods, as indicated by the **FS** column, mark the moments of fire events, offering critical context for evaluating the model's behavior during these high-variability intervals. These visualizations reinforce the observations from the main text and underscore the model's potential utility for fire anomaly detection.



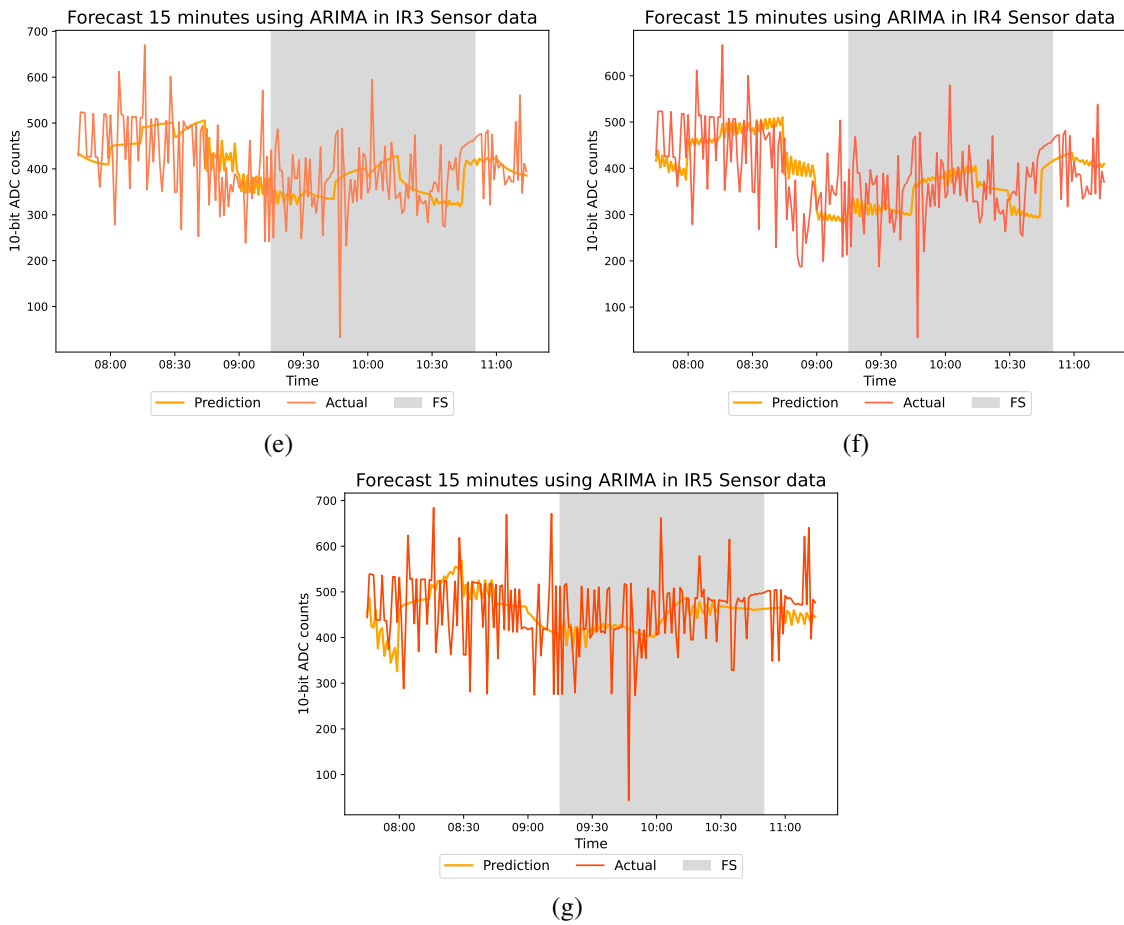


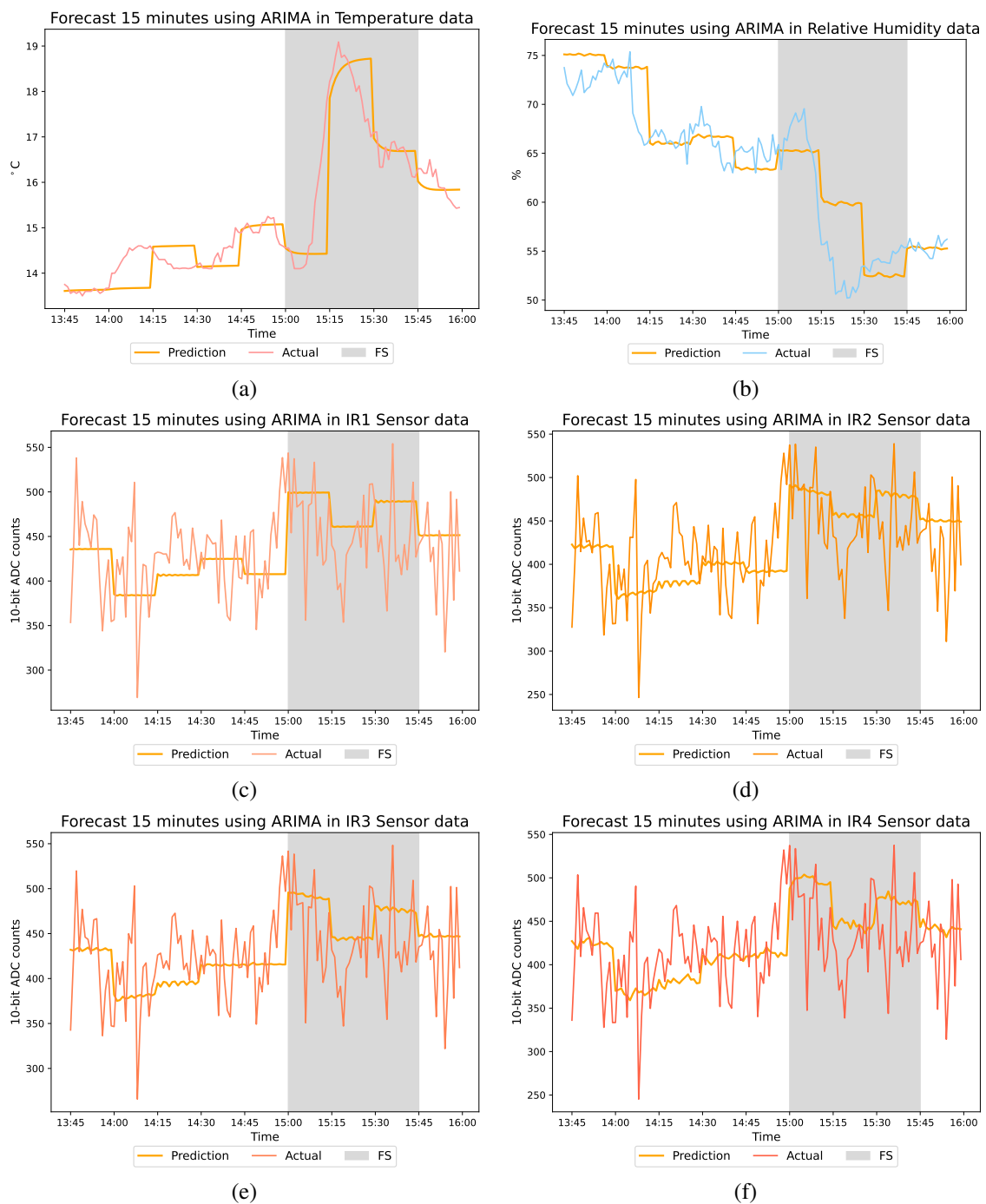
Figure D.18: Graphical representation of ARIMA prediction variables' values in Dataset 1. The plots display the actual versus predicted values for Temperature (a), Humidity (b), and IR1 through IR5 (c)-(g).

In most of the plots, the **ARIMA** model for each variable indicates a smoothing effect, a typical feature of this forecasting technique. In this sense, abrupt differences become evident when the **WSE** begins (9h15), as the model has difficulty adapting quickly to the sudden changes in environmental conditions. Notably, from 9h30 onward, the divergence between the forecasted and actual values is pronounced in all plots, highlighting the model's limitations in real-time response to rapid shifts.

D.7 Visual Comparison of the ARIMA Predicted Values and Actual Sensor Readings for Dataset 2

This section extends the **ARIMA** analysis presented in the main text for Dataset 2 (Section 5.6.2.2). In the main body of the text, the **ARIMA** method was applied and discussed in terms of its performance metrics. Here, supplementary analyses focus on a visual comparison between the predicted and actual sensor readings.

The graphical plots in Figure D.19 aim to provide an intuitive assessment of the ARIMA model's ability to capture patterns and predict sensor values under the high TCD scenario. Highlighted WSE periods, as indicated by the FS column, mark the moments of fire events, offering critical context for evaluating the model's behavior during these high-variability intervals. These visualizations reinforce the observations from the main text and underscore the model's potential utility for fire anomaly detection.



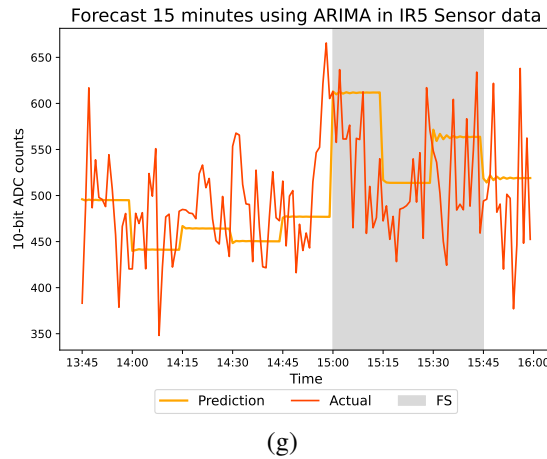


Figure D.19: Graphical representation of ARIMA prediction variables' values in Dataset 2. The plots display the actual versus predicted values for Temperature (a), Humidity (b), and IR1 through IR5 (c)-(g).

All plots illustrate the characteristic smoothing effect of the **ARIMA** model, which often results in forecasted values that follow a gradual trend rather than capturing sudden fluctuations. For the Temperature and Humidity variables, there is a proximity between the predicted and actual lines when compared to the outside of **WSE** region, indicating relatively accurate predictions under stable conditions. However, a more pronounced divergence between the actual and predicted lines is observed within the **WSE** region, particularly between 15h00 and 15h45. This highlights the **ARIMA** model's difficulty in accurately forecasting values during periods of abrupt environmental changes caused by the fire event.

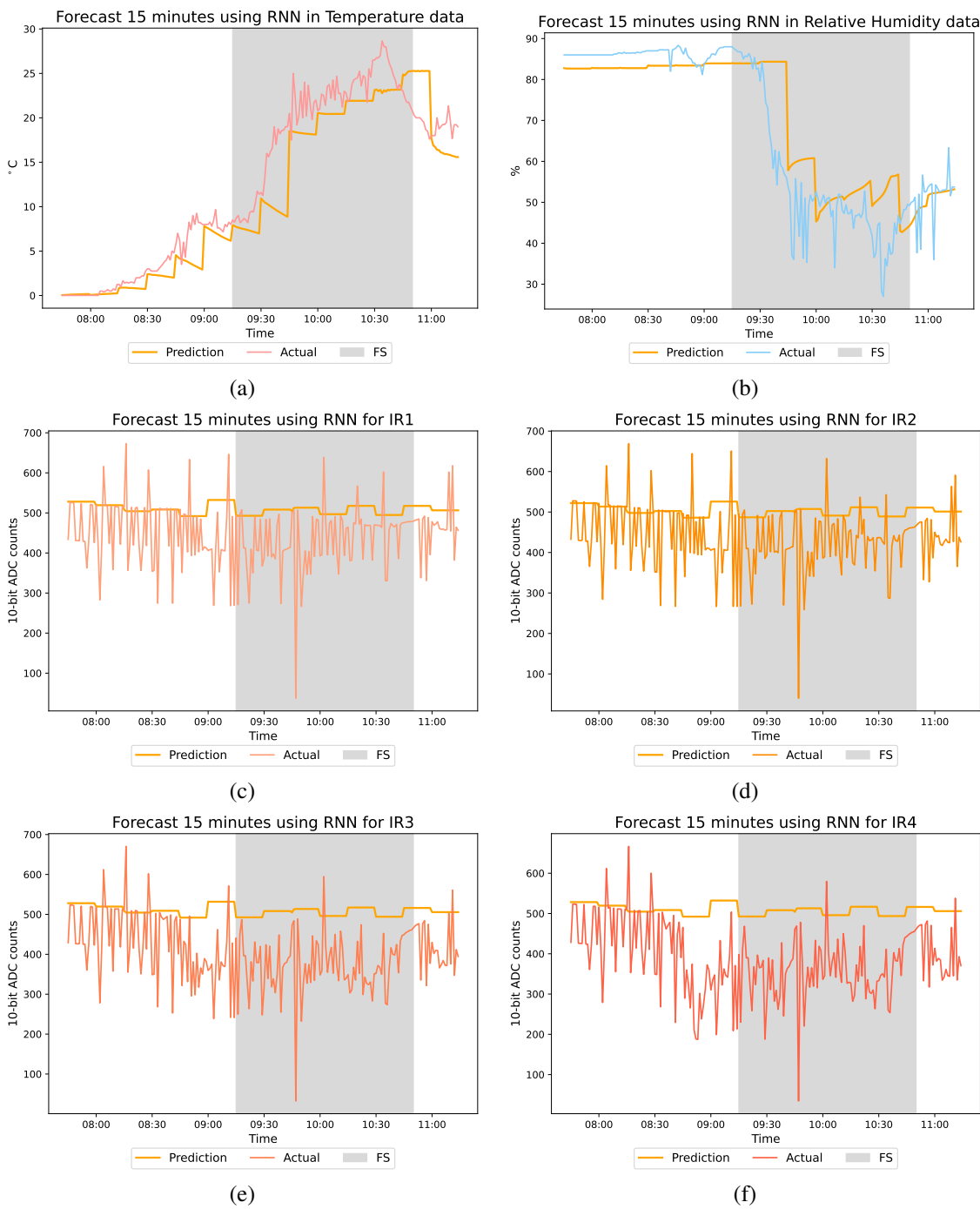
In contrast, the **IR** sensor plots show a different behavior. The **ARIMA** forecast lines tend to align with the average values of the peaks formed by the actual **IR** readings. This behavior reflects the model's inability to fully capture the extreme variations of **IR** data as it smooths over these changes, positioning the forecast line closer to an averaged level.

D.8 Visual Comparison of the RNN Predicted Values and Actual Sensor Readings for Dataset 1

This section extends the **RNN** analysis presented in the main text for Dataset 1 (Section 5.6.3.1). In the main body of the text, the **RNN** method was applied and discussed in terms of its performance metrics. Here, supplementary analyses focus on a visual comparison between the predicted and actual sensor readings.

The graphical plots in Figure D.20 aim to provide an intuitive assessment of the **RNN** model's ability to capture patterns and predict sensor values under the low **TCD** scenario. Highlighted **WSE** periods, as indicated by the **FS** column, mark the moments of fire events, offering critical context for evaluating the model's behavior during these high-variability intervals. These visualizations

reinforce the observations from the main text and underscore the model’s potential utility for fire anomaly detection.



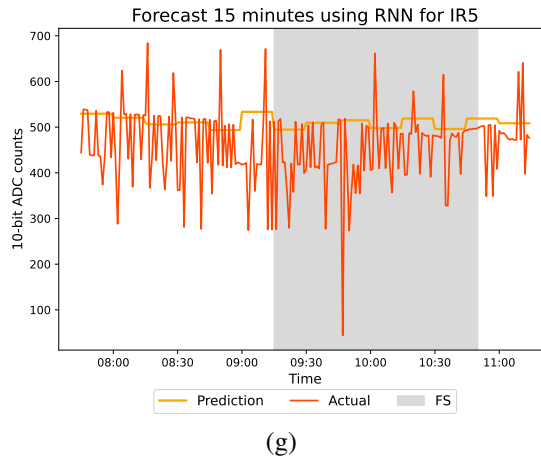


Figure D.20: Graphical representation of RNN prediction variables' values in Dataset 1. The plots display the actual versus predicted values for Temperature (a), Humidity (b), and IR1 through IR5 (c)-(g).

For Temperature, the forecasted values initially remain relatively close. However, immediately after 9h30, a significant divergence between the forecast and actual data becomes evident. Beyond this point, the predicted values attempt to track the actual trends but exhibit some difficulty. Similarly, for Humidity, the initial differences between forecasted and actual values are minimal. However, the divergence becomes pronounced after 9h30, persisting until the end of the **WSE** period. After the **WSE**, the forecasted and actual values align more closely, suggesting that the model effectively stabilizes once the event concludes.

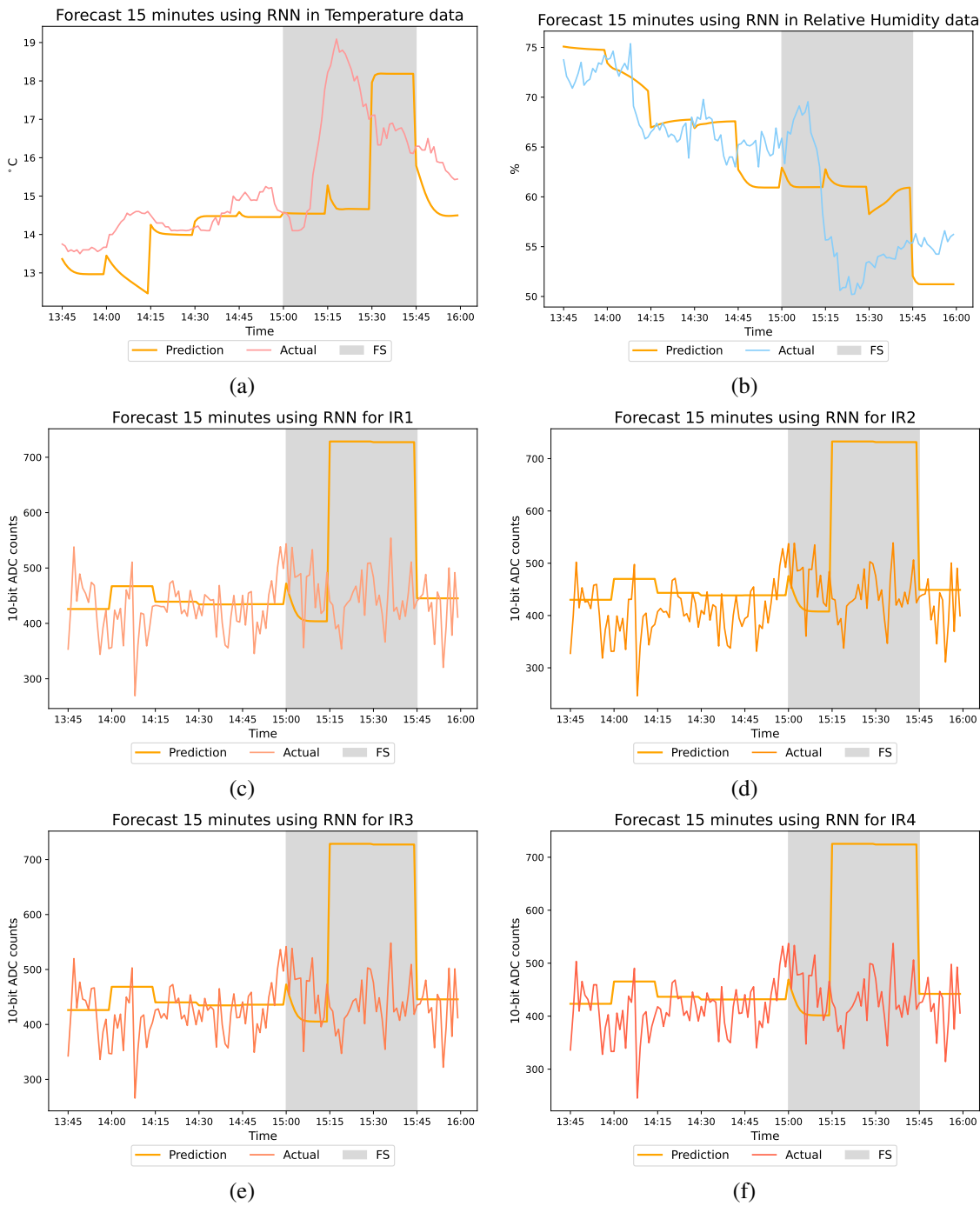
In the case of **IR** sensors (IR1-IR5), the forecasted values for all **IR** variables display a characteristic pattern of step-like increments, reflecting the model's potential difficulty in capturing the intricate patterns present in the data. Nevertheless, the predicted values generally fall within the mean range of the **IR** sensor peaks. Notably, around 9h45, the actual **IR** values exhibit significant differences compared to their corresponding forecasts.

D.9 Visual Comparison of the RNN Predicted Values and Actual Sensor Readings for Dataset 2

This section extends the **RNN** analysis presented in the main text for Dataset 2 (Section 5.6.3.2). In the main body of the text, the **RNN** method was applied and discussed in terms of its performance metrics. Here, supplementary analyses focus on a visual comparison between the predicted and actual sensor readings.

The graphical plots in Figure D.21 aim to provide an intuitive assessment of the **RNN** model's ability to capture patterns and predict sensor values under the high **TCD** scenario. Highlighted **WSE** periods, as indicated by the **FS** column, mark the moments of fire events, offering critical context for evaluating the model's behavior during these high-variability intervals. These visualizations

reinforce the observations from the main text and underscore the model’s potential utility for fire anomaly detection.



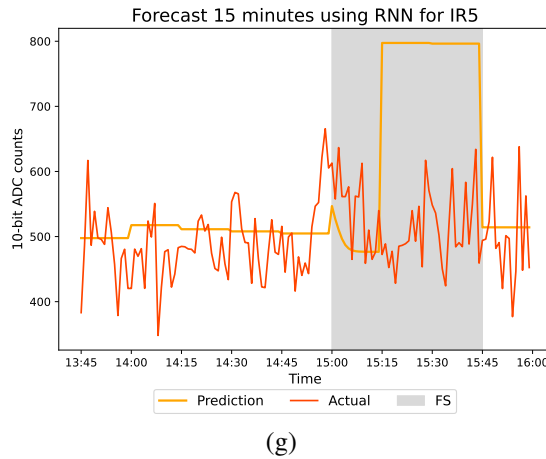


Figure D.21: Graphical representation of RNN prediction variables' values in Dataset 2 during first WSE. The plots display the actual versus predicted values for Temperature (a), Humidity (b), and IR1 through IR5 (c)-(g).

For Temperature, the predicted and actual values show a close alignment at the start of the testing period, indicating the **RNN** model's strong initial performance in forecasting the next 15 min. However, a divergence occurs after 15h00, coinciding with the **WSE** period. This divergence highlights the model's challenge in adapting to sudden, unforeseen changes. The forecast remains significantly different from the actual values throughout the **WSE**, with the discrepancies persisting until 15h45. After this period, the model returns to produce forecasts closer to the actual readings.

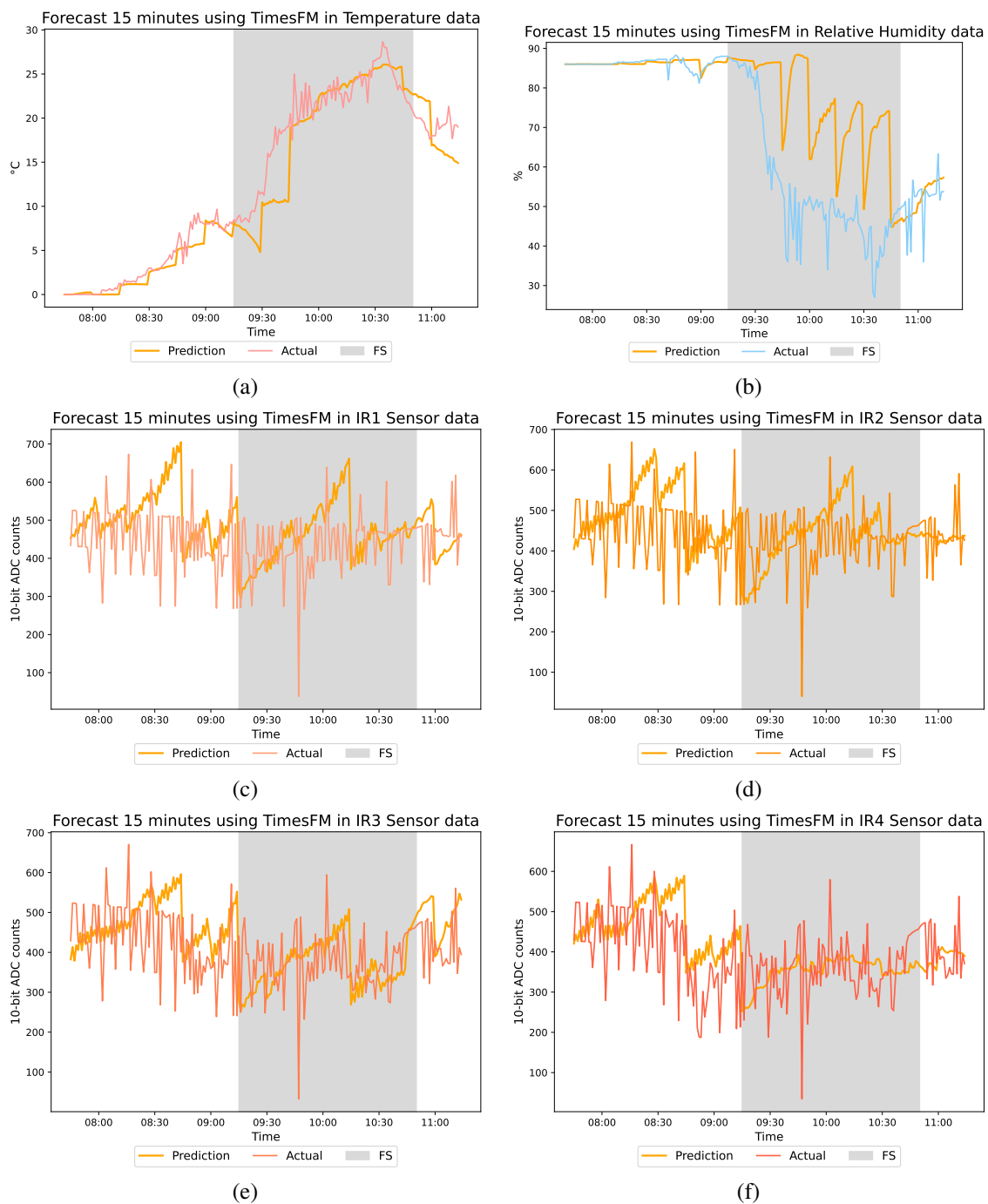
A similar pattern is observed for Humidity. Despite some step behavior in the forecasted values, it remains relatively centered around the midpoints of the peaks in the actual values before the **WSE**. However, during the **WSE**, starting from 15h15, the actual values drop abruptly, leading to a substantial gap between predicted and actual readings. This gap persists throughout the **WSE**, highlighting the model's inability to adapt to rapid environmental changes.

The behavior in the **IR** sensors plots mirrors the earlier observations from the Low **TCD** scenario. The **RNN** forecasts a nearly straight line before the **WSE**, maintaining values close to the midpoints of the peaks in the actual data. Which suggests a difficulty in capturing the patterns of **IR** sensor data. Nevertheless, compared to the Low **TCD** scenario, the predicted values in the High **TCD** context appear slightly closer to the actual peak midpoints.

D.10 Visual Comparison of the TimesFM Predicted Values and Actual Sensor Readings for Dataset 1

This section extends the **TimesFM** analysis presented in the main text for Dataset 1 (Section 5.6.4.1). In the main body of the text, the **TimesFM** method was applied and discussed in terms of its performance metrics. Here, supplementary analyses focus on a visual comparison between the predicted and actual sensor readings.

The graphical plots in Figure D.22 aim to provide an intuitive assessment of the TimesFM model's ability to capture patterns and predict sensor values under the low TCD scenario. Highlighted WSE periods, as indicated by the FS column, mark the moments of fire events, offering critical context for evaluating the model's behavior during these high-variability intervals. These visualizations reinforce the observations from the main text and underscore the model's potential utility for fire anomaly detection.



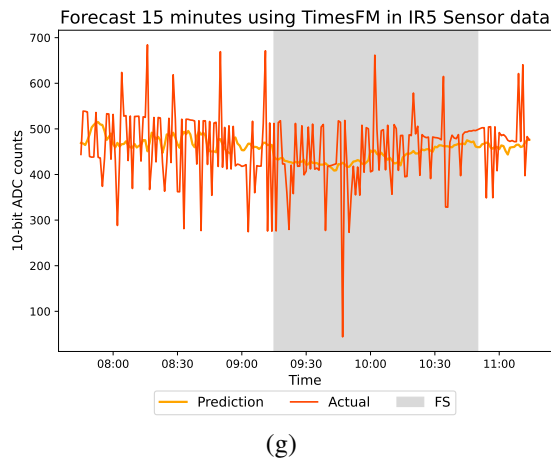


Figure D.22: Graphical representation of TimesFM prediction variables' values in Dataset 1. The plots display the actual versus predicted values for Temperature (a), Humidity (b), and IR1 through IR5 (c)-(g).

The Temperature plot initially shows a strong alignment between the predicted and actual values, indicating the model's ability to capture the general trend. However, after the **WSE** starts around 9h20, a noticeable divergence emerges between the forecasted and actual values. This difference becomes more pronounced between 9h30 and 9h45, highlighting the model's difficulty in adapting to sudden changes caused by the fire event. Although some discrepancy remains until the end of the **WSE**, the predicted and actual values converge significantly after the **WSE** concludes.

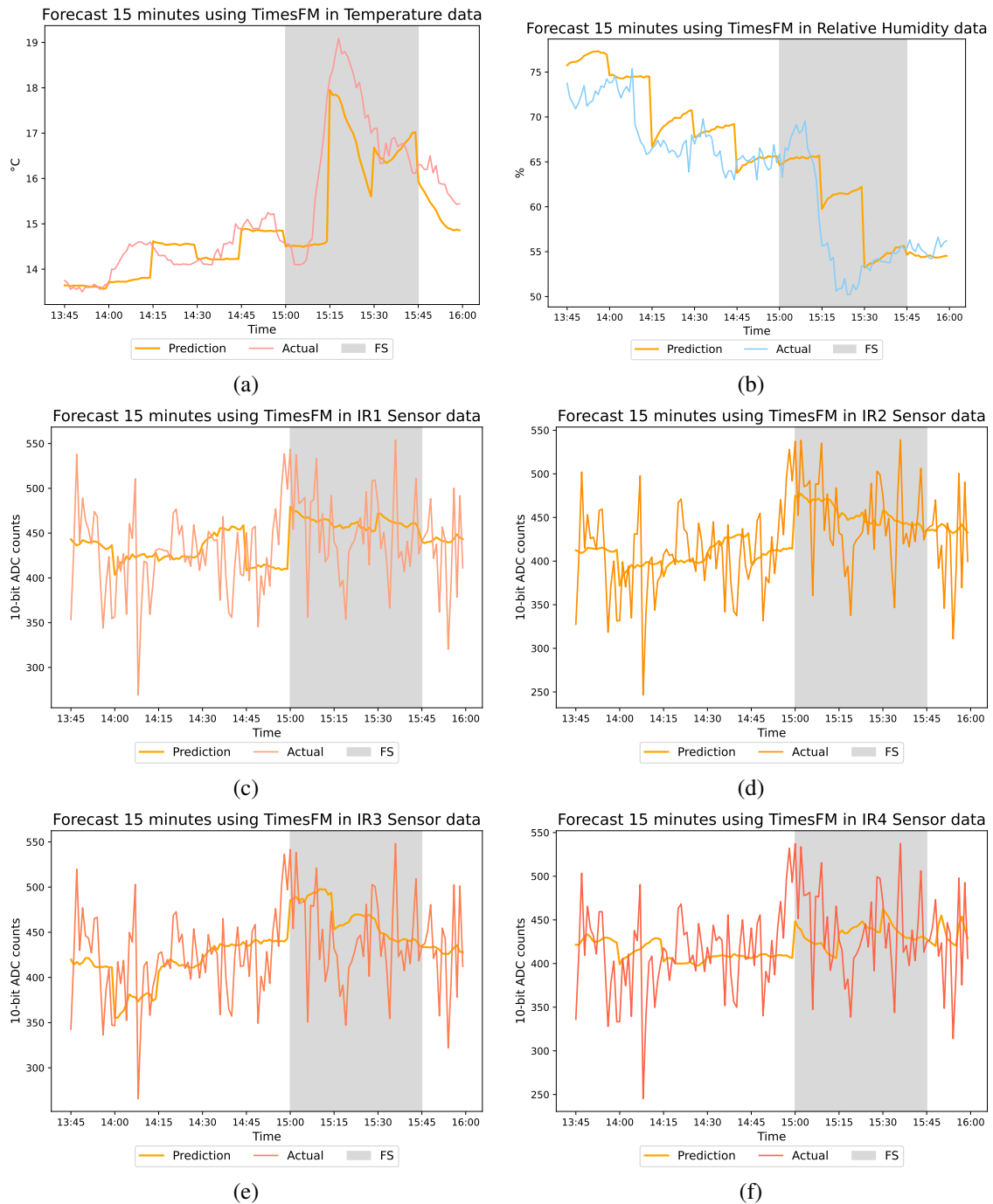
In the Humidity plot, the forecasted values closely match the actual values early in the analysis, maintaining this behavior until around 9h10. After this period, the lines diverge due to the abrupt changes introduced by the **WSE**. As the actual Humidity values drop sharply, the forecasted values fail to adjust quickly, further widening the gap. This behavior underscores the model's limitations in predicting sudden shifts in the data. After the **WSE**, the predicted and actual values become relatively close again.

For the **IR** variables (IR1-IR5), the plots reveal that before the **WSE**, the forecasted values generally follow the behavior of the actual values, indicating the model's ability to capture general patterns. However, a significant discrepancy is observed in all **IR** plots after the **WSE** begins, particularly around 9h45. This divergence indicates that the model had difficulty predicting the higher **IR** intensities introduced by the fire event, deviating from the usual data patterns.

D.11 Visual Comparison of the TimesFM Predicted Values and Actual Sensor Readings for Dataset 2

This section extends the **TimesFM** analysis presented in the main text for Dataset 2 (Section 5.6.4.2). In the main body of the text, the **TimesFM** method was applied and discussed in terms of its performance metrics. Here, supplementary analyses focus on a visual comparison between the predicted and actual sensor readings.

The graphical plots in Figure D.23 aim to provide an intuitive assessment of the **TimesFM** model's ability to capture patterns and predict sensor values under the high **TCD** scenario. Highlighted **WSE** periods, as indicated by the **FS** column, mark the moments of fire events, offering critical context for evaluating the model's behavior during these high-variability intervals. These visualizations reinforce the observations from the main text and underscore the model's potential utility for fire anomaly detection.



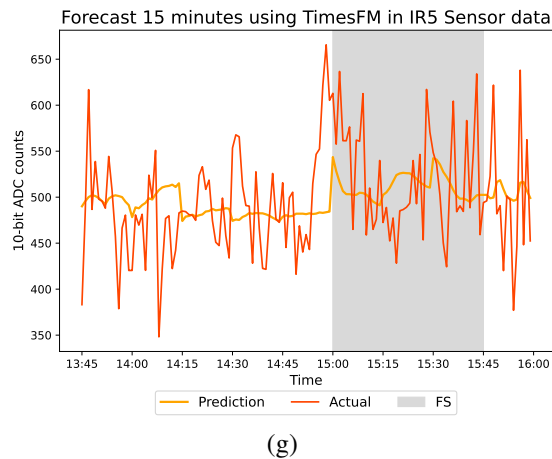


Figure D.23: Graphical representation of TimesFM prediction variables' values in Dataset 2. The plots display the actual versus predicted values for Temperature (a), Humidity (b), and IR1 through IR5 (c)-(g).

The predicted and actual values present distinct patterns across variables during the **WSE** period. For Temperature, the forecasts initially align well with the actual data, reflecting the model's ability to capture stable temporal trends. However, a noticeable divergence emerges after the **WSE** begins around 15h10, peaking by 15h15 and continuing for the subsequent iteration. As the temperature decreases, the model gradually returns its forecasting accuracy. For Humidity, the predicted values exhibit a step-like behavior, staying relatively close to the actual values before the **WSE**. A sudden drop in actual Humidity values occurs around 15h15, causing the forecasts to deviate significantly. This disparity highlights the challenges the **WSE** introduced, with predictions and actual values aligning again after the event concludes.

The **IR** sensor plots show a consistent pattern across IR1-IR5. Before the **WSE**, predicted values attempt to follow the actual trends. However, just before 15h00, significant differences appear for IR4 and IR5, with similar discrepancies occurring for other **IR** sensors around 15h05. Following the **WSE**, the predicted values maintain a position closer to the mean of the actual values' peaks, indicating difficulty in adapting to the spiky nature of the **IR** data.