FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

A Data-Driven Approach to Predicting Itinerary Delays in Multi-Modal Transit Networks

Catarina Isabel Moreira Canelas



Mestrado em Engenharia Informática e Computação

Supervisor: Prof. Rosaldo Rossetti

Second Supervisor: Prof. Lígia Conceição

July 31, 2026

© Catarina Isabel Moreira Canelas, 2025

A Data-Driven Approach to Predicting Itinerary Delays in Multi-Modal Transit Networks

Catarina Isabel Moreira Canelas

Mestrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

President: Prof. Daniel Castro Silva
Examiner: Prof. Francisco Nibau Antunes
Supervisor: Prof. Rosaldo Rossetti

July 31, 2026

Resumo

Os atrasos no transporte público reduzem a fiabilidade das viagens multimodais, ao aumentarem a incerteza dos tempos de viagem e comprometerem as ligações entre diferentes modos de transporte. Embora tenham sido propostas diversas abordagens baseadas em aprendizagem automática para a previsão de atrasos, a maioria dos estudos existentes centra-se em modos de transporte isolados ou na operação individual dos veículos, oferecendo um suporte limitado para a modelação da propagação de atrasos numa perspetiva orientada ao passageiro em redes de transporte interligadas. Esta dissertação propõe uma framework sensível ao contexto para a previsão de atrasos no transporte público multimodal, através da integração de horários *General Transit Feed Specification* (GTFS) com informação contextual sobre perturbações rodoviárias proveniente de eventos DATEX II. Os dados subjacentes foram recolhidos, limpos e integrados espacialmente ao nível da paragem e do segmento de viagem. Foi então aplicado, sobre este conjunto de dados integrado, um processo sistemático de feature engineering, de modo a caracterizar as dimensões temporal, operacional, topológica, de propagação e contextual que influenciam as viagens dos passageiros. Ao contrário das abordagens tradicionais baseadas em paragens, a framework proposta modela viagens completas dos passageiros, incorporando explicitamente eventos de transbordo, ligações perdidas, atrasos acumulados e efeitos de propagação de atrasos em redes multimodais. Foram estudados e comparados três paradigmas de previsão: *Extreme Gradient Boosting* (XGBoost), redes *Long Short-Term Memory* (LSTM) e *Graph Neural Networks* (GNN). Os modelos propostos foram inicialmente avaliados numa tarefa de previsão de atrasos em contexto mono-modal e posteriormente estendidos ao problema da previsão de atrasos em viagens multi-modais. Foram ainda realizadas análises de correlação, avaliação da importância das variáveis e um processo de seleção de atributos contextuais, com o objetivo de otimizar a seleção de atributos, preservando simultaneamente as variáveis essenciais para a modelação da propagação de atrasos. Os resultados experimentais demonstram que as variáveis relacionadas com a propagação de atrasos constituem o principal contributo para a capacidade preditiva dos modelos. As variáveis contextuais apresentaram uma influência preditiva relativamente reduzida, devido à reduzida ocorrência de perturbações rodoviárias nas proximidades das observações, embora tenham sido mantidas por contribuírem para uma representação operacional mais realista do problema. A framework proposta demonstra que a integração de mecanismos de propagação multimodal com informação contextual de natureza operacional permite representar de forma mais realista as viagens dos passageiros do que as abordagens convencionais baseadas num único modo de transporte. A metodologia desenvolvida constitui uma base para sistemas inteligentes de transporte capazes de suportar a previsão de atrasos orientada ao passageiro, a análise da fiabilidade dos transbordos e futuras aplicações de planeamento de viagens multimodais.

Palavras-chave: Previsão de Atrasos em Transportes Públicos • Propagação de Atrasos • Sistemas Inteligentes de Transporte • Previsão Sensível a Contexto • Transporte Multimodal • Aprendizagem Computacional • GTFS • DATEX II

Abstract

Public transport delays degrade the reliability of multimodal journeys by increasing travel time uncertainty and disrupting passenger transfers across interconnected services. While numerous machine learning techniques have addressed delay prediction in public transport, most existing approaches predominantly focus on single transport modes or individual vehicle trajectories, and consequently provide limited support for modelling passenger-oriented delay propagation through transfer-dependent multimodal networks. This dissertation proposes a context-aware machine learning framework for multimodal public transport delay prediction, integrating General Transit Feed Specification (GTFS) operational schedules with road disruption intelligence derived from DATEX II traffic event data. The underlying data was collected, cleaned, and spatially integrated at both stop and journey-segment level. A systematic feature engineering process was then applied to this integrated dataset to characterise the temporal, operational, topological, propagation-related, and contextual dimensions of passenger journeys. In contrast to conventional stop-level prediction approaches, the proposed framework explicitly models complete origin-to-destination passenger journeys, incorporating transfer events, connection failures, accumulated delay, and cross-modal propagation effects through a chain-based dataset representation. Three predictive paradigms were developed, trained, and comparatively evaluated: Extreme Gradient Boosting (XGBoost), Long Short-Term Memory (LSTM) networks, and Graph Neural Networks (GNNs). Models were assessed in two sequential stages: a single-mode delay prediction task establishing baseline performance, and a multimodal passenger delay prediction task evaluating propagation-aware generalisation. Feature space optimisation was guided by correlation analysis, importance evaluation, and contextual feature screening, balancing redundancy reduction with the preservation of propagation-relevant information. Experimental results demonstrate that delay propagation features dominate predictive performance across all evaluated configurations. Contextual road disruption variables exhibited limited direct predictive contribution, attributed primarily to the spatial sparsity of nearby traffic events relative to the stop-level prediction granularity. Nonetheless, these features were retained to preserve the operational realism and validity of the modelling framework. The proposed framework establishes that multimodal propagation mechanisms with contextual operational information create a substantially more realistic characterisation of passenger journey dynamics than conventional single-mode prediction systems. The resulting methodology contributes a foundation for intelligent transport systems oriented towards passenger-centric delay forecasting, transfer reliability assessment, and operational support for multimodal journey planning.

Keywords: Public Transport Delay Prediction • Delay Propagation • Intelligent Transportation Systems • Context-Aware Prediction • Multimodal Transportation • Machine Learning • GTFS • DATEX II

The United Nations Sustainable Development Goals

This dissertation contributes to several of the United Nations Sustainable Development Goals (SDGs) by supporting the development of intelligent, efficient and sustainable public transport systems. In particular, the proposed framework contributes to improving the reliability of multimodal public transport through context-aware delay prediction, enabling better passenger information and more efficient transport network management.

SDG	Target	Contribution	Performance Indicators and Metrics
11	11.2	Improves the reliability and accessibility of multimodal public transport by predicting passenger delays and supporting more informed travel decisions.	Prediction accuracy (MAE, Accuracy, Precision, Recall, F1-score); reliability of multimodal delay prediction.
9	9.1	Supports the development of intelligent transport infrastructure through machine learning models capable of modelling delay propagation across interconnected transport networks.	Prediction performance, model robustness (cross-validation), computational efficiency.
13	13.2	Promotes more efficient use of public transport by improving journey reliability, encouraging modal shift and contributing indirectly to more sustainable urban mobility.	Potential reduction of passenger travel uncertainty and improved operational efficiency.

Acknowledgements

Writing this thesis marks the end of a challenging, demanding, and deeply meaningful journey. Throughout this process, I was fortunate to be surrounded by people whose support, guidance, and encouragement were essential in helping me reach this milestone.

First, I would like to express my sincere gratitude to my supervisors, Professor Rosaldo Rossetti and Professora Lígia Conceição, for their guidance, availability, and support throughout this work. Their valuable feedback, constructive criticism, and expertise were fundamental to the development of this thesis. I am especially grateful for the trust they placed in me and for helping me find direction during moments of uncertainty.

To my family, especially my parents, Isabel and Fernando, and my sister, Sara, I owe my deepest gratitude. Thank you for being my safe place, for believing in me even when I doubted myself, and for giving me the strength to keep going. Your love, patience, and constant support, and the hugs that made everything feel a little easier, carried me through even the most difficult moments.

To my friends, thank you for making this journey lighter and funnier. Your support and laughter were a much-needed escape from the pressure of this work. A special thank you to Ana Sofia, for being present in the moments when I needed it most, for listening, encouraging me, and always knowing how to make me laugh. And a special thank you to Luís, for all the help, support, and encouragement throughout this course. Your help made a real difference in allowing me to reach this point.

To my boyfriend, Gonçalo, thank you for your love, patience, and unlimited support. You were a constant source of comfort and motivation, especially during the most overwhelming moments, always listening, encouraging me, and making even the hardest days feel lighter.

This journey taught me much more than the subject of this thesis. It taught me resilience, perseverance, and the importance of trusting the process, even when the path ahead feels uncertain. Above all, it reminded me that no achievement is truly individual, and that the people who walk beside us are often what make the journey meaningful.

Catarina Canelas

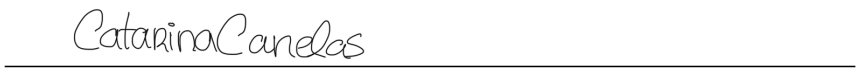
Declaration of honour

I declare that this dissertation is my own work and has not previously been submitted or assessed at this or any other institution. References to other authors (statements, ideas, thoughts), as well as the use of published works, strictly comply with the rules of attribution and are duly identified in the text and in the bibliographic references, in accordance with the applicable referencing standards. I am aware that the practice of plagiarism or self-plagiarism constitutes an academic offence, as established in the U.Porto Code of Ethics.

I further declare that I have included text(s) published in co-authorship, and that I have specified, with transparency and rigor, the contribution of each co-author, as well as stated whether there exists any other dissertation in which the same text(s) may also have been included. The authorization from the journal in which the work is published is also included, together with the express agreement signed by all co-authors authorizing the inclusion of the article.

I also declare that I have used Artificial Intelligence tools to complete part(s) of this dissertation, and that this use and its results are duly noted and have been carefully checked for errors, inaccuracies, unfounded attributions or other forms of bias in content and arguments, as well as determining authorship.

Catarina Isabel Moreira Canelas



"By recording your dreams and goals on paper, you set in motion the process of becoming the person you most want to be. Put your future in good hands - your own."

Mark Victor Hansen

Contents

1 Introduction **1**
- 1.1 Context . . . 1
- 1.2 Motivation . . . 2
- 1.3 Problem Definition . . . 3
- 1.4 Research Questions . . . 3
- 1.5 Document Structure . . . 4

2 Literature Review **5**
- 2.1 Review Protocol . . . 5
- 2.2 Identification Phase . . . 6
  - 2.2.1 Search Queries . . . 7
  - 2.2.2 Search Queries Results . . . 7
- 2.3 Screening Phase . . . 7
- 2.4 Eligibility Phase . . . 8
- 2.5 Included Documents . . . 8
- 2.6 State of the Art . . . 8
  - 2.6.1 Delay Prediction in Single-Mode Public Transport Systems . . . 8
  - 2.6.2 Machine Learning Approaches for Travel Time and Delay Prediction . . . 10
  - 2.6.3 Passenger-Centric and Decision-Oriented Delay Prediction . . . 13
  - 2.6.4 Multimodal and Intermodal Delay Propagation in Public Transport Systems 14
  - 2.6.5 Data Sources and Availability in Public Transport Delay Prediction . . . 14
  - 2.6.6 Critical Discussion and Research Gap . . . 15
- 2.7 Main Findings from the Literature . . . 16

3 Methodological Approach **20**
- 3.1 Proposed Framework . . . 20
- 3.2 Problem Formulation . . . 21
- 3.3 Research Context and Data Sources . . . 23
- 3.4 Data Collection . . . 24
  - 3.4.1 GTFS Data . . . 24
  - 3.4.2 Road Closure Data . . . 26
  - 3.4.3 Traffic Event Data . . . 27
  - 3.4.4 Vehicle Occupancy Data . . . 28
- 3.5 Data Understanding . . . 29
  - 3.5.1 Multimodal Network Representation . . . 29
- 3.6 Dataset Construction . . . 30
  - 3.6.1 Initial Single-mode Stop-Level Dataset Construction . . . 30
  - 3.6.2 Multimodal Transfer-Based Dataset Construction . . . 32

3.7 Delay Simulation . . . 33
3.7.1 Vehicle Delay Simulation . . . 33
3.7.2 Passenger Delay Propagation . . . 35
3.8 Data Preprocessing . . . 37
3.8.1 Initial GTFS Dataset Preprocessing . . . 37
3.8.2 Multimodal Dataset Preprocessing . . . 37
3.9 Contextual Data Integration . . . 38
3.9.1 Stop-Level Contextual Integration . . . 38
3.9.2 Transfer-Level Contextual Integration . . . 39
3.10 Feature Engineering . . . 40
3.10.1 Single-Mode Feature Selection . . . 40
3.10.2 Single-Mode Correlation Analysis . . . 42
3.10.3 Single-Mode Feature Importance and Contextual Feature Screening . . . 45
3.11 Multimodal Feature Engineering . . . 47
3.11.1 Features Characterization . . . 47
3.11.2 Final Feature Selection . . . 56
3.12 Expected Outcomes . . . 57

4 Prediction Models **58**
4.1 Overview of the Evaluated Predictive Models . . . 58
4.2 Single-mode Prediction Models . . . 59
4.2.1 Extreme Gradient Boosting (XGBoost) . . . 59
4.2.2 Long Short-Term Memory Networks (LSTM) . . . 60
4.2.3 Graph Neural Networks (GNN) . . . 61
4.3 Multimodal Prediction Models . . . 63
4.3.1 Prediction Objectives . . . 63
4.4 XGBoost Model for Multimodal Passenger Delay Propagation . . . 64
4.4.1 Input Representation . . . 64
4.4.2 Model Implementation . . . 65
4.4.3 Hyperparameter Configuration . . . 65
4.4.4 Long Short-Term Memory Model for Multimodal Passenger Delay Propagation . . . 66
4.4.5 Input Representation . . . 66
4.4.6 Model Implementation . . . 66
4.4.7 Sequence Representation . . . 66
4.4.8 Hyperparameter Configuration . . . 67
4.5 Graph Neural Network Model for Multimodal Passenger Delay Propagation . . . 67
4.5.1 Graph Representation . . . 68
4.5.2 Model Implementation . . . 68
4.6 Evaluation Methodology . . . 69
4.6.1 Experimental Configurations and Validation . . . 69
4.6.2 Evaluation Metrics . . . 70

5 Experimental Results and Discussion **71**
5.1 Qualitative Inspection of the Multimodal Network . . . 71
5.2 Contextual Data Analysis . . . 71
5.2.1 Contextual Event Coverage . . . 71
5.2.2 Contextual Event Intensity . . . 73
5.3 Contextual Feature Screening . . . 75

5.4 Single-Mode Delay Prediction . . . 75
5.4.1 Experimental Setup . . . 75
5.4.2 XGBoost Results . . . 76
5.4.3 LSTM Results . . . 77
5.4.4 GNN Results . . . 77
5.4.5 Single-Mode Model Performance Comparison . . . 78
5.5 Multimodal Passenger Delay Prediction . . . 80
5.5.1 Problem Formulation . . . 80
5.5.2 Experimental Setup . . . 81
5.5.3 XGBoost Results . . . 81
5.5.4 LSTM Results . . . 88
5.5.5 GNN Results . . . 90
5.5.6 Multimodal Model Performance Comparison . . . 93
5.5.7 Comparative Error Analysis . . . 94
5.6 Single-Mode vs Multimodal Prediction . . . 96
5.7 Discussion . . . 100
5.8 Final Model Selection . . . 100
5.9 Practical Prediction Examples . . . 101

6 Conclusions **103**
6.1 Summary of Work . . . 103
6.2 Addressing the Research Questions . . . 104
6.3 Validation of Hypotheses . . . 105
6.4 Overall Contributions . . . 106
6.5 Limitations and Future Work . . . 107

References **110**

A Feature Engineering and Correlation Analysis **114**

B Supplementary Experimental Results **121**
B.1 Network Visualization Examples . . . 121
B.2 Complete Cross-Validation Results . . . 121
B.3 Additional Performance Comparisons . . . 121

List of Figures

1.1 Illustrative example of delay propagation across a multimodal passenger journey: an initial delay accumulates through successive transfers, increasing overall journey delay. . . . 2

2.1 PRISMA flowchart . . . 6

3.1 Overview of the proposed framework, from data integration and dataset construction to predictive modelling and experimental evaluation. . . . 21
3.2 Data sources and integration pipeline, from raw GTFS and DATEX II feeds to the integrated context-aware dataset. . . . 25
3.3 Spatial representation of the Dutch multimodal public transport network derived from GTFS shapes and stops. . . . 30
3.4 Construction pipeline of the initial single-mode, stop-level dataset. . . . 31
3.5 Construction pipeline of the multimodal, transfer-based dataset. . . . 32
3.6 Transfer feasibility and passenger delay propagation process at a transfer stop. . . . 36
3.7 Initial feature correlation matrix. . . . 43
3.8 Correlation matrix with final features. . . . 45
3.9 Feature importance analysis. . . . 46
3.10 Initial correlation matrix before feature engineering. . . . 49
3.11 Correlation matrix after feature engineering process. . . . 54
3.12 Feature importance. . . . 56

5.1 Interactive visualization interface for inspecting the Dutch multimodal public transport network. . . . 72
5.2 Identification of multimodal stops and associated transport services in the visualization interface. . . . 72
5.3 Contextual event coverage across observations . . . 73
5.4 Distribution of nearby contextual events in affected transport segments, comparing the number of road closures and traffic events detected around each segment. . . . 74
5.5 Cross-validation distributions after contextual feature screening. . . . 75
5.6 Cross-validation accuracy distributions . . . 79
5.7 Cross-validation MAE distributions. . . . 79
5.8 XGBoost prediction scatter plot for *target_passenger_delay_after*, showing strong alignment for most low-to-moderate passenger delays and larger deviations for rare extreme delay cases. . . . 83
5.9 XGBoost prediction scatter plot for *target_final_passenger_delay*, showing the relationship between actual and predicted final passenger delays. The dashed line represents the ideal prediction line. . . . 84

5.10 SHAP feature importance summary for the final journey delay prediction model, showing the mean absolute SHAP value for each feature across the sampled test instances. . . . 87
5.11 Predicted versus observed next-segment passenger delays obtained by the LSTM model. . . . 89
5.12 Predicted versus observed final passenger delays obtained by the LSTM model. . . . 90
5.13 Predicted versus observed next-segment passenger delays obtained by the GNN model. . . . 92
5.14 Predicted versus observed final passenger delays obtained by the GNN model. . . . 92
5.15 Cross-validation accuracy distribution for the next-segment passenger delay target across the XGBoost, LSTM, and GNN models. . . . 93
5.16 Cross-validation accuracy distribution for the final passenger delay target, showing consistently high classification performance across the multimodal models, with the GNN and LSTM achieving higher median accuracy than XGBoost. . . . 94
5.17 Distribution of absolute prediction errors for the next-segment passenger delay target across actual delay intervals, comparing XGBoost, LSTM, and GNN. . . . 95
5.18 Absolute prediction error distribution for the final passenger delay target across actual delay intervals, showing increasing error dispersion for longer accumulated delays, with the LSTM maintaining lower errors than XGBoost and GNN in high-delay journeys. . . . 95
5.19 Classification accuracy comparison between the single-mode prediction task and the two multimodal passenger delay targets across XGBoost, LSTM, and GNN. . . . 99
5.20 Mean absolute error comparison between the single-mode prediction task and the two multimodal passenger delay targets across XGBoost, LSTM, and GNN. . . . 99

A.1 Multimodal correlation matrix after M1. . . . 114

A.2 Multimodal correlation matrix after M2. . . . 116
A.3 Multimodal correlation matrix after M3. . . . 116
A.4 Multimodal correlation matrix after M4. . . . 117
A.5 Multimodal correlation matrix after M5. . . . 117
A.6 Multimodal correlation matrix after M6. . . . 118
A.7 Multimodal correlation matrix after M7. . . . 118
A.8 Multimodal correlation matrix after M8. . . . 119
A.9 Multimodal correlation matrix after M9. . . . 119
A.10 Multimodal correlation matrix after M10. . . . 120

B.1 Example of an individual tram route selected in the interactive network visualization interface. . . . 122
B.2 Comparison of classification performance metrics across XGBoost, LSTM and GNN in the single-mode setting. . . . 124
B.3 Cross-validation MAE distribution for next-segment passenger delay prediction in the multimodal setting. . . . 125
B.4 Cross-validation MAE distribution for final passenger delay prediction in the multimodal setting. . . . 125
B.5 Absolute prediction error as a function of actual passenger delay for the final passenger delay target. . . . 126
B.6 Absolute prediction error as a function of actual passenger delay for the next-segment passenger delay target. . . . 126

List of Tables

2.1 Summary of Search Queries . . . 7
2.2 Selected studies included in the State of the Art . . . 17
2.3 Selected studies included in the State of the Art . . . 18
2.4 Main research themes, focus, limitations, and associated studies . . . 19

3.1 Main GTFS files available in the Netherlands OpenOV dataset. . . . 25
3.2 Traffic event severity weights used to compute the traffic disruption score. . . . 28
3.3 Features extracted for the multimodal delay propagation dataset before feature selection. . . . 48
3.4 Final feature set used for multimodal delay prediction after feature engineering and feature selection. . . . 57

4.1 Hyperparameter configuration adopted for the XGBoost model. . . . 60
4.2 Hyperparameter configuration adopted for the LSTM model. . . . 61
4.3 Hyperparameter configuration adopted for the GNN model. . . . 62
4.4 Differences between the two prediction objectives. . . . 63
4.5 XGBoost model hyperparameter configuration. . . . 65
4.6 LSTM model hyperparameter configuration. . . . 67
4.7 GNN model configuration. . . . 69
4.8 Evaluation metrics used in the experimental study. . . . 70

5.1 XGBoost performance obtained using 10-fold cross-validation. . . . 76
5.2 LSTM performance obtained using 10-fold cross-validation. . . . 77
5.3 GNN performance obtained using 10-fold cross-validation. . . . 78
5.4 Comparison of the predictive performance obtained by the evaluated single-mode models using 10-fold cross-validation. . . . 78
5.5 Comparison between the single-mode and multimodal prediction problems. . . . 81
5.6 XGBoost predictive performance obtained using 10-fold cross-validation. Values are reported as mean [95% Confidence Interval (CI)]. . . . 82
5.7 Average XGBoost feature importance obtained over the 10 cross-validation folds for both prediction targets. Only the fifteen most relevant features are presented. . 85
5.8 Average SHAP feature importance obtained for the final journey delay prediction task. Values correspond to the mean absolute SHAP values computed over 30,000 sampled observations across the ten cross-validation folds. Only the fifteen most relevant features are presented. . . . 86
5.9 LSTM predictive performance obtained using 10-fold cross-validation. Values are reported as mean [95% CI]. . . . 88
5.10 GNN predictive performance obtained using 10-fold cross-validation. Values are reported as mean [95% CI]. . . . 90

5.11 Average predictive performance of the evaluated multimodal models obtained using 10-fold cross-validation for both passenger delay prediction targets. Best results for each target are highlighted in bold. 94
5.12 Regression performance comparison between the evaluated single-mode and multimodal prediction models using 10-fold cross-validation. Results are reported as mean ± standard deviation. Lower MAE and higher R^2 indicate better predictive performance. 97
5.13 Classification performance comparison between the evaluated single-mode and multimodal prediction models using 10-fold cross-validation. Results are reported as mean ± standard deviation. Higher Accuracy, Precision, Recall and F1-score indicate better predictive performance. 98
5.14 Summary of the selected predictive models for each prediction scenario. 101
5.15 Practical examples of final passenger delay predictions obtained with the selected multimodal LSTM model. 102
5.16 Transfer-level passenger delay predictions for a successful multimodal journey without missed connections (Journey ID = 31492) and transfer-level passenger delay predictions for a successful multimodal journey with missed connections (Journey ID = 22469). 102

A.1 Summary of the feature engineering process based on the correlation analysis. . . 115

B.1 10-fold cross-validation results for the XGBoost model in the single-mode setting. 121
B.2 10-fold cross-validation results for the LSTM model in the single-mode setting. . 122
B.3 10-fold cross-validation results for the GNN model in the single-mode setting. . . 122
B.4 10-fold cross-validation results for the XGBoost model in the multimodal setting. 123
B.5 10-fold cross-validation results for the LSTM model in the multimodal setting. . 123
B.6 10-fold cross-validation results for the GNN model in the multimodal setting. . . 124

List of Acronyms

ACM	Association for Computing Machinery
API	Application Programming Interface
AVL	Automatic Vehicle Location
CI	Confidence Interval
DATEXII	DATEX II (Data EXchange)
DNN	Deep Neural Network
GCN	Graph Convolutional Network
GNN	Graph Neural Network
GTFS	General Transit Feed Specification
IEEE	Institute of Electrical and Electronics Engineers
JSON	JavaScript Object Notation
LSTM	Long Short-Term Memory
NDW	Nationaal Dataportaal Wegverkeer (Dutch National Data Warehouse for Traffic Information)
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
RQ	Research Question
STGCN	Spatio-Temporal Graph Convolutional Network
STCP	Sociedade de Transportes Colectivos do Porto
WOA	Whale Optimization Algorithm
XGBoost	eXtreme Gradient Boosting
XML	Extensible Markup Language
AVL	Automatic Vehicle Location
SHAP	SHapley Additive exPlanations

Chapter 1

Introduction

1.1 Context

Urban public transport is widely regarded as a critical foundation of sustainable mobility, supporting cities in reducing private vehicle use, traffic congestion, and greenhouse gas emissions. In dense, well-integrated metropolitan areas, the coexistence of bus, tram, metro, rail, and ferry services creates highly interdependent multimodal networks. Although these systems can operate efficiently under normal conditions, disruptions in one mode, such as a late bus or a temporary road closure, can propagate through the network, causing passengers to miss connections, increasing overall travel times, and reducing journey reliability [30].

A substantial quantity of research has addressed delay prediction in public transport. Studies on bus arrival forecasting, for instance, have applied gradient boosting and deep learning models that integrate operational, traffic, and contextual variables, achieving improved prediction accuracy compared with traditional statistical approaches [5, 32]. Similarly, research on railway systems has explored ensemble and sequential learning methods to capture delay evolution along individual trips [36, 40]. In multimodal contexts, [30] proposes a data-driven approach that evaluates candidate itineraries to identify the most reliable routes, aggregating the probability of making each connection within a given travel-time budget.

Although these studies have improved prediction accuracy and journey planning, most existing approaches remain confined to a single transport mode or model delay propagation only implicitly, without explicitly representing passenger transfers, missed connections, and the cumulative delay experienced across an entire multimodal journey. This is a significant limitation, since the practical impact of operational disruptions is ultimately experienced by passengers at the journey level, not at the level of individual vehicles. Addressing this gap is essential to improve the reliability of multimodal travel planning and to support the development of passenger-centred predictive systems.

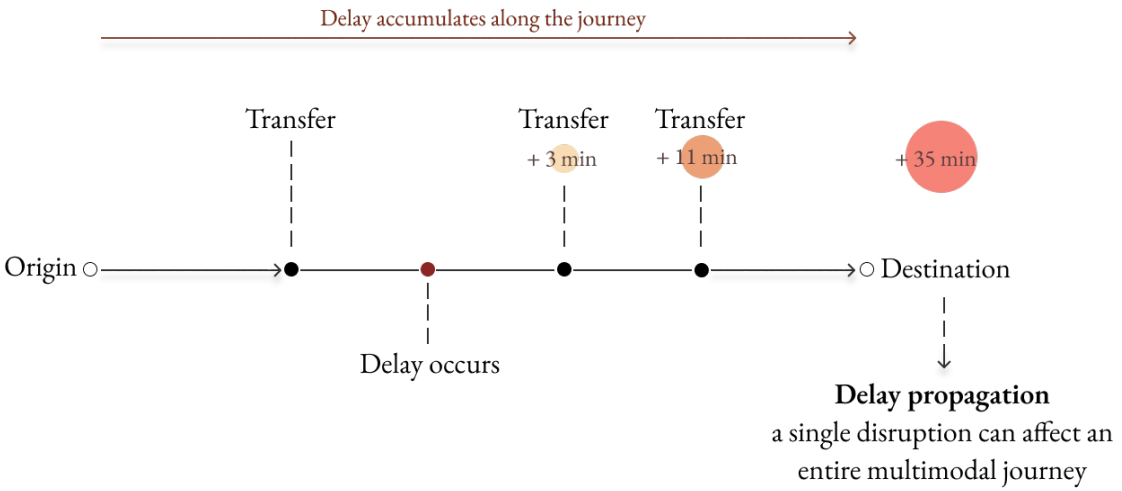


Figure 1.1: Illustrative example of delay propagation across a multimodal passenger journey: an initial delay accumulates through successive transfers, increasing overall journey delay.

1.2 Motivation

Reliable public transport is essential for equitable access to employment, education, and healthcare, while also supporting environmental and economic sustainability goals. In dense multimodal networks, even minor delays, such as a bus running a few minutes late, can propagate through connecting services, causing missed transfers, longer travel times, and reduced confidence in the system. Most passengers currently rely on static timetables or, at best, real-time information limited to the vehicle they are travelling on, without any indication of how a delay might cascade through subsequent legs of their journey. Without reliable predictions of cascading delays, passengers are left to make travel decisions under uncertainty, often relying on intuition rather than data. As illustrated in Figure 1.1, a single disruption can affect an entire multimodal journey.

The growing availability of open, standardised transport data, such as General Transit Feed Specification (GTFS) schedules and DATEX II (Data EXchange) (DATEXII) traffic disruption feeds, creates an opportunity to integrate heterogeneous data sources into unified, data-driven predictive frameworks. Such frameworks can move beyond conventional vehicle-level delay prediction towards modelling how delays accumulate and propagate through passenger journeys involving multiple transport modes and transfer points. This is particularly relevant given that few existing studies explicitly model the transfer-dependent, cumulative nature of passenger delay, instead treating each transport service in isolation.

Despite progress in delay prediction research, there remains a lack of integrated frameworks capable of combining real-time operational data, contextual disruption information, and explicit transfer modelling to estimate passenger-level delay propagation across complete multimodal journeys. Addressing this gap is relevant not only from a methodological perspective, but also for supporting passenger-centred intelligent transport systems that can ultimately encourage the use of public transport, reduce car dependency, and contribute to broader smart mobility and sustainability goals.

1.3 Problem Definition

Urban multimodal transit networks are highly interdependent: buses, trams, metro, rail, and ferry services rely on each other's punctuality to maintain reliable passenger connections. A delay in one service can propagate across the network, causing passengers to miss planned connections and increasing total journey time. Although operational schedule data, such as GTFS feeds, and contextual disruption data, such as DATEX II traffic and road closure feeds, are increasingly available, existing predictive approaches remain limited in scope. Most focus on estimating delays at the level of an individual vehicle or stop, without explicitly representing how a disruption on one service may affect the feasibility of a passenger's subsequent connections within a multimodal itinerary.

Most current delay prediction approaches either focus on a single transport mode, rely primarily on historical averages, or do not explicitly model cross-modal delay propagation through passenger transfers. Consequently, predictions generated at the vehicle level do not directly translate into reliable estimates of the delay actually experienced by a passenger completing a multimodal journey, particularly when connections are missed and additional waiting time is incurred. This limitation reduces the practical value of existing prediction systems for journey-level reliability assessment and discourages passengers from making informed, real-time travel decisions.

Developing a context-aware, data-driven framework that integrates operational schedule data with contextual disruption information, and that explicitly models passenger journeys as sequences of transfer-connected segments, can provide more realistic and informative estimates of delay propagation. Such a framework would support passenger-centred decision-making and contribute to the broader goal of more reliable and resilient multimodal public transport systems.

1.4 Research Questions

Given the problem identified in the previous section, the purpose of this dissertation is to investigate how data-driven and machine learning approaches can be used to model and predict delay propagation across multimodal public transport networks, from the perspective of the passenger journey rather than the individual vehicle. To address this problem, the following research questions were formulated:

Research Question (RQ)1 How can heterogeneous open data sources, namely operational schedule data (GTFS) and contextual disruption data (DATEX II), be integrated into a unified, passenger-oriented representation suitable for modelling multimodal delay propagation?

RQ2 How does explicitly modelling passenger transfers, missed connections, and cumulative delay propagation, rather than predicting delays at the level of individual vehicles, affect the accuracy and realism of public transport delay prediction?

RQ3 Among complementary machine learning paradigms, tabular (XGBoost), sequential(LSTM), and graph-based (GNN) models, which is best suited to capturing delay propagation at the operational (single-mode) level and at the passenger-journey (multimodal) level?

RQ4 To what extent does the inclusion of contextual traffic disruption information improve the predictive performance of multimodal passenger delay propagation models, compared with relying solely on operational and historical delay features?

1.5 Document Structure

The remainder of this document is organised into six further chapters, as follows. Chapter 2 presents the systematic literature review conducted to identify and characterise the existing body of research on delay prediction in public transport, following the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) methodology, and concludes with the state-of-the-art analysis of the selected studies and the identification of the research gap addressed by this dissertation. Chapter 3 presents the methodological approach, covering the problem formulation, data collection and integration, the delay simulation and passenger delay propagation mechanism, data preprocessing, the feature engineering procedures applied to both the single-mode and multimodal datasets, and the evaluation strategy adopted to assess the proposed framework. Chapter 4 describes the predictive models evaluated in this work, namely eXtreme Gradient Boosting (XGBoost),Long Short-Term Memory (LSTM), and Graph Neural Network (GNN), detailing their architectures, input representations, and hyperparameter configurations for both the single-mode and multimodal prediction tasks. Chapter 5 reports and discusses the experimental results obtained for both prediction scenarios, including the contextual feature screening analysis, the comparative evaluation of the three learning paradigms, and the final model selection. Finally, Chapter 6 revisits the main contributions of this work, discusses how the research questions and hypotheses were addressed, and outlines the limitations of the proposed framework alongside directions for future research.

Chapter 2

Literature Review

This chapter presents the systematic literature review conducted to identify, characterise, and critically assess the existing body of research on delay prediction in public transport. The chapter first describes the review protocol and the resulting screening process, following the PRISMA methodology, and then presents the state of the art derived from the documents retained for analysis, concluding with a synthesis of the main findings that motivate this dissertation.

2.1 Review Protocol

The literature review followed the PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) methodology. This approach was chosen because of its structured process to identify, screen, and select relevant documents in a reproducible way.

To demonstrate this process, a PRISMA flowchart was created (Figure 2.1), presenting the number of records retrieved from each database, how many were removed as duplicates and excluded during screening, and the final number of documents included for analysis.

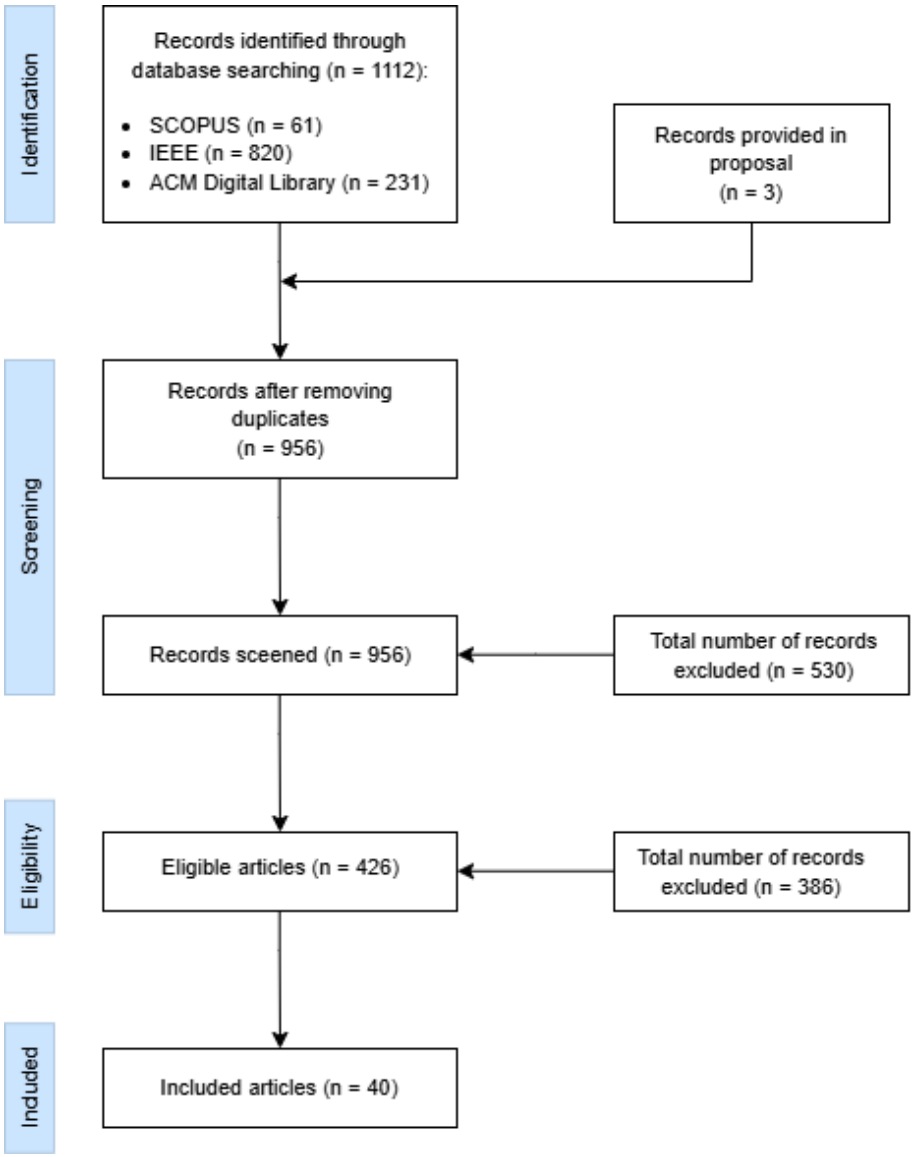


Figure 2.1: PRISMA flowchart

2.2 Identification Phase

The search was conducted in three digital libraries: Scopus, Institute of Electrical and Electronics Engineers (IEEE) Xplore, and Association for Computing Machinery (ACM) Digital Library.

To ensure a comprehensive overview of the topic, three search queries were defined, each addressing a specific aspect of the research problem and combining keywords related to public transport, machine learning, multi-modality, and intelligent transport systems. To ensure a systematic and replicable search process, the same queries were applied across all databases, searching the article title, abstract, and keywords fields.

2.2.1 Search Queries

The following queries were created to retrieve the most relevant documents related to this research problem:

Q1. Delay Prediction in Public Transport using Machine Learning

Focused on studies addressing delay or arrival-time prediction in buses, metro, or train systems through data-driven or machine-learning models.

Q2. Multi-Modal Journey Reliability and Delay Propagation

Focused on studies exploring delay propagation and reliability in multi-modal or intermodal transport networks, including how delays in one mode affect others.

Q3. Real-time Data and Intelligent Transport Systems

Focused on research applying predictive analytics and real-time data to improve public-transport reliability and smart mobility systems.

2.2.2 Search Queries Results

The results obtained from each query are presented in Table 2.1.

Digital Library	Q1	Q2	Q3	Total
Scopus	48	4	9	61
IEEE	715	1	104	820
ACM	111	10	110	231
Total	774	15	223	1112

Table 2.1: Summary of Search Queries

2.3 Screening Phase

A total of 1,112 documents were initially retrieved from the combined database searches across Scopus, IEEE Xplore, and the ACM Digital Library. After removing duplicate records, the remaining documents were filtered according to a set of criteria intended to ensure the relevance and quality of the selected documents:

- Documents must be written in English
- Documents must be published between the years 2020 and 2025
- Documents must be classified as an article
- Documents must provide full-text access

After applying all filters, a total of 426 documents remained for the eligibility phase.

2.4 Eligibility Phase

During the eligibility phase, the 426 documents obtained after filtering were analysed to determine their relevance to the research theme. First, the title was examined, then the abstract, and finally, where necessary, the introduction and conclusion sections of each document were reviewed. This approach allowed for a more accurate selection of documents relevant to the subject of this dissertation.

To be considered eligible, a document had to address topics such as the prediction of delays or travel times in public transport systems, mention multi-modal or intermodal networks, and apply data-driven or machine-learning approaches.

After applying these criteria, a total of 40 documents were obtained.

2.5 Included Documents

The included documents are listed in the References section (references [2] to [31] and [33] to [43]).

2.6 State of the Art

Tables 2.2 and 2.3 summarise the main studies selected for detailed analysis in this section.

2.6.1 Delay Prediction in Single-Mode Public Transport Systems

Several papers have explored deep learning models for predicting bus arrival times using historical and real-time data. In *A Bus Arrival Time Prediction Method Based on Position Calibration and LSTM* (2020), the authors propose an LSTM-based model reinforced with a position calibration mechanism to mitigate GPS noise and spatial inaccuracies commonly present in real-time vehicle tracking systems. By combining real-time vehicle positioning with historical trajectory data, the model achieves significantly lower prediction errors than traditional neural networks and statistical baselines. Despite its strong temporal modelling capability, the approach is limited to bus-only operations and does not consider external disruptions such as traffic incidents or interactions with other transport modes.

Similar LSTM-based approaches have been adopted to capture spatio-temporal dependencies in bus operations. For example, *Bus Arrival Time Prediction Based on LSTM and Spatial-Temporal Feature Vector* (2020) offers standard LSTM models that incorporate spatial correlations between neighbouring stops along with temporal travel patterns, leading to improved prediction accuracy under varying traffic conditions. Another representation of traffic dynamics is proposed in *Long and Short-Term Bus Arrival Time Prediction with Traffic Density Matrix* (2020), where traffic conditions surrounding a bus route are modelled using a spatio-temporal traffic density matrix. This matrix captures how congestion evolves over time and across adjacent road segments, allowing the model to jointly learn short-term disturbances and long-term congestion tendencies

that influence the variability of bus arrival. Although effective, these methods remain focused on single bus routes and do not account for propagated impacts on passenger transfers.

A different perspective is introduced in *A Novel Bus Arrival Time Prediction Method Based on Spatio-Temporal Flow Centrality Analysis and Deep Learning* (2022), where the authors model incoming and outgoing flow relationships through Bus Flow Centrality (BFC) metrics integrated into a deep learning framework. The model captures how congestion propagates across connected road segments, leading to improved prediction accuracy under irregular and peak-hour traffic conditions. Related studies, such as *Robust Real-Time Delay Predictions in a Network of High-Frequency Urban Buses* (2022), further emphasise the importance of modelling interactions between multiple bus routes operating within the same urban network. By accounting for correlated delays across high-frequency services sharing infrastructure and operating in close spatial proximity, these approaches improve robustness under fluctuating operational conditions. Nevertheless, these approaches remain limited to a single-mode bus network and do not evaluate how delays affect subsequent multimodal connections, such as transfers to metro or rail services.

The influence of traffic infrastructure is addressed in *A Prediction Model for Bus Arrival Time at Bus Stop Considering Signal Control and Surrounding Traffic Flow* (2020). This study integrates traffic signal timing parameters and surrounding flow density with bus trajectory data, demonstrating improved prediction accuracy at signalised intersections, where delays are often highly variable. By accounting for signal control effects, the model improves short-term arrival estimates in constrained urban environments. However, the approach operates at a local operational level, focusing on individual stops and intersections, and does not capture passenger impacts, such as missed connections and transfer reliability, which are critical for multimodal journey planning.

Deep learning has also been widely adopted for train delay prediction, particularly for modelling long-term temporal dependencies. In *A Hybrid LSTM–CPS Approach for Long-Term Prediction of Train Delays in Multivariate Time Series* (2023), the authors propose a hybrid architecture that combines LSTM networks with the Critical Point Search (CPS) method. This model integrates schedule information, historical delays, and operational variables, such as run-time and dwell time, demonstrating long-term prediction performance in large railway networks. Nevertheless, this framework is designed for train-only scenarios, disregarding interdependencies with other transport services.

Beyond deep learning, alternative modelling paradigms have also been explored. *Bus Arrival Time Prediction Using Wavelet Neural Network Trained by Improved Particle Swarm Optimization* (2020) combines wavelet neural networks with meta-heuristic optimisation to improve convergence and prediction accuracy, while *IoT-Based Bus Arrival Time Prediction Using Artificial Neural Networks* (2020) uses sensor-based data collection within smart transport systems. Although these approaches demonstrate the feasibility of different learning paradigms, they remain limited to single-mode bus prediction and lack integration with broader transport networks.

Finally, *Prediction Method of Train Delay Using Deep Learning Technique* (2022) explores Deep Neural Network (DNN) for capturing non-linear relationships between performance parameters and train delays. These parameters include metrics such as propagated delay, running time,

and dwell time. The results demonstrate improved performance over traditional regression-based models. Nevertheless, the evaluation is limited to train datasets, without modelling intermodal interactions, transfer reliability, or the propagation of disruptions, which are essential aspects of smart mobility systems.

2.6.2 Machine Learning Approaches for Travel Time and Delay Prediction

Several studies have explored deep learning and ensemble methods to improve prediction accuracy and robustness. In *Bus Arrival Time Prediction Based on the Optimized Long Short-Term Memory Neural Network Model With the Improved Whale Algorithm* (2024), the authors improve an LSTM model using an improved Whale Optimization Algorithm (WOA) to optimise hyperparameters, such as learning rate and network structure, which are known to affect LSTM performance. The model uses historical bus trajectory data, real-time GPS information, and temporal features, such as time of day and day of week. This approach improves convergence speed and prediction accuracy, particularly under varying traffic conditions. While effective, the method remains focused on bus-only arrival prediction, without incorporating network-level interactions or multimodal dependencies.

More recent studies have explored transformer-based architectures to improve long-range dependency modelling in bus arrival prediction. In *BAT-Transformer: Prediction of Bus Arrival Time with Transformer Encoder* (2024), the authors propose a Transformer-based encoder model that replaces recurrent structures with self-attention mechanisms. By allowing the model to focus on relevant temporal segments of historical bus trajectories, the approach improves prediction accuracy, particularly for long-term arrivals. Compared with LSTM-based baselines, the Transformer model demonstrates stronger robustness to irregular traffic patterns and varying service frequencies. However, the framework remains applied only to bus arrival prediction and does not consider interactions with other transport modes or transfer dependencies.

The study *Bus Dynamic Travel Time Prediction: Using a Deep Feature Extraction Framework Based on RNN and DNN* (2019) introduces a two-stage deep learning framework where Recurrent Neural Network (RNN) are used for temporal feature extraction, followed by Deep Neural Networks (DNNs) for travel time estimation. In the first stage, an RNN extracts temporal features from sequential bus travel data, capturing evolving patterns in vehicle movement; in the second stage, a DNN uses these extracted features to estimate future travel times. The results demonstrate improved accuracy over traditional methods, particularly in dynamic traffic environments, including weather-related disruptions and irregular congestion caused by incidents or roadworks. While the proposed framework improves robustness and prediction accuracy, it operates only on buses and does not consider multimodal connections within passenger journeys.

In *Data-Driven Train Delay Prediction Incorporating Dispatching Commands: An XGBoost-Metaheuristic Framework* (2023), the authors present a machine learning framework that improves train delay prediction by embedding dispatching control actions within the predictive model. The approach uses XGBoost as the main learner, with metaheuristic optimisation techniques used to

fine-tune hyperparameters and improve generalisation. Experimental results demonstrate that including dispatching information significantly improves predictive accuracy, particularly in disrupted scenarios. However, the study remains focused on railway operations, without extending predictions to passenger transfers or intermodal reliability.

The paper *Deep Encoder Cross Network for Estimated Time of Arrival* (2023) proposes a deep learning model designed to improve estimated arrival time prediction by better modelling interactions between input features. The model combines a deep encoder with a cross-network structure, allowing it to learn both complex patterns and direct relationships between variables such as location, speed, time, and contextual features. The framework is tested on large-scale datasets and demonstrates strong performance across different prediction scenarios, except for multimodal scenarios, which are not tested, remaining a mode-specific model that does not account for delay propagation between transport services.

In *Heterogeneous Machine Learning Ensembles for Predicting Train Delays* (2023), the authors investigate the use of ensemble learning to improve train delay prediction robustness. The proposed approach combines different types of machine learning models, including tree-based methods, neural networks, and linear predictors, each addressing a different aspect of the delay prediction problem. By aggregating predictions from diverse models, the ensemble reduces sensitivity to noise and improves generalisation across different operating conditions, such as varying traffic levels and transport disruption, showing that heterogeneous ensembles can outperform individual models in both accuracy and stability. Despite these advantages, the approach remains focused on predicting delays for a single mode, without considering multimodal interactions.

The paper *Prediction of Arrival Time of Pure Electric Bus Based on FA-BP Algorithm* (2023) proposes a hybrid prediction model combining a Backpropagation (BP) neural network with the Firefly Algorithm (FA) for parameter optimisation, incorporating vehicle trajectory data and time-of-day information. Experimental results show improved prediction accuracy compared with standard BP models. Nevertheless, the approach is limited to single-mode bus operations and does not explore interactions with other transport services.

In *Short-Term Arrival Delay Time Prediction in Freight Rail Operations Using Data-Driven Models* (2023), the authors address delay prediction in freight rail systems using machine learning techniques adjusted for short-term forecasting. The proposed models integrate real-time operational data, including train movement logs, track occupancy, and dispatching information, to predict arrival delays. Unlike passenger rail studies, this work focuses on the unique characteristics of freight operations, such as variable train lengths, cargo priorities, and scheduling constraints. The results demonstrate that data-driven models can effectively capture short-term delays in freight corridors, although the study remains limited to freight rail operations, without considering shared infrastructure with other transport modes.

The study *Short-Term Bus Travel Time Prediction for Transfer Synchronization with Intelligent Uncertainty Handling* (2023) proposes a prediction framework designed to support transfer coordination in public transport systems, combining data-driven travel time prediction with uncertainty modelling to enable more reliable synchronisation between connecting services. A major contri-

bution of this work is the treatment of prediction uncertainty, which is critical when coordinating transfers under fluctuating traffic conditions. However, the model is applied within bus-centric scenarios and does not extend to multimodal networks involving rail or metro services.

In *Spatio-Temporal Forecasting of Bus Arrival Times Using Context-Aware Deep Learning Models in Urban Transit Systems* (2025), the authors propose a deep learning framework that integrates spatial, temporal, and contextual information, including traffic conditions, weather, and route topology, to improve bus arrival time prediction. By modelling spatial correlations between stops and the temporal evolution of traffic states, the approach achieves improved prediction accuracy under both normal and congested conditions. Despite its comprehensive feature integration, the study remains focused only on bus networks and does not analyse delay propagation across different transport modes.

The paper *Towards Forecasting Bus Arrival Through a Model Based on GNN+LSTM Using GTFS and Real-Time Data* (2024) introduces a hybrid architecture combining Graph Neural Networks (GNNs) and LSTM networks. The GNN component models the spatial structure of the bus network using GTFS data, while the LSTM models temporal dynamics using real-time vehicle data. This architecture allows the model to represent both network topology and time-dependent behaviour, resulting in improved arrival time predictions compared with traditional sequence-based models. However, the framework is applied exclusively to bus networks, without addressing intermodal connectivity or passenger transfers involving rail or metro services.

The study *XGBoost in Public Transportation for Multi-Attribute Data: Delay Prediction in Railway Systems in Real-Time* (2024) applies the XGBoost algorithm to predict train delays using high-dimensional, multi-attribute operational datasets, incorporating features such as schedule adherence, infrastructure status, weather conditions, and historical delay patterns. The results show that XGBoost effectively handles heterogeneous data sources and provides robust real-time predictions at relatively low computational cost. Nevertheless, the approach remains centred on railway systems, without extending predictions to passenger transfers or multimodal network interactions.

Network-wide robustness has also been addressed in *Robust Real-Time Delay Predictions in a Network of High-Frequency Urban Buses* (2022), which focuses on improving prediction stability under high-frequency operations, where small disturbances can rapidly propagate across services. While the study highlights the importance of network-level robustness, it remains limited to bus systems and does not evaluate cross-modal delay propagation or passenger transfer reliability.

In *Real-Time Passenger Train Delay Prediction Using Machine Learning: A Case Study With Amtrak Passenger Train Routes* (2022), the authors apply supervised learning models to real-time operational data from intercity passenger rail services, demonstrating that machine learning models can provide accurate real-time predictions for long trajectories. However, the approach focuses only on railway operations and does not consider interactions with other urban transit connections.

Graph-based deep learning models have also gained attention for capturing spatial dependencies in rail networks. In *Train Time Delay Prediction for High-Speed Train Dispatching Based on Spatio-Temporal Graph Convolutional Network* (2021), the authors propose a Spatio-Temporal

Graph Convolutional Network (STGCN) that models both the topological structure of the rail network and the evolution of temporal delay, achieving superior prediction accuracy compared with sequence-based models. Although the approach captures spatial dependencies within rail systems, it remains limited to a single transport mode and does not address intermodal delay interactions.

2.6.3 Passenger-Centric and Decision-Oriented Delay Prediction

While the majority of existing research focuses on predicting vehicle-level delays and arrival times, fewer studies consider how these predictions translate into passenger-level information and decision support. From a user perspective, delays are not only operational metrics but also directly affect transfer success, journey reliability, and overall travel experience. As a result, recent work has begun to shift from purely service-centric prediction towards passenger-centric modelling and decision-oriented frameworks.

In *Short-Term Bus Travel Time Prediction for Transfer Synchronization with Intelligent Uncertainty Handling* (2023), the authors propose a prediction framework designed to support transfer coordination in public transport systems. Instead of providing point estimates, the model incorporates uncertainty quantification, allowing planners and real-time systems to evaluate the likelihood of successful transfers under fluctuating traffic conditions. However, the framework remains limited to bus-based scenarios and does not address cross-modal transfers involving rail or metro services.

A passenger-oriented perspective is further explored in *When Did the Train Arrive? A Bayesian Approach to Enrich Timetable Information Using Smart Card Data* (2021), where the focus shifts from vehicle movement to passenger behaviour by using smart card data to infer actual arrival times experienced by travellers. Using Bayesian inference, the model reconstructs arrival time distributions when operational data are incomplete or noisy. Although the work does not directly predict future delays, it highlights the importance of passenger data for understanding the impacts of delays and improving the reliability of timetable information in complex transit systems.

Human-centric service design is also addressed in *Analysis on the Bus Arrival Time Prediction Model for Human-Centric Services Using Data Mining Techniques* (2022). The authors emphasise the role of arrival time prediction in improving passenger satisfaction by personalising information services to user needs. Although the predictive models themselves remain relatively simple, the work reinforces the idea that accuracy alone is insufficient if predictions are not aligned with passenger decision-making processes.

Together, these studies highlight a growing awareness that delay prediction should not be treated solely as an operational forecasting task. Instead, predictive models must be evaluated based on their ability to support passenger decisions, improve transfer reliability, and enhance journey-level experience. Nevertheless, existing approaches remain fragmented, often focusing on a single mode or a specific aspect of passenger interaction, without offering an integrated framework that connects multimodal delay prediction with decision-support mechanisms for transfer reliability and multimodal journey planning.

2.6.4 Multimodal and Intermodal Delay Propagation in Public Transport Systems

As urban public transport systems evolve into interconnected networks, the limitations of single-mode delay prediction models become more evident. In multimodal environments, delays rarely remain isolated within a single service; instead, they propagate across lines and modes, affecting transfer reliability and overall journey performance.

A possible way to capture delay propagation within transport networks is presented in *A Two-Stage Train Delay Prediction Method Based on Data Smoothing and Multimodel Fusion Using Asymmetry Features in Urban Rail Systems* (2022). This work proposes a two-stage framework that first applies data smoothing techniques to reduce noise and irregularities in train operation logs, before combining multiple predictive models through a multimodel fusion strategy that integrates asymmetry features, distinguishing between upstream and downstream delay propagation effects. Experimental results show that accounting for asymmetric propagation improves prediction accuracy compared with single-model and symmetric approaches. Although the approach remains limited to a single transport mode, it highlights the importance of modelling network dependencies, which becomes even more critical in multimodal systems.

A journey-oriented perspective is adopted in *Enhancing Travel Time Prediction for Intelligent Transportation Systems: A High-Resolution Origin–Destination-Based Approach with Multi-Dimensional Features* (2025). Instead of predicting delays or travel times on individual services, the authors model travel times between origin–destination (OD) pairs using high-resolution spatial, temporal, and contextual features, allowing the model to align more closely with the passenger's perception of travel, which is inherently journey-based rather than route-based. Although the study does not explicitly model transfer disruptions or cascading delays between transport modes, it represents a significant shift towards journey-level prediction, which is essential for reliable multimodal itinerary planning.

The role of contextual and heterogeneous data sources is further explored in *Exploiting Multimodal Contextual Sensing for City-Bus's Stay Location Characterization: Towards Sub-60 Seconds Accurate Arrival Time Prediction* (2022). This study uses multimodal sensing data, including environmental conditions, mobility patterns, and contextual signals, to improve the characterisation of bus stop dwell behaviour and arrival time prediction. Although the focus remains on bus operations, the use of multimodal contextual sensing illustrates how combining diverse data sources can significantly enhance prediction performance in complex urban environments.

2.6.5 Data Sources and Availability in Public Transport Delay Prediction

Most existing studies rely on a limited set of data sources, typically focused on historical operational records or real-time vehicle positioning within a single transport mode. In bus arrival and travel time prediction, the most commonly used data sources include GPS-based vehicle trajectories, scheduled timetables, and stop-level arrival records. These data sources are often supplemented with traffic-related variables such as traffic density and signal control information to improve the accuracy of short-term predictions [10, 20, 27, 29].

Several works further enrich bus-focused datasets by incorporating external contextual information, including weather conditions, passenger load, and IoT-based traffic measurements, demonstrating that such factors can significantly influence arrival time variability [12, 24, 32]. More recently, standardised data formats such as the General Transit Feed Specification (GTFS) have been used to combine static network descriptions with live vehicle updates, allowing network-aware modelling approaches based on graph representations [21].

In railway systems, delay prediction studies usually rely on operational logs, timetable adherence records, dwell times, and dispatch information. Deep learning and ensemble-based approaches exploit these datasets to capture complex temporal patterns and delay propagation [2, 36, 37, 40]. More advanced work integrates multi-attribute datasets, including infrastructure status and weather conditions, improving prediction accuracy under disrupted conditions [5, 8, 17]. In addition, passenger-centric data sources, such as Automatic Fare Collection (AFC), have been used to infer missing operational information and estimate travel time reliability from the user's perspective [19, 41].

Despite these advances, only a limited number of studies integrate heterogeneous data sources or attempt to combine multiple transport modes within a unified predictive framework. Context-aware approaches based on multimodal and spatio-temporal representations have shown effective results in bus systems by incorporating environmental, mobility, and contextual signals [15, 23]. However, these approaches remain limited to a single mode, and even when network-level or graph-based representations are adopted, they typically focus on a single network, such as bus routes or rail lines, without capturing interactions across transport modes [9, 18, 35].

In contrast, the availability of open and standardised datasets in cities such as the Netherlands, where the OpenOV GTFS feed and DATEXII traffic information are provided through Nationaal Dataportaal Wegverkeer (Dutch National Data Warehouse for Traffic Information) (NDW), creates advantageous conditions for overcoming these data limitations and enables the development of integrated, data-driven models for multimodal delay prediction and itinerary reliability analysis.

Table 2.4 presents the main focus, limitations, and related studies of the research themes addressed in this section.

2.6.6 Critical Discussion and Research Gap

These limitations reveal a clear research gap, since there is a lack of integrated, data-driven frameworks that combine real-time delay prediction across multiple transport modes with modelling of delay propagation and journey-level impacts. In particular, dense urban environments with tightly coupled bus, tram, metro, rail, and ferry services remain underexplored in the literature. Addressing this gap requires using heterogeneous open datasets, such as GTFS feeds, real-time vehicle data, and traffic condition reports, to predict not only individual service delays but also their propagated effects on multimodal itineraries.

Most existing approaches either focus on isolated vehicle-level predictions or model network dependencies within a single transport mode; few explicitly represent complete passenger journeys, transfer feasibility, missed connections, and accumulated passenger delay across interconnected modes.

This dissertation aims to address this gap by developing a context-aware machine learning framework for multimodal public transport delay prediction. The proposed approach integrates GTFS schedule data with DATEX II contextual disruption information and models passenger journeys as transfer-based chains of connected transport segments. By comparing XGBoost, LSTM, and GNN models, the work evaluates how different predictive paradigms capture delay propagation and journey-level reliability. By focusing on itinerary-level reliability and dynamic journey adaptation, the proposed approach seeks to move beyond static planning tools and single-mode prediction models, contributing towards passenger-centred intelligent transport systems capable of supporting delay forecasting, transfer reliability assessment, and future multimodal journey planning applications.

2.7 Main Findings from the Literature

This literature review demonstrates significant progress in the application of data-driven methods to delay prediction in public transport systems. Across both bus and rail domains, machine learning approaches, such as LSTM networks, graph-based models, ensemble learning techniques, and tree-based algorithms, have consistently shown improved predictive accuracy compared with traditional statistical methods. Deep learning architectures have proven particularly effective in capturing temporal dependencies and non-linear relationships in operational data, while graph-based approaches enable the modelling of spatial relationships within transport networks. Integration of contextual information, including traffic conditions and weather variables, has further improved prediction robustness under disrupted operational scenarios, and recent research has begun to recognise the importance of network-level and uncertainty-aware modelling, particularly in the context of transfer synchronisation and reliability analysis.

Despite these advances, three limitations remain evident across the reviewed studies. First, most existing studies focus on single-mode transport systems, treating bus, metro, and rail networks independently. Even when network-level representations are adopted, they are typically restricted to a single transport mode and do not account for cross-modal interactions. Second, while delay propagation within individual networks has been explored, few works model the cascading effects of disruptions across interconnected multimodal systems. Third, although some passenger-centric approaches exist, the majority of research remains prediction-oriented rather than decision-oriented, offering limited support for journey-level reliability assessment.

Taken together, these findings confirm the absence of integrated frameworks that combine multimodal delay prediction, propagation modelling, and journey-level reliability evaluation within a unified, passenger-centred system, providing the motivation and methodological direction for the framework proposed in this dissertation.

Table 2.2: Selected studies included in the State of the Art

ID	Title	Main Focus	Year
S01	Robust Real-Time Delay Predictions in a Network of High-Frequency Urban Buses	Network-level bus delay prediction	2022
S02	A Prediction Model for Bus Arrival Time at Bus Stop Considering Signal Control and Surrounding Traffic Flow	Infrastructure-aware bus prediction	2020
S03	A Hybrid LSTM–CPS Approach for Long-Term Prediction of Train Delays in Multivariate Time Series	Long-term train delay prediction	2023
S04	Bus Arrival Time Prediction Using Wavelet Neural Network Trained by Improved Particle Swarm Optimization	Optimized neural networks for bus arrival	2020
S05	IoT-Based Bus Arrival Time Prediction Using Artificial Neural Networks	IoT-driven bus arrival prediction	2020
S06	Prediction Method of Train Delay Using Deep Learning Technique	Deep learning for railway delay prediction	2022
S07	Bus Arrival Time Prediction Based on the Optimized Long Short-Term Memory Neural Network Model With the Improved Whale Algorithm	Metaheuristic-optimized LSTM for buses	2024
S08	BAT-Transformer: Prediction of Bus Arrival Time with Transformer Encoder	Transformer-based bus arrival prediction	2024
S09	Bus Dynamic Travel Time Prediction: Using a Deep Feature Extraction Framework Based on RNN and DNN	Deep feature learning for bus travel time	2019
S10	Data-Driven Train Delay Prediction Incorporating Dispatching Commands: An XGBoost-Metaheuristic Framework	Dispatch-aware railway delay prediction	2023
S11	Deep Encoder Cross Network for Estimated Time of Arrival	Feature interaction modeling for ETA	2023
S12	Heterogeneous Machine Learning Ensembles for Predicting Train Delays	Ensemble learning for rail delay robustness	2023
S13	Prediction of Arrival Time of Pure Electric Bus Based on FA-BP Algorithm	Hybrid neural optimization for bus ETA	2023
S14	Short-Term Arrival Delay Time Prediction in Freight Rail Operations Using Data-Driven Models	Short-term freight rail delay prediction	2023
S15	Short-Term Bus Travel Time Prediction for Transfer Synchronization with Intelligent Uncertainty Handling	Transfer-aware bus travel time prediction	2023

Table 2.3: Selected studies included in the State of the Art

ID	Title	Main Focus	Year
S16	Spatio-Temporal Forecasting of Bus Arrival Times Using Context-Aware Deep Learning Models in Urban Transit Systems	Context-aware bus arrival prediction	2025
S17	Towards Forecasting Bus Arrival Through a Model Based on GNN+LSTM Using General Transit Feed Specification (GTFS) and Real-Time Data	Graph-based bus arrival modeling	2024
S18	XGBoost in Public Transportation for Multi-Attribute Data: Delay Prediction in Railway Systems in Real-Time	Real-time railway delay prediction (XGBoost)	2024
S19	Real-Time Passenger Train Delay Prediction Using Machine Learning: A Case Study With Amtrak Routes	Real-time passenger rail delay prediction	2022
S20	Train Time Delay Prediction for High-Speed Train Dispatching Based on Spatio-Temporal Graph Convolutional Network	Spatio-temporal graph modeling for rail delays	2021
S21	When Did the Train Arrive? A Bayesian Approach Using Smart Card Data	Timetable enrichment using passenger data	2021
S22	Analysis on the Bus Arrival Time Prediction Model for Human-Centric Services Using Data Mining Techniques	Human-centric bus arrival prediction	2022
S23	A Two-Stage Train Delay Prediction Method Based on Data Smoothing and Multimodel Fusion Using Asymmetry Features in Urban Rail Systems	Delay propagation in urban rail networks	2022
S24	Enhancing Travel Time Prediction for Intelligent Transportation Systems: A High-Resolution Origin–Destination-Based Approach with Multi-Dimensional Features	Journey-level travel time prediction	2025
S25	Exploiting Multi-modal Contextual Sensing for City-Bus's Stay Location Characterization: Towards Sub-60 Seconds Accurate Arrival Time Prediction	Contextual sensing for bus arrival accuracy	2022
S26	A Bus Arrival Time Prediction Method Based on Position Calibration and LSTM	LSTM-based model for less prediction errors	2020
S27	Bus Arrival Time Prediction Based on LSTM and Spatial-Temporal Feature Vector	LSTM-based models with improved prediction accuracy under varying traffic conditions	2020
S28	Long and Short-Term Bus Arrival Time Prediction with Traffic Density Matrix	Traffic density matrix that predicts bus arrival time under variable conditions	2020
S29	A Novel Bus Arrival Time Prediction Method Based on Spatio-Temporal Flow Centrality Analysis and Deep Learning	Deep learning framework that captures how congestion propagates across connected road segments	2022

Table 2.4: Main research themes, focus, limitations, and associated studies

Research Theme	**Main Focus**	**Limitations**	**Studies**
Single-Mode Delay and Arrival Time Prediction	Predict delays for a single transport mode using historical and real-time data	No intermodal interactions	S01, S02, S03, S04, S05, S06, S13, S22, S26, S27, S28, S29
ML Architectures for Travel Time and Delay Prediction	Advanced ML, hybrid models, ensembles	Mode-specific, prediction-only	S01, S06, S07, S08, S09, S10, S11, S12, S13, S14, S15, S16, S17, S18, S19, S20, S28
Network-Aware and Propagation-Oriented Models	Model delay propagation within transport networks	Single-mode networks	S01, S17, S20, S23
Multimodal Perspectives	Origin-Destination-based prediction, transfer reliability	Partial coverage	S15, S23, S24, S25

Chapter 3

Methodological Approach

3.1 Proposed Framework

The proposed framework models and predicts delay propagation in multimodal transport networks, incorporating passenger transfers, intermodal connections, and external disruptions [6, 22, 28]. Unlike conventional stop-level delay prediction approaches, which estimate delays at successive stops within a single vehicle trip, the proposed methodology adopts a passenger-oriented perspective in which delays may propagate across multiple transport services, routes, and modes [3].

The framework was developed incrementally. An initial stop-level formulation was implemented to establish a baseline, based on GTFS operational schedules [15, 21], synthetic delay patterns, and DATEX II contextual information. This stage validated the preprocessing pipeline, the feature engineering strategy, and the predictive modelling procedures. Although effective at capturing delay evolution within individual trips, the stop-level formulation cannot represent passenger transfers or the cascading effects resulting from missed connections [28, 34]. To address this limitation, the framework was extended to a transfer-aware multimodal formulation.

The final methodology models passenger journeys as sequences of interconnected transport segments linked by scheduled transfer connections. In this representation, delays are no longer restricted to individual vehicle trips but can propagate through the network whenever late arrivals compromise the feasibility of future connections [19, 28]. The framework therefore captures both vehicle-level delay accumulation and passenger-level delay propagation.

As complete observations of multimodal delay propagation were unavailable, realistic delay patterns were simulated using preprocessed GTFS schedules and DATEX II contextual data. The resulting datasets were enriched through feature engineering and used to train and evaluate eXtreme Gradient Boosting (XGBoost) [5, 8], Long Short-Term Memory (LSTM) [36, 40], and Graph Neural Network (GNN) [6, 21, 35] models for both single-mode and multimodal delay prediction tasks.

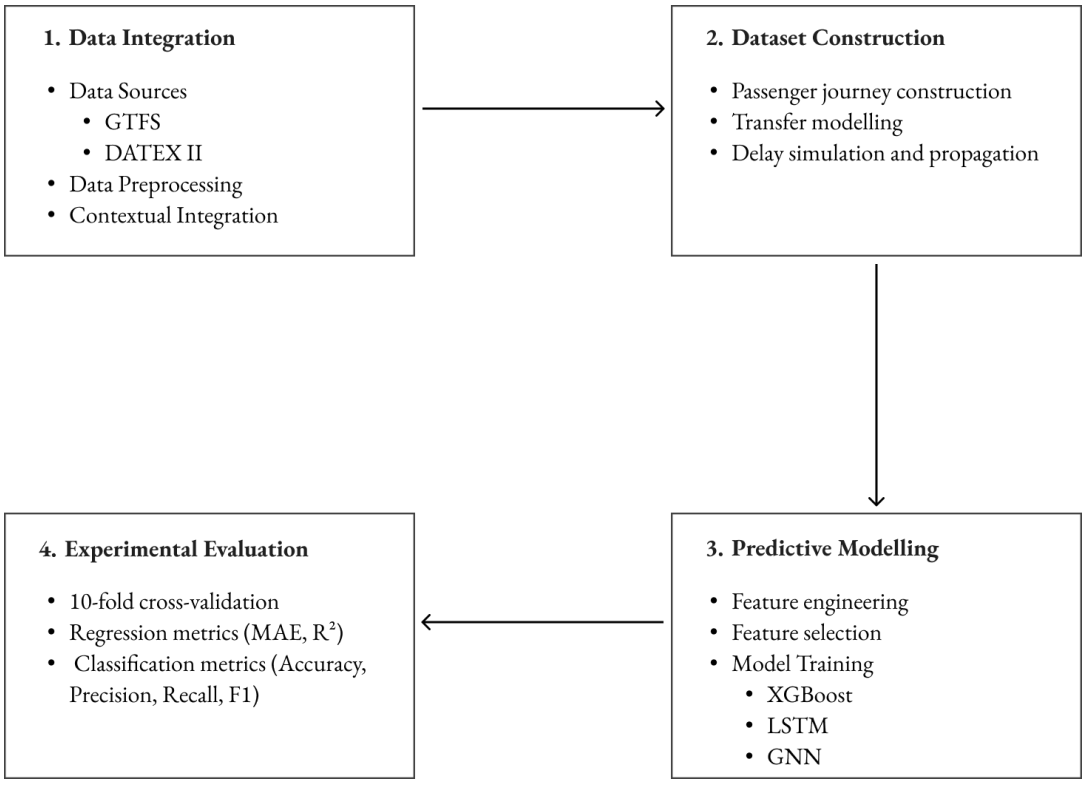


Figure 3.1: Overview of the proposed framework, from data integration and dataset construction to predictive modelling and experimental evaluation.

3.2 Problem Formulation

The proposed formulation models delay propagation in multimodal public transport journeys involving transfers between different transport modes. Whereas conventional stop-level approaches focus on delays within a single vehicle trip, this formulation considers passenger movements across interconnected services, where delays propagate through missed connections and transfer dependencies.

A passenger journey is represented as an ordered sequence of transport segments connected through transfer operations:

$$J = s_1, s_2, \ldots, s_n \tag{3.1}$$

where J denotes a passenger journey and each segment s_i corresponds to a specific transport route connecting two locations within the journey. Consecutive segments may belong to the same transport mode or involve intermodal transfers between bus, tram, metro, rail, and ferry services.

In contrast to the initial stop-level formulation, the multimodal formulation performs predictions at the transfer level, such that each observation corresponds to a transfer event within a passenger journey rather than an individual stop within a single vehicle trip.

Each transfer represents a potential propagation point at which operational disturbances may affect subsequent journey segments. Let a_i denote the actual arrival time of a passenger at a

transfer location and d_i the scheduled departure time of the intended connecting service. A missed connection occurs whenever:

$$a_i > d_i \tag{3.2}$$

In such cases, passengers must wait for the next available service, incurring an additional transfer penalty that contributes to the accumulation of delays [28] throughout the remainder of their journey.

Each transfer observation is described by the following feature vector:

$$x_i = [f_1, f_2, \ldots, f_m] \tag{3.3}$$

where x_i denotes the set of explanatory variables associated with transfer observation i and f_j represents an individual feature describing the operational state of the journey at the transfer location. The feature space integrates information from multiple sources, including:

- temporal characteristics [2, 9], such as journey start hour and peak-period indicators;
- transfer-related attributes [28], including planned waiting times and transfer penalties;
- operational vehicle delay information [5, 6] for arriving and connecting services;
- passenger delay propagation indicators derived from previously accumulated delays;
- multimodal journey descriptors, including the transport modes involved;
- contextual traffic disruption features [3, 8] extracted from DATEX II feeds.

The feature vector associated with each transfer observation captures the conditions under which a passenger attempts a connection, including the operational state of the arriving service, the characteristics of the intended subsequent connection, and the surrounding traffic context.

The primary prediction target is defined as the passenger delay following the transfer operation:

$$y_i = Delay_{i+1}^{passenger} \tag{3.4}$$

where y_i represents the cumulative delay experienced by a passenger after completing the transfer associated with observation i. This target incorporates both operational vehicle delays and additional delays resulting from missed connections, extended transfer waiting times, or propagated disruptions affecting subsequent journey segments.

The formulation assumes that passenger delays may increase through two complementary mechanisms. First, operational disturbances affect vehicle arrivals and departures, generating delays in individual transport services. Second, these vehicle-level delays influence transfer feasibility, potentially causing missed connections that introduce additional waiting times and further delay accumulation across the journey [6, 8]. Passenger delay propagation is therefore modelled as

a cumulative phenomenon in which operational disruptions can cascade across multiple transport segments and modes [22, 28].

Since complete observations of passenger-level delay propagation were unavailable for the full transport network, a simulation approach was employed to generate realistic operational delay patterns while preserving transfer dependencies and multimodal propagation behaviour [2, 27]. Vehicle delays were generated using a cumulative stochastic process:

$$delay_t = delay_{t-1} + trend_t + noise_t \tag{3.5}$$

where $delay_{t-1}$ represents the previously accumulated operational delay, $trend_t$ models systematic influences such as recurrent congestion effects, and $noise_t$ captures stochastic variability through Gaussian perturbations [9].

The generated vehicle delays were propagated throughout passenger journeys by evaluating whether each planned connection could be successfully reached. Whenever a delayed arrival caused a passenger to miss the intended service, an additional waiting time was introduced for the next available connection. This waiting time was modelled as a transfer penalty and incorporated into the cumulative passenger delay [28]. The passenger delay propagation process can be represented as:

$$Delay_i^{passenger} = Delay_{i-1}^{passenger} + Delay_i^{vehicle} + Penalty_i^{transfer} \tag{3.6}$$

where $Delay_i^{vehicle}$ denotes the delay in the transport service associated with transfer event i, and $Penalty_i^{transfer}$ corresponds to the additional delay caused by missed connections and extended transfer waiting times [3].

Although the primary prediction task was formulated as a regression problem, a complementary categorical evaluation was conducted to facilitate the operational interpretation of results. Predicted delays were grouped into service reliability categories representing distinct operational conditions, such as on-time and delayed services. This analysis enables model performance to be assessed not only in terms of numerical prediction accuracy but also in terms of its ability to identify operationally relevant delay situations [19, 40].

3.3 Research Context and Data Sources

The research was initially conceived around the Porto metropolitan area, owing to its dense and well-integrated public transport network, combining bus services operated by Sociedade de Transportes Colectivos do Porto (STCP), the Metro do Porto, and municipal traffic data. However, the predictive modelling of delay propagation requires a network with extensive multimodal coverage and a sufficiently long and complete operational schedule, including synchronised data describing several interconnected transport modes. Real-time delay records meeting these requirements were not consistently available for the Porto network, and the proposed methodology instead relies on

the simulation of realistic delay patterns over scheduled operations, for which a richer and more extensive multimodal feed was required.

For this reason, the case study was conducted using the Netherlands OpenOV GTFS feed, distributed through the Dutch National Data Warehouse for Traffic Information (NDW) [39], which provides a national-scale, multimodal public transport network comprising bus, tram, metro, rail, and ferry services. This dataset was complemented with DATEX II road closure and traffic event data [25], also obtained from NDW, providing contextual information on infrastructure disruptions and operational traffic conditions. Together, these sources offer a representative and sufficiently dense multimodal environment for studying delay propagation across transfer-dependent passenger journeys, while remaining publicly available and reflective of real-world operational constraints. The detailed characterisation of these data sources is presented in Section 3.3 of Chapter 3.

3.4 Data Collection

The proposed delay prediction framework relies on the integration of heterogeneous data sources describing different aspects of public transport operations and surrounding traffic conditions. Combining multiple data sources enables a richer representation of the factors influencing delay propagation in urban transport systems [15, 21].

Three primary datasets were used in this study:

- General Transit Feed Specification (GTFS) data describing the planned operation of the Dutch public transport network;
- DATEX II road closure data describing temporary road restrictions and infrastructure disruptions;
- DATEX II traffic event data containing operational traffic management and disruption information.

The GTFS dataset constitutes the core data source of the framework, providing the temporal, operational, and spatial information required to reconstruct transport trips and model delay propagation. The DATEX II datasets complement the operational schedules by introducing contextual information on external disruptions that can affect network performance [5, 8]. Integrating these sources enabled the development of context-aware predictions that capture both internal transport dynamics and external disturbances.

3.4.1 GTFS Data

The main dataset used in this study was obtained from the Netherlands OpenOV General Transit Feed Specification (GTFS) repository, made available through the Dutch National Data Warehouse for Traffic Information (NDW). GTFS is an internationally adopted standard for publishing public

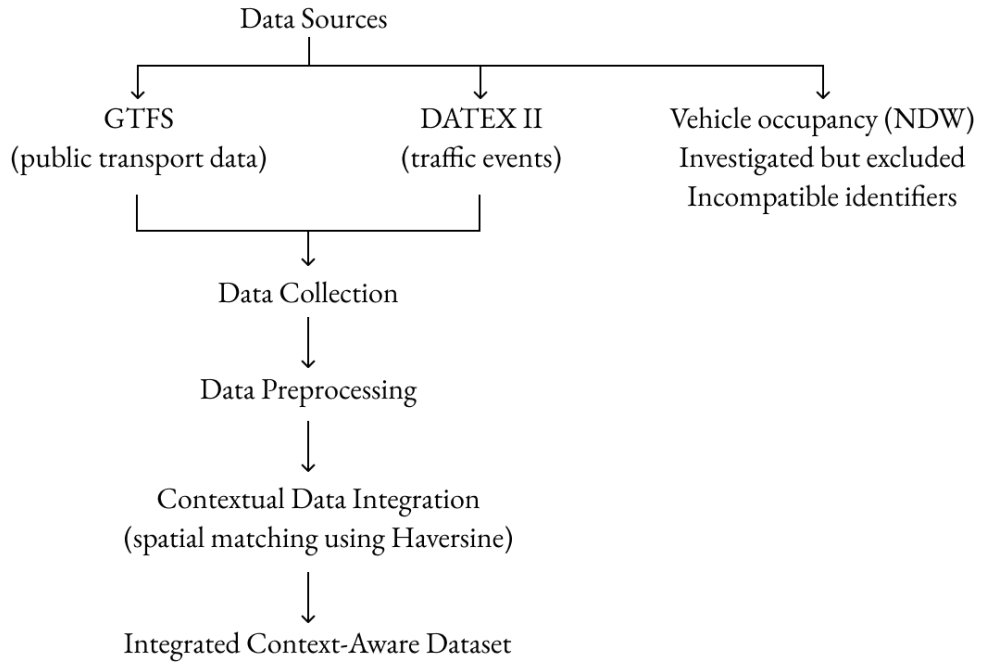


Figure 3.2: Data sources and integration pipeline, from raw GTFS and DATEX II feeds to the integrated context-aware dataset.

transport schedules and geographic information, enabling transport agencies to describe routes, trips, stop locations, service calendars, and timetable data in a structured, interoperable format [21, 34]. The GTFS feed used in this study corresponds to a timetable validity period between April and December 2026 and contains data from multiple public transport operators across the Netherlands, providing a comprehensive representation of the national transport network. The feed comprises several interconnected files describing different aspects of transport operations, as summarised in Table 3.1 [21].

Among the available files, particular importance was given to those required to model transport trajectories and operational schedules. The *stop_times.txt* file contains the planned arrival and departure times at each stop along a trip. The *trips.txt* file associates individual trips with route

Table 3.1: Main GTFS files available in the Netherlands OpenOV dataset.

GTFS Dataset Components		
File	**Records**	**Description**
agency.txt	38	Public transport operators participating in the feed
routes.txt	3,155	Route and line definitions
trips.txt	882,656	Individual service trips associated with routes
stop_times.txt	16,793,469	Scheduled arrival and departure times per stop
stops.txt	57,291	Geographic information for transport stops
transfers.txt	16,623	Transfer connections between transport services
calendar_dates.txt	302,131	Service availability and calendar exceptions
shapes.txt	6,843,534	Geometric representations of route trajectories

identifiers and operational services. The *routes.txt* file provides route-level information, while *stops.txt* contains the geographic coordinates necessary for spatial analysis and contextual data integration.

The *transfers.txt* file was also of particular importance, as it describes transfer connections between transport stops [21, 28]. Unlike the initial stop-level formulation, the multimodal framework explicitly models passenger transfers and intermodal connections, making transfer information a central component for reconstructing passenger journeys and analysing delay propagation across modes. The geographic structure provided by stop coordinates, route definitions, and transfer data is also essential for integrating external contextual information, including traffic disruptions and road closure events, which require spatial matching between transport stops and nearby incidents [21]. The GTFS dataset therefore constitutes the foundational layer of the framework, supporting the construction of multimodal transport journeys, transfer relationships, and delay propagation chains [21].

3.4.2 Road Closure Data

To incorporate the impact of infrastructure-related disruptions, a road closure dataset was collected from the Dutch National Data Warehouse for Traffic Information (NDW) [39]. The dataset is distributed through DATEX II feeds, a European standard for exchanging road traffic and transport information between traffic management centres, public authorities, and mobility service providers [25].

Road closure information provides valuable contextual knowledge of temporary restrictions that may affect vehicle circulation and, consequently, public transport operations. Such events can reduce road capacity, alter traffic flow patterns, generate congestion, and contribute to delays across the transport network [5, 27]. The original dataset was obtained as a DATEX II Extensible Markup Language (XML) feed describing temporary traffic management measures and road restrictions in the Dutch road network. Each record corresponds to a traffic situation and may contain one or more operational events associated with network disruptions. The feed includes several attributes characterising each closure event:

- event severity;
- start and end timestamps;
- cause of disruption;
- road management measures applied;
- number of restricted and operational lanes;
- geographic location information.

A parsing procedure was implemented to convert the hierarchical XML structure into a tabular format suitable for machine learning [26]. During this process, spatial information was extracted

from point-based geographic references or approximated from linear geometries when only polyline representations were available [21].

A filtering step was subsequently applied to retain only events associated with road, highway, and lane management restrictions [39], removing unrelated traffic management records such as rerouting recommendations and informational notices. After preprocessing and filtering, the final road closure dataset comprised 248 valid closure events. Each event was represented by a unique identifier, a temporal validity period, disruption characteristics, and geographic coordinates, enabling subsequent spatial integration with transfer locations and passenger journey connections [21].

To further characterise the operational impact of each event, a closure intensity indicator was derived from the ratio of restricted to operational lanes [4]. This indicator approximates the severity of each closure and was incorporated into the feature engineering stage as a contextual variable for delay prediction. The resulting dataset provides a spatially referenced representation of infrastructure disruptions that may influence public transport performance and delay propagation dynamics [5, 27].

3.4.3 Traffic Event Data

A second contextual dataset was incorporated to capture operational traffic conditions that may indirectly influence public transport performance. This dataset was obtained from the Dutch National Data Warehouse for Traffic Information (NDW) through DATEX II situation feeds containing real-time traffic management and disruption information [25]. While the road closure dataset focuses on physical infrastructure restrictions, the traffic event dataset provides a broader representation of operational conditions by describing incidents, management measures, obstructions, and temporary traffic control actions occurring throughout the road network [38]. These events may affect vehicle circulation, increase travel times, and contribute to delay propagation in interconnected transport services [8, 27].

The original feed was distributed in DATEX II XML format, comprising a collection of traffic situations, each containing one or more operational records describing specific events. The dataset includes several categories of traffic-related disruptions:

- road management operations;
- speed management restrictions;
- vehicle obstructions;
- rerouting measures;
- general network management events.

Each event record contains temporal, operational, and spatial information, including:

- event severity;

- start and end timestamps;
- disruption causes and management actions;
- temporary speed restrictions;
- vehicle obstruction descriptions;
- rerouting information;
- geographic coordinates.

A dedicated parsing procedure was developed to convert the hierarchical DATEX II XML structure into a tabular dataset. Relevant event attributes were extracted and geographic locations were obtained from direct coordinate references or from linear geometries embedded within the feed. To facilitate subsequent analysis, traffic events were grouped into higher-level categories according to their operational characteristics, enabling the construction of aggregated traffic-related indicators during feature engineering [8, 33].

A heuristic severity weighting scheme was adopted based on the expected operational impact of each event category, as presented in Table 3.2. Events directly restricting traffic flow or obstructing vehicle movement were assigned higher weights, while broader management and informational events received lower values [4, 8].

Table 3.2: Traffic event severity weights used to compute the traffic disruption score.

Event Category	**Weight**
Road Management	1.0
Vehicle Obstruction	0.8
Rerouting	0.6
Speed Management	0.4
Network Management	0.3
Other	0.2

These weights were incorporated into the contextual integration process and aggregated to produce traffic disruption indicators describing the severity of the traffic environment surrounding transport operations [27].

After preprocessing, cleaning, and event classification, the final traffic dataset comprised 601 valid events distributed throughout the Dutch transport network. Each event was associated with a temporal validity period and geographic location, enabling spatial integration with GTFS transport data. Together with the road closure dataset, this source provides a complementary contextual layer describing the operational traffic conditions that may influence public transport reliability and delay propagation in multimodal transport systems [15, 27].

3.4.4 Vehicle Occupancy Data

As part of the effort to enrich the framework with additional contextual information, a vehicle occupancy dataset was investigated. The dataset was obtained from the Dutch National Data

Warehouse for Traffic Information (NDW) [25] and contained passenger occupancy measurements collected from public transport services [7].

Vehicle occupancy represents a potentially relevant source of information for delay prediction, as crowded vehicles may experience longer boarding and disembarking times and increased dwell times at stops, increasing the probability of delay propagation [3, 9, 34, 36]. Occupancy information was therefore initially considered as a candidate contextual feature within the proposed framework.

The dataset contains operational records describing passenger occupancy levels for specific transport services, including attributes such as route, journey, and stop identifiers, alongside occupancy measurements. To facilitate integration with the GTFS data, occupancy records were mapped to a common representation using route, trip, and stop identifiers, and an exploratory integration process was conducted to establish correspondences between the two datasets. Several matching strategies were evaluated, including direct associations based on shared identifiers. Additional analyses were performed at the operator level, using EBS as a representative transport operator to simplify the identifier matching process.

Despite these efforts, no reliable correspondence could be established between the two datasets. The route and journey identifiers used in the occupancy records were found to be incompatible with those in the GTFS representation, resulting in no meaningful overlap between the datasets. Consequently, the vehicle occupancy dataset was excluded from the final framework, as no robust mapping mechanism could be identified to associate occupancy measurements with specific GTFS trips and stops [19, 21].

3.5 Data Understanding

3.5.1 Multimodal Network Representation

To support analysis of the GTFS data and provide a clearer picture of the case study's spatial structure, a visual representation of the Dutch multimodal public transport network was created. The network was derived from the GTFS files by combining route, trip, shape, and stop information. Rather than connecting consecutive stops with straight-line segments, the official GTFS `shapes.txt` file was used to represent the actual geometry of each service. This gives a more realistic spatial depiction of the network and avoids artificial long-distance links between consecutive stops on express or variant services.

Figure 3.3 shows the resulting multimodal network, including bus, rail, metro, tram, and ferry services. Stops are shown as point features, while transport lines are represented through their corresponding GTFS shape geometries. This visualisation highlights the scale and density of the Dutch public transport system and provides an initial overview of the multimodal structure considered in this work.

An interactive visualisation tool was also developed to inspect the multimodal network derived from the GTFS data. The preprocessing script exports the relevant network information to



Figure 3.3: Spatial representation of the Dutch multimodal public transport network derived from GTFS shapes and stops.

a JavaScript Object Notation (JSON) structure containing stop coordinates, route metadata, transport modes, and GTFS shape geometries. This file is then rendered through a Leaflet-based web interface [1]. The interface allows routes to be filtered by transport mode, and individual routes to be selected directly from the map or from the side panel. Stops can also be displayed with their associated modes and routes, making it possible to identify multimodal stops where more than one transport mode is available. Where available, route colours were taken from the GTFS `route_color` field, preserving the operator-defined visual identity of each service. When this field was missing or empty, a mode-based fallback colour was assigned to keep visual consistency across transport modes. This tool was used mainly for data inspection, validation, and qualitative interpretation of the network's multimodal structure.

3.6 Dataset Construction

The dataset was constructed in two stages to validate the methodology before scaling to multimodal complexity. In the first stage, a stop-level dataset was developed for exploratory analysis and baseline validation. In the second stage, a multimodal transfer-based dataset was constructed to model passenger delay propagation across interconnected transport services [21, 22].

3.6.1 Initial Single-mode Stop-Level Dataset Construction

The first stage modelled delay propagation within individual transport services. This formulation was designed to establish a baseline prediction framework capable of representing the sequential

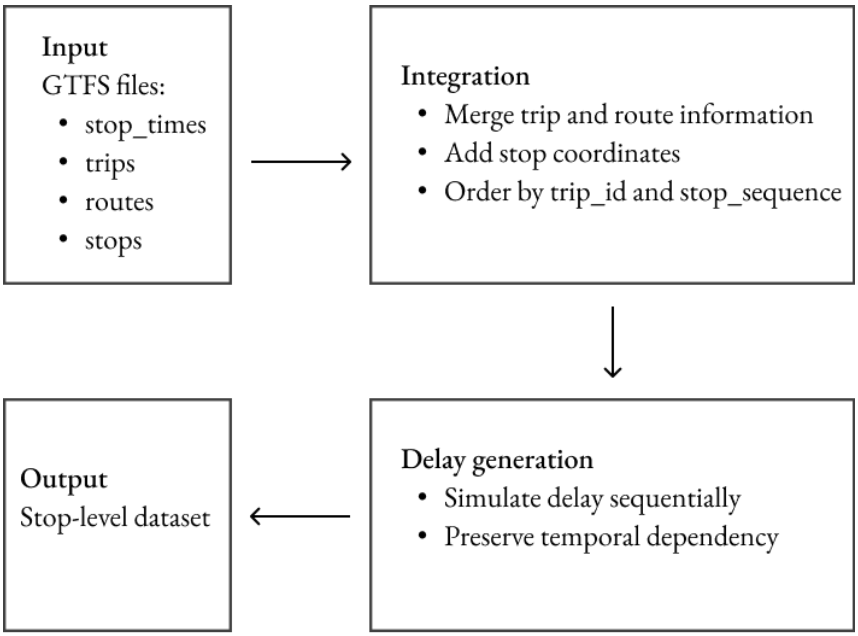


Figure 3.4: Construction pipeline of the initial single-mode, stop-level dataset.

evolution of delays along a single trip [2, 8, 22] while validating the data integration and preprocessing procedures [24] required for subsequent stages of the research.

The dataset was constructed using the GTFS *stop_times.txt*, *trips.txt*, *routes.txt*, and *stops.txt* files [21]. Stop-level timetable records were merged with trip and route information to reconstruct complete transport trajectories [21], and the resulting observations were ordered by trip identifier and stop-sequence position to preserve the temporal structure of each service [15].

In this formulation, each observation corresponds to a stop event within a specific transport trip, and the objective is to predict the delay at the subsequent stop based on the operational state of the service at the current stop [6, 8]. The dataset therefore represents delay propagation as a sequential process unfolding along a single vehicle trajectory.

The stop-level representation was adopted at this stage because it provides a simpler and computationally efficient environment for analysing delay behaviour. The primary objective was not to model complete passenger journeys, but to understand how delays evolve throughout a trip and to assess the predictive relevance of different operational and contextual variables. To this end, a synthetic delay generation process was applied to the GTFS timetable data [27]. Delays were propagated sequentially along each trip, producing realistic accumulation patterns [34] while preserving the temporal order of stop events. The simulation methodology is described in detail in Section 3.7.

The resulting stop-level dataset provided the baseline experimental environment used to validate the preprocessing pipeline, feature engineering procedures, and predictive models before extending the framework to multimodal passenger delay propagation.

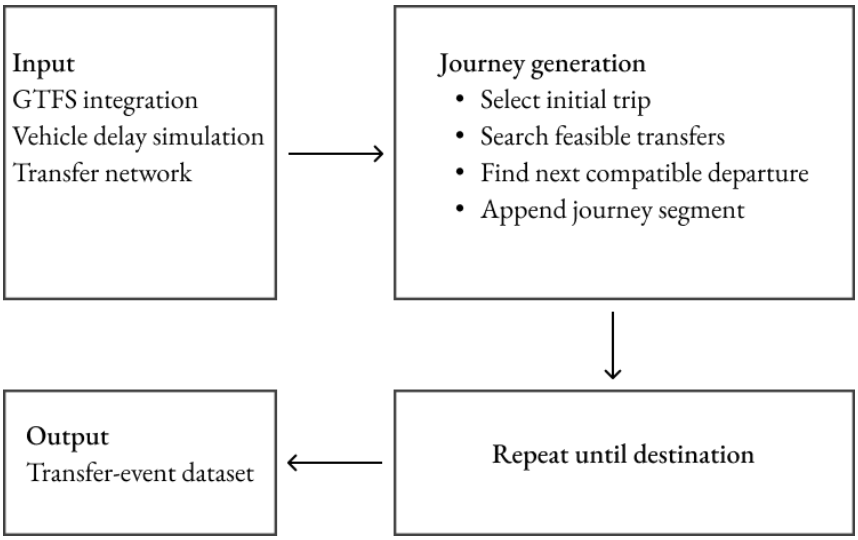


Figure 3.5: Construction pipeline of the multimodal, transfer-based dataset.

3.6.2 Multimodal Transfer-Based Dataset Construction

Unlike the stop-level dataset, which models delay evolution within individual trips, the multimodal formulation represents passenger journeys as sequences of transport segments connected by transfer events [28]. The prediction unit is therefore shifted from individual stop observations to transfer events occurring within multimodal passenger journeys [3, 19]. This formulation captures delay propagation mechanisms that emerge when operational disturbances compromise the feasibility of planned connections [28] and consequently affect subsequent journey segments.

The construction process relied on the complete GTFS transport network, including scheduled stop times, routes, trips, stops, and transfer information [21]. Transfer relationships were initially extracted from the GTFS `transfers.txt` file, which provides explicit connection information between transport stops, including the minimum transfer time required to move between services. However, relying exclusively on these predefined connections would substantially limit network connectivity. Additional transfer relationships were therefore generated through spatial inference, connecting nearby stops whose geographic proximity suggested a feasible walking transfer. The resulting transfer network combines explicit GTFS connections and spatially inferred transfers, producing a richer representation of multimodal accessibility throughout the transport system [21, 28].

Using this transfer network, synthetic passenger journeys were constructed by traversing sequences of connected transport services. Each journey consists of multiple travel segments linked through transfer operations and may involve single-mode or multimodal travel patterns. The generated journeys therefore include a variety of intermodal combinations, including bus-to-bus, rail-to-rail, bus-to-metro, metro-to-tram, tram-to-rail, and other transitions observed within the network [9, 28].

For each journey, the scheduled arrival and departure times were used to identify the intended transfer connections. The vehicle delays generated through the simulation process [27] were then

propagated throughout each journey to evaluate whether passengers could successfully reach their planned connections. Whenever a delayed arrival occurred after the scheduled departure of the intended connecting service, the connection was considered missed and the passenger was reassigned to the next available departure. This mechanism captures one of the most significant delay propagation phenomena in multimodal transport systems: the escalation of relatively small vehicle delays into substantially larger passenger delays as a result of missed connections and additional waiting times [28].

Each observation in the resulting dataset corresponds to a transfer event within a passenger journey and contains information describing:

- the current transport segment;
- the intended transfer operation;
- the characteristics of the connecting service;
- the accumulated passenger delay prior to the transfer;
- the delay state of the vehicles involved;
- the transfer feasibility conditions;
- potential transfer penalties arising from missed connections;
- the resulting passenger delay after the transfer.

The final dataset therefore represents passenger-centred delay propagation rather than vehicle-centred delay evolution [3, 19]. This distinction is particularly relevant because the primary impact of operational disruptions is experienced by passengers through increased travel times, missed connections, and reduced service reliability [2, 28], not through vehicle delays alone. By explicitly modelling transfer operations and multimodal interactions, the dataset provides a suitable foundation for predicting propagated delays across complex public transport journeys and constitutes the basis for the machine learning experiments presented in this dissertation.

3.7 Delay Simulation

3.7.1 Vehicle Delay Simulation

The proposed framework aims to model delay propagation in multimodal passenger journeys. Designing such a dataset requires information on the operational delays experienced by transport services across the network. GTFS provides detailed scheduled data describing trips, routes, stop sequences, and planned arrival and departure times [21], but it does not include synchronised real-time delay observations. Without these, delay propagation cannot be analysed consistently across transport modes and transfer operations. A delay simulation mechanism was therefore introduced to generate realistic operational delay patterns while preserving temporal dependencies [27, 37].

The simulation was designed to produce vehicle delay trajectories that reproduce three properties commonly observed in real operations: cumulative delay growth, temporal dependency between consecutive stops, and propagation effects [6, 8] that may later affect passenger transfers. The process assumes that delays evolve sequentially along a vehicle's trip: a disturbance at one stop may influence later stops, producing cumulative effects that propagate through the service. This reflects typical public transport dynamics, in which delays rarely occur in isolation and instead tend to accumulate due to congestion, operational disruptions, passenger boarding activity, traffic conditions, and limited scheduling recovery [34, 36].

For each vehicle trip, delays were generated through a cumulative stochastic process, defined as:

$$Delay_t^{vehicle} = Delay_{t-1}^{vehicle} + Trend_t + Noise_t \tag{3.7}$$

where:

- $Delay_{t-1}^{vehicle}$ is the delay accumulated at the previous stop;
- $Trend_t$ represents systematic operational influences affecting service performance;
- $Noise_t$ represents stochastic variability introduced through Gaussian perturbations [9, 27].

The trend component reflects the operating conditions typically associated with different periods of the day. Higher trend values were assigned during peak periods, between 07:00 and 09:00 and between 17:00 and 19:00, to simulate the increased probability of congestion and operational disruption during morning and evening commuting hours [3, 4, 13]. Outside these periods, lower trend values were applied to represent more stable conditions. The stochastic component, generated through Gaussian noise, introduces variability between successive observations and prevents the simulated delays from evolving deterministically. This allows the generated delay trajectories to reproduce the uncertainty and randomness typical of real transport operations, while maintaining temporal continuity between consecutive stops.

As delays accumulate recursively along each trip, the resulting delay patterns display several properties relevant to operational transport systems: temporal dependency, cumulative growth, service variability, and cascading propagation effects [22]. These properties provide a suitable basis for studying how vehicle-level disruptions propagate through transfer operations and ultimately affect passenger travel time.

The generated vehicle delays form the first stage of this propagation framework. In the next stage, these delays are transferred to passenger journeys, where missed connections and additional waiting time [28] can further increase the overall disruption experienced by travellers.

Although this formulation is intentionally simple, it follows the same structure adopted in several simulation-based delay generation studies, which combine a systematic trend component with a stochastic disturbance term to approximate real operational variability [9, 27]. This design choice was deliberate: a more complex generative process (for example, one incorporating

autoregressive dependencies or mode-specific disturbance distributions) would improve statistical realism at the cost of interpretability and control over the ground-truth propagation mechanism, which is essential for validating the multimodal delay propagation framework proposed in this work. The trend and noise parameters were calibrated to produce delay magnitudes and variability consistent with the ranges reported in delay prediction studies [3, 4, 13], rather than being fitted directly to observed Dutch operational data, which was not available at the required granularity. The main limitation of this approach is therefore not the additive structure itself, but the absence of empirical calibration against real delay distributions, a point discussed further in Section 6.5.

3.7.2 Passenger Delay Propagation

Vehicle delays provide an important representation of service performance, but they do not fully capture the impact of disruptions on passenger journeys [2, 28]. In multimodal networks, travellers often rely on transfers to reach their final destination, so the consequences of a delay can extend well beyond the affected vehicle. The proposed framework models passenger delay propagation by evaluating the feasibility of planned transfers under disrupted operating conditions. While the stop-level formulation analyses delay evolution within individual vehicle trips, the multimodal formulation considers whether passengers are able to reach their subsequent connections [28].

For each transfer event, the passenger's actual arrival time at the transfer location is obtained from the planned arrival time of the incoming service and its associated delay:

$$PassengerArrival_i^{actual} = VehicleArrival_i^{planned} + Delay_i^{vehicle} \tag{3.8}$$

where $VehicleArrival_i^{planned}$ is the scheduled arrival time of the incoming service at the transfer location, and $Delay_i^{vehicle}$ is the accumulated delay of that service [2, 9].

After reaching the transfer location, passengers require a minimum amount of time to board the connecting service. This minimum transfer time is the time needed to move from the arriving service to the connecting one and become available for the next boarding opportunity [28]. Transfer times were obtained from two sources. Where available, explicit transfer relationships from the GTFS *transfers.txt* file were used [21]. The *min_transfer_time* attribute gives the minimum transfer duration in seconds, converted to minutes for consistency with the other temporal variables (for example, 120 seconds corresponds to 2 minutes). Where no explicit GTFS transfer existed but a route or mode change was possible, additional spatial transfer relationships were inferred between nearby stops, with the minimum transfer time estimated from the geographic distance between stops and an assumed walking speed. A lower bound of 2 minutes was imposed on these estimated times, to avoid unrealistically optimistic values resulting from very short transfers. Each transfer event is therefore associated with a minimum transfer time, which determines the earliest moment at which a passenger can become available to board the intended connecting service. This value was used to compute the passenger's readiness time:

$$PassengerReady_i = PassengerArrival_i^{actual} + TransferTime_i \tag{3.9}$$

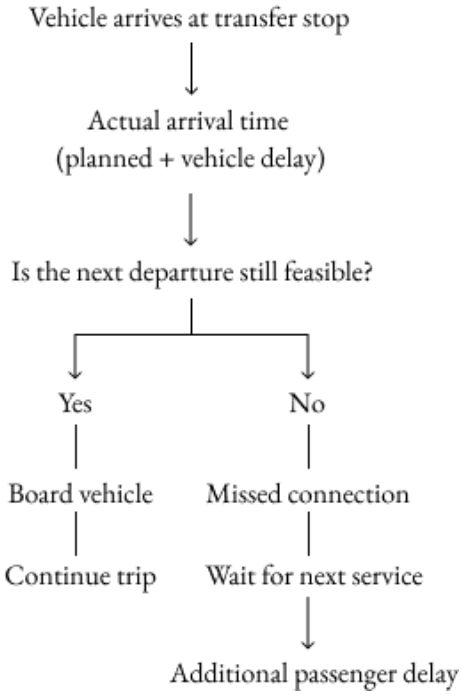


Figure 3.6: Transfer feasibility and passenger delay propagation process at a transfer stop.

where $TransferTime_i$ is the minimum time required to complete transfer event (i), either taken directly from the GTFS transfer definitions or estimated from spatial proximity for inferred walking transfers [21, 28].

The feasibility of the planned connection is evaluated by comparing the passenger's readiness time with the scheduled departure time of the intended connecting service. A transfer is considered unsuccessful when [19, 28]:

$$PassengerReady_i > Departure_i^{planned} \tag{3.10}$$

In this case, the passenger cannot board the intended service and must wait for the next available departure. The additional waiting time caused by the missed connection is modelled as a transfer penalty [3, 28]:

$$TransferPenalty_i = Departure_i^{next} - Departure_i^{planned} \tag{3.11}$$

where $Departure_i^{next}$ is the departure time of the next available service after the passenger becomes ready to board, and $Departure_i^{planned}$ is the departure time of the originally intended connection.

The transfer penalty is then added to the passenger's delay, giving the propagated delay after the transfer:

$$Delay_{i+1}^{passenger} = Delay_i^{vehicle} + TransferPenalty_i \tag{3.12}$$

where $Delay_i^{vehicle}$ is the delay of the vehicle arriving at the transfer location, and $TransferPenalty_i$ is the additional waiting time caused by a missed connection [28]. Passenger delay propagation therefore depends not only on vehicle operational performance, but also on whether transfer operations are completed successfully throughout the journey. By explicitly modelling transfer feasibility, missed connections, and transfer penalties, the proposed framework captures the cascading delay propagation that characterises multimodal public transport systems [6, 22].

3.8 Data Preprocessing

Following dataset construction, several preprocessing procedures were applied to ensure data consistency, temporal validity, and suitability for predictive modelling. As the research evolved from an initial stop-level, single-mode delay prediction formulation to a multimodal, transfer-based delay propagation framework, preprocessing was carried out at both stages. The aim was to remove invalid observations, standardise temporal information, handle missing values, validate the generated delay variables, and ensure that all datasets were ready for feature engineering and predictive modelling.

3.8.1 Initial GTFS Dataset Preprocessing

The first preprocessing stage was applied to the GTFS stop-level dataset used in the preliminary delay prediction experiments. After merging the *stop_times.txt*, *trips.txt*, *routes.txt*, and *stops.txt* files [21], several cleaning steps were carried out. GTFS schedules can contain extended-hour representations beyond 24 hours, particularly for services operating after midnight. All arrival times were therefore normalised to valid time representations before being converted to standard timestamp format [6]. Records with missing temporal information (arrival times) or missing stop coordinates were removed, to ensure temporal and spatial consistency. Stop coordinates, obtained from the GTFS *stops.txt* file, were associated with each operational observation, representing a vehicle's arrival at a specific stop within a trip. Temporal consistency was further checked by examining the chronological order of stops within each trip: for every pair of consecutive stops, the difference between their scheduled arrival times was computed, and observations producing negative intervals, indicating that a vehicle would appear to arrive at a later stop before reaching the previous one, were removed [6, 22]. Finally, duplicate records and non-essential attributes, such as *stop_headsign*, were removed, as they were not needed for delay modelling and provided redundant descriptive information. The resulting dataset is a temporally ordered and spatially consistent stop-level representation of transport operations, suitable for delay simulation and subsequent feature engineering.

3.8.2 Multimodal Dataset Preprocessing

The second preprocessing stage was applied after constructing the multimodal, transfer-based dataset. The generated transfer observations were first reviewed in terms of dimensionality, at-

tribute types, duplicate records, and missing values. Duplicate observations were removed, and invalid numerical values, including infinite values produced during intermediate calculations, were replaced with missing values [22, 24], to allow consistent validation and removal in later cleaning steps. Particular attention was given to the historical propagation variables: because the first transfer event in a passenger journey may have no prior delay history, the variables describing historical passenger and vehicle delay propagation were set to zero whenever no previous observations were available. This preserved the start of each passenger chain while keeping the dataset numerically consistent. Categorical attributes were converted to a consistent string format, to ensure uniform handling in the subsequent feature engineering and modelling stages. These attributes included the transport mode before and after each transfer (*from_mode* and *to_mode*), the corresponding mode transition (*mode_pair*, e.g., bus-to-rail or rail-to-tram), the transfer-type classification (*transfer_type*), and the set of transport modes used in the journey (*modes_used*). Missing categorical values were assigned to an *unknown* category [34]. Any remaining observations with incomplete information were then removed. Finally, several validation checks were carried out to confirm the consistency of the multimodal dataset, including the distribution of propagated passenger delays, transfer penalties, missed connections, and mode-pair combinations. The resulting dataset contains 50,000 transfer-level observations, with no duplicate records and no missing values, providing a consistent basis for contextual integration, feature engineering, and predictive modelling.

3.9 Contextual Data Integration

The DATEX II road closure and traffic event datasets provide external information on road network disruptions. However, these datasets do not share direct identifiers with the GTFS data, such as *trip_id*, *route_id*, or *stop_id*, so contextual information could not be incorporated through direct relational joins [21]. A spatial matching strategy was adopted instead, based on the assumption that disruptions located near the path followed by a transport service can affect vehicle delays and, in turn, passenger delay propagation. As both GTFS stops and DATEX II events contain geographic coordinates, spatial proximity was used as the common basis for linking the two datasets.

3.9.1 Stop-Level Contextual Integration

In the initial stop-level modelling stage, contextual information was linked to GTFS stops, since each observation in that dataset represented a vehicle's arrival at a stop. The stop location therefore provided a natural spatial reference for associating nearby road disruptions with transport operations. For each unique *stop_id*, the coordinates from the GTFS *stops.txt* file were compared with the coordinates of the parsed road closure and traffic event records. The distance between each stop and each contextual event was calculated using the Haversine formula [27]:

$$d = Haversine(lat_s, lon_s, lat_e, lon_e) \tag{3.13}$$

where (lat_s, lon_s) are the coordinates of a GTFS stop, and (lat_e, lon_e) are the coordinates of a contextual event.

Events within a 2 km radius were considered spatially relevant to the corresponding stop. This threshold balances the need to capture nearby disruptions that may affect local traffic and transport conditions against the risk of including events too far from the operation being modelled.

For each stop, road closure events were aggregated into three descriptors: an indicator of whether at least one closure occurred within the defined radius, the number of nearby closures, and the average closure intensity of nearby events [4], derived from the closure intensity metric calculated during DATEX II preprocessing.

Traffic events were aggregated in a similar way: an indicator and count of nearby events were created, along with an aggregated disruption score based on the event weights listed in Table 3.2 [8], and counters for specific event types such as speed management and vehicle obstruction.

The contextual matching procedure was performed once per unique stop, rather than separately for every row of the stop-level dataset. This choice was made for computational efficiency: several observations can correspond to the same physical stop, so repeating the same distance calculations for every occurrence would add unnecessary processing cost. Since all observations linked to the same stop share the same location, the contextual descriptors computed for that stop can be consistently assigned to every corresponding observation.

This strategy was used to build context-enriched stop-level datasets during the initial experimental stage. Several dataset configurations were designed to assess the contribution of external contextual information. A baseline configuration used only GTFS operational data. A second configuration added road closure features, allowing the effect of infrastructure-related disruptions to be assessed in isolation. A third configuration combined both road closure and traffic event features, providing a richer representation of the surrounding traffic environment. These configurations supported an analysis of whether external contextual information improved delay prediction, before the methodology was extended to the final multimodal formulation.

3.9.2 Transfer-Level Contextual Integration

In the final multimodal formulation, contextual information was integrated at the segment level, rather than only at the stop or transfer location. This change aligns the contextual variables more closely with the passenger delay propagation mechanism modelled in the final dataset. Each observation in the multimodal dataset corresponds to a transfer event during a passenger journey, but the delay observed at a transfer depends not only on conditions at the transfer location, but also on disruptions encountered by the vehicle during the segment travelled before reaching that transfer. Contextual exposure was therefore modelled over the stops traversed in the current journey segment.

During construction of the multimodal dataset, an auxiliary segment-level dataset was generated, describing each transport segment of a passenger journey, including the service used, the

boarding and disembarking stops, the transport mode, and the actual trip taken. For each transfer observation, the corresponding segment was identified by matching the *chain_id* and *transfer_index* of the final dataset with the *chain_id* and *segment_index* of the segment dataset. This identifies the segment that carries the passenger to the transfer being evaluated. The stops traversed during that segment were reconstructed using the *actual_trip_id*, *board_stop_id*, and *disembark_stop_id* values: the *actual_trip_id* identifies the service used by the passenger for that segment, while *board_stop_id* and *disembark_stop_id* give the stops where the passenger boards and leaves the service. The GTFS *stop_times.txt* file was used to retrieve the ordered stop sequence for the corresponding trip [21], from which the subset of stops between the boarding and disembarking stops was extracted, representing the part of the route actually travelled by the passenger.

Once the traversed stops were reconstructed, contextual events were linked to the segment through spatial matching. For each stop in the segment, nearby road closures and traffic events were identified using the same Haversine distance [21, 27] and 2 km threshold as in the stop-level integration. Events detected across all stops in the segment were aggregated using a set union, ensuring that the same event was counted only once even if it was near several stops in the same segment. Road closure information was aggregated into segment-level descriptors indicating whether the segment was exposed to at least one nearby closure, the number of distinct nearby closures, and the average closure intensity of the detected closures [4]. Traffic event information was aggregated into descriptors indicating exposure to nearby traffic events, the number of distinct events, the sum of their disruption weights, and the number of speed management and vehicle obstruction events. The resulting descriptors were merged into the multimodal dataset using *chain_id* and *transfer_index*, with missing contextual values replaced by zero, indicating that no nearby contextual event was identified along the stops traversed in that segment.

Disruptions affecting the current segment can increase vehicle delays, which in turn can compromise transfer feasibility, trigger missed connections [2, 28], introduce additional transfer penalties [28], and propagate through later segments of the journey [6, 22]. As a result, local operational disruptions can accumulate and influence the passenger's final delay at the end of the journey [34]. The final contextual integration approach therefore characterises disruption exposure across the whole journey by analysing each travelled segment individually [28]. Rather than describing only the environment around the transfer stop, the contextual variables characterise the external conditions along the part of the route that produced the delay observed at the transfer point, making the contextual representation more consistent with the passenger-centred delay propagation framework proposed in this work.

3.10 Feature Engineering

3.10.1 Single-Mode Feature Selection

The aim of this phase was to build a feature space able to represent not only the current operational state of a transport service, but also the temporal, topological, and contextual conditions that

influence future delay evolution. The engineered features are grouped into five categories: temporal features, operational trip features, network topology features, delay propagation features, and contextual traffic features.

3.10.1.1 Temporal Features

Temporal features capture systematic variation in transport performance throughout the operating day. Public transport services are strongly influenced by demand patterns, particularly during morning and evening peak periods, when higher vehicle occupancy and road congestion tend to increase delay generation and propagation [3, 4, 34]. The variable *hour* represents the hour of day of each vehicle arrival. A binary indicator, *is_peak*, identifies observations that fall within typical peak commuting windows. An additional categorical variable, *time_period*, was initially created to divide the operating day into five broader intervals: night, morning peak, midday, evening peak, and late evening [42].

3.10.1.2 Operational Trip Features

Several features describe how a vehicle progresses along a route and how operational complexity varies across trips. The feature *stop_sequence_norm* describes the relative position of a stop within a trip. Each observation corresponds to one stop in a trip, and the sequence number indicates its position along the route. Raw stop sequence numbers were normalised by dividing by the maximum sequence index of the corresponding trip, giving values between 0 and 1: values near zero correspond to stops near the start of a trip, and values near one correspond to stops near the terminus. This normalisation makes trips of different lengths comparable and is particularly useful for modelling delay accumulation, since delays tend to build up as vehicles move along a route [42]. The feature *trip_length* is the total number of stops in a trip, providing an approximate measure of route length and operational complexity. Longer routes offer more opportunities for delay to accumulate and are therefore expected to be associated with higher average delay severity [17, 34]. The feature *time_diff* is the scheduled travel time between consecutive stops, computed as the difference between their scheduled arrival times. This gives an estimate of the expected travel duration between successive observations and allows the model to distinguish between closely spaced stops and longer route segments.

3.10.1.3 Network Topology Features

Two features characterise the structural properties of the transport network from a graph-theoretic perspective. The variable *route_frequency* counts the number of trips associated with a given route and indicates how frequently that route operates. High-frequency routes tend to operate under tighter scheduling constraints and may show different delay dynamics from lower-frequency services [9]. The variable *stop_degree* counts how many times each stop appears in the dataset,

providing a simple measure of stop importance and connectivity within the network. Highly connected stops are expected to behave differently, in terms of delay, from peripheral stops with limited connectivity [18].

3.10.1.4 Delay Propagation Features

As delay propagation is the central problem investigated in this work, several features were specifically designed to capture how delays evolve along a trip. The variable *delay* is the simulated delay observed at the current stop. The feature *prev_delay* is the delay observed at the immediately preceding stop. The feature *cumulative_delay* sums the delays observed at all preceding stops within the same trip, giving an overall measure of the total delay accumulated up to the current stop and capturing progressive congestion and cascade effects [22, 40]. The feature *delta_delay* measures the instantaneous change in delay between consecutive stops, computed as the difference between the current and previous delay observations. This gives a local picture of delay dynamics, showing whether conditions are deteriorating, recovering, or stable at each point along the route.

3.10.1.5 Contextual Traffic Features

To bring external operational conditions into the prediction framework, contextual features derived from DATEX II road closure and traffic event records were added to the dataset. Road closure information is represented through three variables: *has_closure*, a binary indicator of whether at least one closure event was detected near a stop; *num_closures*, the count of nearby closures; and *closure_intensity_nearby*, the average severity score of those closures. Traffic disruption conditions are captured through five further variables: *has_traffic_event*, a binary indicator of nearby traffic activity; *num_traffic_events*, the count of nearby traffic events; *traffic_disruption_score*, a weighted aggregate measure of disruption intensity; *num_speed_events*, the count of speed-related incidents; and *num_obstruction_events*, the count of physical obstruction events. Together, these variables characterise the local disruption environment around each stop and capture different forms of operational disruption that can contribute to delay generation and propagation [4, 15].

3.10.2 Single-Mode Correlation Analysis

Having built the feature space, feature relevance and redundancy were assessed through correlation analysis [2, 11]. This included Pearson correlation matrices, individual correlations with the prediction target, and a systematic search for highly correlated feature pairs. Three successive refinement steps were carried out, each testing a different way of handling temporal redundancy.

As shown in Figure 3.7, the initial analysis found extremely high correlations between the prediction target and both the current delay ($\rho = 0.993$) and the previous delay ($\rho = 0.982$). This is consistent with the sequential and persistent nature of delay propagation in public transport: once a vehicle accumulates delay [5, 8], that delay tends to persist and carry over to later stops.

Beyond these strong temporal variables, several operational features showed moderate correlations with the target: *is_peak* ($\rho = 0.517$), *trip_length* ($\rho = 0.405$), and *stop_sequence_norm*

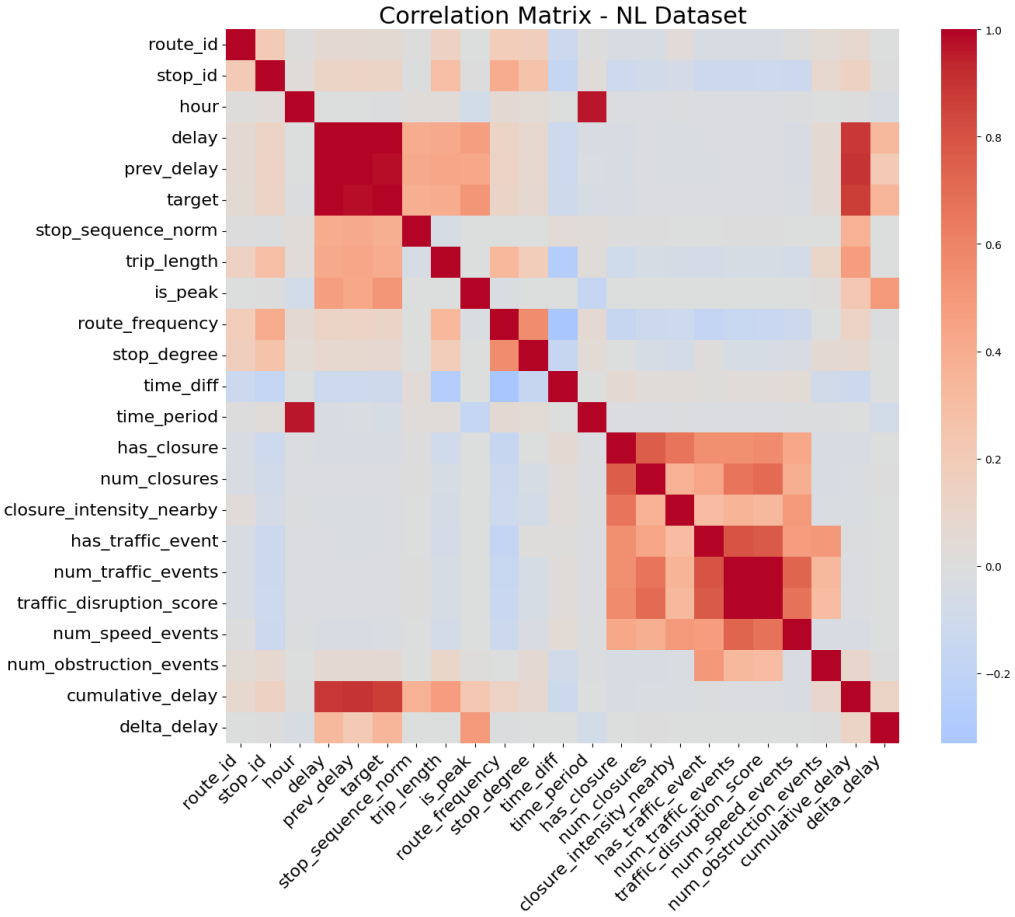


Figure 3.7: Initial feature correlation matrix.

($\rho = 0.385$). These results confirm that peak-hour operation, route length, and a stop's relative position in a trip all carry predictive information [34], in line with the expectation that delays build up over longer routes and worsen during high-demand periods [34].

Two patterns of redundancy were also identified. First, *hour* and *time_period* showed a near-perfect correlation ($\rho = 0.964$), since *time_period* was derived directly from *hour* by discretisation. As both variables represent the same temporal information at different levels of detail, including both was considered unnecessary, and *time_period* was removed from later experiments. Second, *delay* and *prev_delay* had a mutual correlation of 0.992, indicating substantial temporal overlap between consecutive operational states. Since these two variables describe closely related delay conditions separated by only one stop, including both risks introducing temporal multicollinearity and redundant information. Although each variable has predictive value on its own, their strong association with each other and with the target suggests that they largely capture the same delay propagation behaviour [5, 8] rather than offering complementary information. Three strategies were tested to resolve this redundancy.

Approach 1 – Removing *prev_delay*. The simplest strategy was to keep only the current delay and remove the immediately preceding one. After this removal, the correlation structure became

more balanced overall, but the correlation between *delay* and the target remained at 0.993, showing that the current delay was still the dominant explanatory factor for subsequent delay. This suggests that, although the feature space was simplified, the underlying temporal dependency structure of the dataset remained largely unchanged: future delays were still strongly driven by the current operational state.

Approach 2 – Aggregating *prev_delay* and *delay*. A composite feature, *delay_history_mean*, was built as the arithmetic mean of the two consecutive delay observations. This feature correlated at 0.998 with *delay* and 0.989 with the target, effectively reproducing the same information as *delay* alone. Given the extremely high mutual correlation between *delay* and *prev_delay* ($\rho = 0.992$), their average produced a feature almost identical to each of the original variables, so this approach failed to reduce temporal redundancy.

Approach 3 – Weighted temporal history. The third strategy replaced *prev_delay* with a temporally weighted aggregation of the five most recent delay observations, giving more weight to recent observations to capture recency effects [9, 16]:

$$\begin{aligned} WeightedDelayHistory_t = 0.40 \cdot Delay_{t-1} + 0.25 \cdot Delay_{t-2} + 0.15 \cdot Delay_{t-3} \\ +0.10 \cdot Delay_{t-4} + 0.10 \cdot Delay_{t-5} \end{aligned} \tag{3.14}$$

where $Delay_{t-k}$ is the delay observed k stops before the current observation. This representation allows cumulative propagation effects across several previous operational states to be modelled, rather than relying solely on single-step temporal dependencies. The resulting feature correlated at 0.850 with the target, substantially lower than the 0.982 obtained with *prev_delay*, and no longer appeared among the highly correlated feature pairs. This shows that the weighted history captures a complementary temporal dimension of delay propagation, reducing excessive concentration around the immediately preceding delays. Of the three approaches tested, the weighted temporal history feature gave the best balance between preserving temporal information, reducing redundancy, and remaining interpretable. It was therefore adopted as the temporal propagation representation in all subsequent experiments [5, 9].

To further improve feature compactness and reduce unnecessary redundancy, the contextual traffic features were also analysed. A near-perfect correlation ($\rho = 0.994$) was found between *num_traffic_events* and *traffic_disruption_score*, reflecting the fact that *traffic_disruption_score* was derived as a weighted aggregation of traffic event counts. As both variables represented essentially the same information, *traffic_disruption_score* was removed, keeping *num_traffic_events* as the more directly interpretable representation. The remaining contextual variables, covering both closure and traffic event features, showed low correlations with the target and no problematic dependencies between features, so they were kept unchanged.

The resulting final feature space, illustrated in Figure 3.8 presents a more compact and balanced correlation structure suitable for predictive modelling, while preserving variables essential for modelling delay propagation dynamics. Feature engineering aimed not to eliminate all

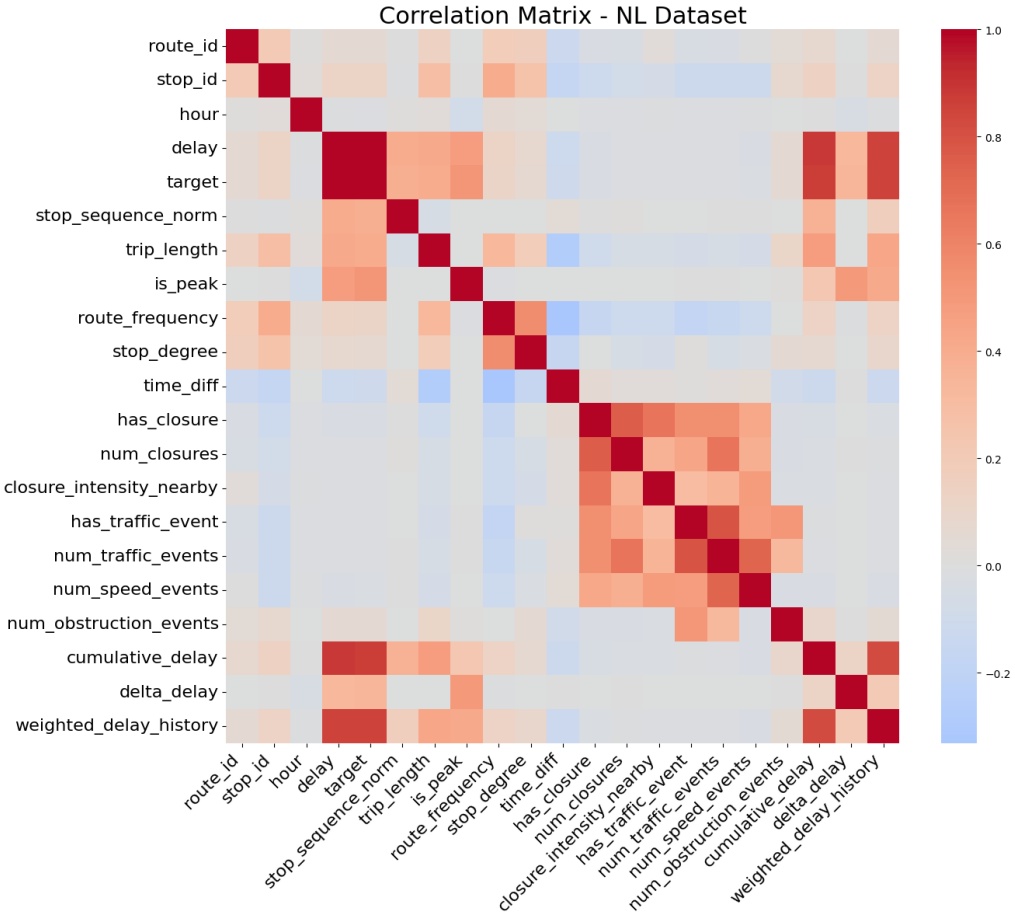


Figure 3.8: Correlation matrix with final features.

correlations, but to remove variables that redundantly represented the same underlying information. The final set of features was designed to balance redundancy reduction with information preservation. Variables were removed only when they represented alternative formulations of the same underlying information. Correlated features describing distinct transport phenomena, such as *cumulative_delay*, *route_frequency*, and *stop_degree*, were retained to preserve complementary operational and topological information [21].

3.10.3 Single-Mode Feature Importance and Contextual Feature Screening

A feature importance analysis was carried out to empirically verify the relevance of the engineered features and quantify their relative contribution to predictive performance. As shown in Figure 3.9, the results confirm the dominant role of the temporal propagation variables [5, 8]. The current delay, *delay*, was the most informative feature, followed by the weighted temporal history representation and the peak-hour indicator *is_peak*. Operationally meaningful features such as *stop_sequence_norm* and *trip_length* also made a meaningful contribution, consistent with the moderate correlations seen in the correlation analysis. Contextual variables related to road closures and traffic events ranked lowest in importance. Nevertheless, these features were kept in the

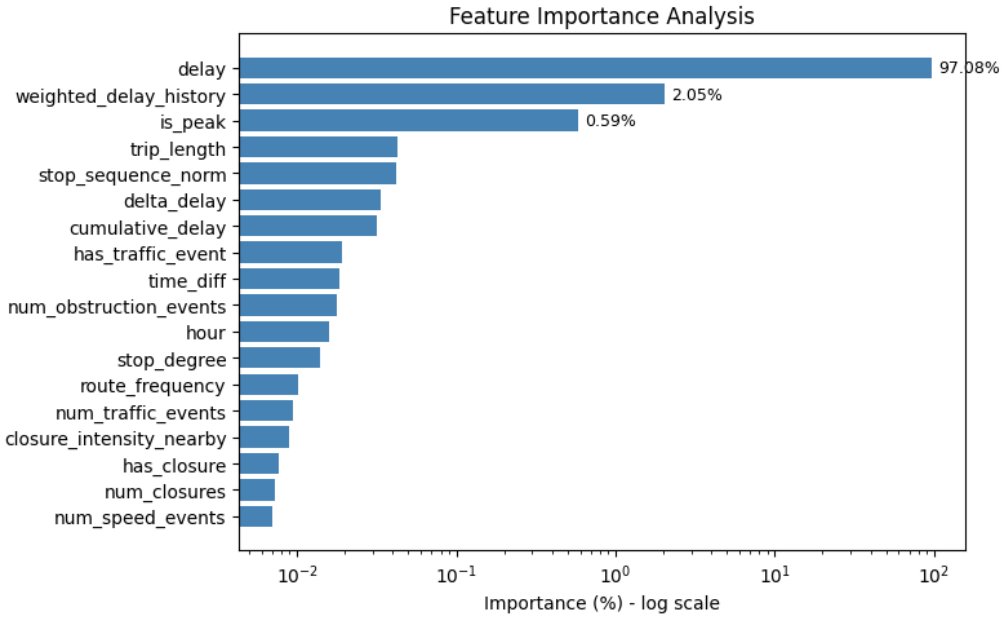


Figure 3.9: Feature importance analysis.

final framework because of their relevance for representing realistic multimodal operating conditions and external disruption scenarios [15], following a screening analysis of their effect on model performance.

3.10.3.1 Contextual Feature Screening

Although the contextual variables showed relatively low correlation with the prediction target (road closure-related features ranging from approximately -0.036 to -0.024, and traffic-event features from 0.029 to 0.056) and comparatively low importance scores, their removal was not assumed automatically. As these variables represent real operational disruptions that can affect transport performance, a further feature screening procedure was carried out to assess their practical impact on predictive modelling. The aim of this screening was to determine whether contextual variables should be kept in the feature space or removed because of limited predictive relevance. Separate experiments were therefore designed for the road closure and traffic event features, comparing model performance under alternative feature configurations with and without the corresponding contextual variables. This screening analysis used the XGBoost model, allowing the influence of contextual variables on predictive performance to be assessed [5, 8] while keeping the rest of the feature space unchanged. The results of this analysis are presented and discussed in Section 5.3.

3.11 Multimodal Feature Engineering

3.11.1 Features Characterization

In the multimodal framework, each observation corresponds to a transfer event within a passenger journey chain [21]. The feature space was designed to capture not only vehicle-level delay dynamics, but also passenger delay propagation across multiple transport segments, transfer operations, network topology, and contextual disruptions. Features were extracted from passenger chains, transfer events, chain summaries, and contextual infrastructure data, giving a comprehensive representation of multimodal delay propagation, presented in Table 3.3.

Unlike conventional delay prediction approaches, which focus exclusively on vehicle-level delays, the proposed feature space explicitly models passenger-level delay propagation [3, 28]. Transfer-related variables, missed-connection indicators, and transfer penalties were introduced to capture how delays build up and propagate [28] across multiple transport segments and modal transitions, until the passenger reaches the final destination.

3.11.1.1 Multimodal Correlation Analysis

After building the multimodal feature space, a correlation analysis was carried out to assess relevance, identify redundancy, and check for multicollinearity [9, 11]. Unlike the single-mode dataset, the multimodal representation includes variables describing passenger delay propagation, transfer operations, vehicle delays, journey topology, and contextual operating conditions. The aim of this analysis was therefore not to eliminate all correlations, but to identify variables representing the same underlying transport phenomenon, while preserving the complementary information needed [5] to model multimodal delay propagation.

As shown in Figure 3.10, several groups of highly correlated features were identified. Each group was analysed individually to decide whether the observed correlations reflected unwanted redundancy or meaningful dependencies inherent to passenger journeys. Feature engineering decisions were therefore guided by both statistical evidence and domain knowledge, to keep the final feature space compact while preserving the information needed to model delay propagation across successive transport segments and transfers.

The first feature engineering change addressed the representation of passenger delay at transfer points. The initial correlation analysis showed that the features *arrival_delay_at_transfer*, *ready_delay_at_transfer*, and *passenger_delay_before* all had the same correlations ($\rho = 1.0$), indicating that the three variables represented the same operational state. Although introduced at different stages of the transfer process, all three ultimately quantify the accumulated passenger delay immediately before boarding the next service. More specifically, *arrival_delay_at_transfer* is the delay on arrival at the transfer location, *ready_delay_at_transfer* is the delay after accounting for the minimum transfer time [28], and *passenger_delay_before* is the passenger's delay immediately before attempting to board the connecting service. Since the minimum transfer time is constant for each transfer event, these three variables evolve identically throughout the dataset,

Feature	**Description**
Journey structure and topology	
transfer_index	Position of the current transfer within the passenger journey chain.
n_transfers	Total number of transfers required to complete the journey.
n_modes	Number of distinct transport modes used throughout the journey.
is_multimodal	Binary indicator identifying whether multiple transport modes are involved.
is_intermodal	Binary indicator indicating whether the current transfer occurs between different transport modes.
from_route_id, to_route_id	Identifiers of the routes involved before and after the transfer.
from_stop_id, to_stop_id	Origin and destination stops associated with the transfer event.
origin_stop_id, destination_stop_id	Journey origin and final destination stops.
Temporal and scheduling features	
start_hour	Hour at which the passenger journey begins.
is_peak	Binary indicator identifying peak-hour operations.
planned_duration_min	Scheduled duration of the complete passenger journey.
planned_wait_min	Scheduled waiting time available for the transfer.
min_transfer_time	Minimum transfer time required between the two transport services.
intended_departure_min	Scheduled departure time of the intended connecting service.
Delay propagation features	
passenger_delay_before	Accumulated passenger delay immediately before the current transfer.
vehicle_delay_current_segment	Delay of the vehicle arriving at the transfer point.
vehicle_delay_next_leg	Delay associated with the connecting vehicle after the transfer.
initial_vehicle_delay	Delay affecting the first vehicle of the journey chain.
missed_connection	Binary indicator identifying whether the intended connection was missed.
transfer_penalty_departure	Additional delay caused by waiting for a later departure after a missed connection.
Contextual features	
num_closures	Number of nearby road closure events associated with the traversed segment.
closure_intensity_nearby	Aggregated severity indicator of nearby closure events.
num_traffic_events	Number of nearby traffic-related events.
num_speed_events	Number of nearby speed-management events.
num_obstruction_events	Number of nearby vehicle obstruction events.

Table 3.3: Features extracted for the multimodal delay propagation dataset before feature selection.

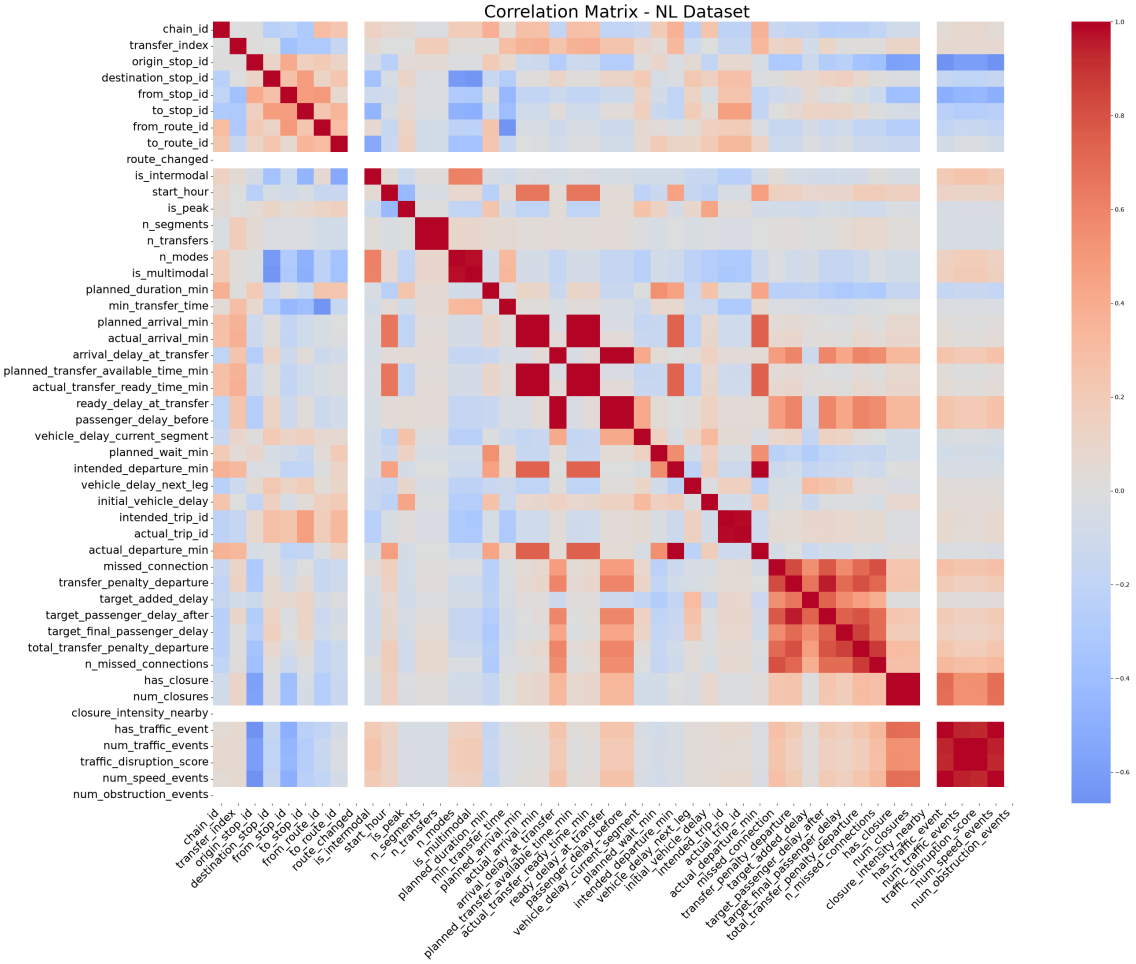


Figure 3.10: Initial correlation matrix before feature engineering.

making them fully redundant. All three also had identical correlations with the final prediction target ($\rho = 0.432$), confirming that none added predictive information beyond the others. *arrival_delay_at_transfer* and *ready_delay_at_transfer* were therefore removed, keeping *passenger_delay_before* as the representative variable. This feature was considered the most intuitive representation of the passenger's accumulated delay immediately before the transfer, and serves as the natural reference state from which subsequent delay propagation occurs. While keeping only *passenger_delay_before* removed the immediate redundancy identified above, this feature still represented only the passenger's instantaneous delay just before the transfer, so the temporal information available to the model remained limited to a single operational state.

To address this, the same feature engineering strategy previously used for the single-mode dataset was extended to the multimodal framework. Rather than relying solely on the current passenger delay, the instantaneous observation was replaced with *weighted_passenger_delay_history*, which summarises the passenger delays observed throughout the previously completed journey segments using a weighted temporal representation [5]. More recent delays receive larger weights, while earlier observations still contribute to the historical context, allowing the feature to capture the evolution of delay propagation rather than a single delay measurement.

Following this change, the direct observation of the current passenger delay was removed from the feature space, leaving the weighted historical representation as the primary descriptor of passenger delay propagation. The correlation with the final prediction target decreased slightly, from 0.432 to 0.394, but this was expected. The aim of the replacement was not to maximise the linear correlation with the target, but to provide a richer temporal representation able to model cumulative delay propagation across multiple journey segments [28], while reducing dependence on a single instantaneous observation [34].

This weighted-history replacement follows the same logic used in single-mode feature engineering, where *delay* and *prev_delay* were replaced with a weighted historical representation. In both cases, the aim was to model delay evolution over several previous operational states, rather than relying solely on the most recent observation, resulting in a more informative and temporally robust feature space. As with the passenger delay representation, the delay of the current vehicle describes only the operational state of the active journey segment, and therefore says nothing about how vehicle delays evolved during previously completed segments of the journey. To capture this evolution, *weighted_vehicle_delay_history* summarises the delays experienced by vehicles serving previous journey segments using a weighted historical representation, turning the instantaneous vehicle delay into a propagation-aware feature [9] that models the recent evolution of delay accumulation. Although *vehicle_delay_current_segment* and *weighted_vehicle_delay_history* are strongly correlated ($r = 0.902$), they describe different aspects of the delay propagation process: the former is the instantaneous operational state of the vehicle serving the current segment, while the latter captures the accumulated delay evolution across previously completed segments [8, 22]. Since the aim of the proposed framework is to model passenger delay propagation rather than isolated operational states, keeping both variables lets the model jointly use the current vehicle condition and its recent temporal history. Both features were therefore kept in the final feature

space, providing complementary temporal information despite their statistical correlation.

The next step addressed the structural descriptors of the passenger journey. The correlation analysis found a perfect correlation ($\rho = 1.0$) between *n_segments* and *n_transfers*, indicating complete redundancy. This was expected, since each additional transfer necessarily introduces a new journey segment, giving the deterministic relationship:

$$n_{\text{segments}} = n_{\text{transfers}} + 1 \tag{3.15}$$

As shown in Equation (3.15), each additional transfer adds exactly one journey segment, giving a deterministic linear relationship between the two variables. Keeping both would therefore add redundant structural information without improving the predictive capability of the framework, so *n_segments* was removed.

n_transfers was kept because it gives a more meaningful representation of multimodal journey complexity. Unlike the number of segments travelled, the number of transfers directly quantifies the opportunities for delay propagation, missed connections, and transfer penalties [2, 28], making it a more interpretable descriptor for modelling how passenger delay accumulates across multimodal networks.

The next step addressed how transport mode diversity is represented throughout the journey. The initial correlation analysis found a very strong association between *is_multimodal* and *n_modes* ($\rho = 0.976$), indicating that both describe the same underlying journey characteristic. Unlike *is_multimodal*, which only indicates whether more than one transport mode is used, *n_modes* preserves the full information on how many distinct modes are involved: it distinguishes journeys using one, two, three, or more modes, whereas the binary descriptor assigns the same value to every multimodal journey regardless of its modal complexity. Although both variables had relatively weak linear correlations with the final prediction target, *n_modes* was kept, as it is the more informative representation of multimodal journey structure. Keeping the continuous descriptor preserves additional information without introducing redundancy, and allows the learning algorithms to exploit possible non-linear relationships between modal diversity and delay propagation. *is_multimodal* was therefore removed from the final feature space: the binary indicator only answers whether a journey is multimodal, whereas the number of modes quantifies its degree of multimodality.

The next change addressed the group of temporal variables related to transfer timing. The features *planned_arrival_min*, *actual_arrival_min*, *planned_transfer_available_time_min*, and *actual_transfer_ready_time_min* showed extremely high mutual correlations in the initial feature space. This was expected, since these variables represent closely related timestamps around the same transfer operation. In particular, the transfer availability variables are derived directly from the corresponding arrival times by adding the minimum transfer time [28]. Although these features describe different stages of the transfer process, they mostly encode absolute schedule positions rather than delay propagation itself. Their correlations with the final prediction target were also very low, indicating that the exact planned or actual clock time of a transfer adds little direct

information once more meaningful temporal descriptors are already available.

These absolute timestamp variables were therefore removed from the final feature space. The temporal representation was instead preserved through variables with clearer operational meaning: *start_hour*, which captures the time period in which the journey begins; *planned_wait_min*, which describes the scheduled waiting time available for the transfer; and the engineered feature *weighted_passenger_delay_history*, which represents the accumulated passenger delay state [5, 34]. This reduced redundancy among highly correlated timing variables while keeping the information most relevant for modelling transfer feasibility and passenger delay propagation.

The following step focused on how departure time is represented during passenger transfers. The variable *actual_departure_min* is the absolute clock time at which the passenger boards the connecting service, whereas *planned_wait_min* describes the scheduled transfer time available between successive segments. Although both relate to the transfer process, they carry fundamentally different information: the absolute departure time mainly reflects the timetable position of a journey within the day and depends largely on the scheduled service, whereas the planned waiting time directly characterises the operational robustness of the transfer itself, showing the time margin available to accommodate incoming delays before a connection is missed. *actual_departure_min* also showed a weak correlation with the final prediction target ($\rho = -0.074$), suggesting that the exact departure timestamp adds little once the remaining temporal and propagation-related features are available. *actual_departure_min* was therefore removed from the final feature space, while *planned_wait_min* was kept as the representative descriptor of transfer robustness, since it more directly captures the passenger's opportunity to complete a transfer successfully and so contributes more directly to modelling delay propagation across multimodal journeys.

The next step addressed the trip identifier variables, *intended_trip_id* and *actual_trip_id*. Although these identify the planned and the actually boarded services after each transfer, their numerical values carry no intrinsic ordinal or quantitative meaning. They serve only as unique identifiers for individual vehicle trips. The initial correlation analysis also found a very strong association between the two identifiers ($\rho = 0.984$), reflecting the fact that, in most journeys, the passenger successfully boards the intended service, and only a relatively small share of transfers result in an alternative trip due to a missed connection. Keeping both identifiers would therefore add redundant categorical information without offering further insight into the mechanisms driving delay propagation [13]. More importantly, including trip identifiers as numerical inputs could encourage the learning algorithms to associate specific trip IDs with particular delay patterns, reducing the generalisation ability of the framework. Since the operational characteristics of each service are already represented through variables such as route identifiers, departure times, vehicle delays, transfer penalties, and contextual information, the trip identifiers themselves add no meaningful predictive information. Both *intended_trip_id* and *actual_trip_id* were therefore removed from the final feature space.

Following the same decision made at the single-mode stage, the aggregated contextual variable *traffic_disruption_score* remained excluded from the multimodal dataset, while the individual traffic-related descriptors (*num_traffic_events*, *num_speed_events*, and *num_obstruction_events*)

were kept. This ensured methodological consistency between both stages of the framework, while avoiding the reintroduction of redundant aggregated information. The binary contextual variables *has_closure* and *has_traffic_event* were subsequently removed from the multimodal feature space: both simply indicate the presence or absence of nearby road closures or traffic events, whereas the corresponding quantitative variables (*num_closures*, *num_traffic_events*, and the individual traffic event descriptors) already capture this information, with the added detail of the severity of the surrounding disruptions [4].

Compared with the single-mode dataset, the quantitative contextual variables showed stronger associations with the prediction target in the multimodal framework. This is mainly explained by the level at which contextual information was integrated. In the single-mode dataset, contextual features were computed at stop level and assigned individually to each stop-level observation, so each observation captured only the disruptions near the corresponding stop. The multimodal framework, in contrast, aggregates contextual events over the stops traversed within the current journey segment, so each transfer observation captures the contextual conditions encountered along the travelled segment rather than at a single stop. This segment-level representation is more consistent with the multimodal delay propagation problem, since disruptions may affect vehicle movement at any point along the segment before it reaches the transfer location. The contextual variables therefore become more informative for predicting propagated passenger delay, reflected in the higher correlations observed for *num_closures*, *num_traffic_events*, and *num_speed_events* compared with the single-mode formulation. At the same time, the binary indicators remained highly correlated with their corresponding count-based variables, with correlation coefficients of $\rho = 0.993$ between *has_closure* and *num_closures*, and $\rho = 0.939$ between *has_traffic_event* and *num_traffic_events*. The binary variables therefore added little beyond what the quantitative descriptors already captured, so only the quantitative contextual features were kept, preserving richer operational information while avoiding redundant binary representations.

The chain-level variables *total_transfer_penalty_departure* and *n_missed_connections* were subsequently removed from the feature space. Although both correlated relatively strongly with the final passenger delay, they represent information aggregated over the entire journey rather than information available at the current transfer event. Since each observation in the multimodal dataset corresponds to a single transfer, these variables implicitly incorporate the outcomes of later transfers in the same journey. Including them would therefore expose the learning algorithms to information not yet available at prediction time, introducing target leakage and giving overly optimistic predictive performance [15]. The local transfer descriptors *transfer_penalty_departure* and *missed_connection* were kept instead, as they describe only the outcome of the current transfer and remain consistent with the sequential nature of delay propagation. Moreover, the cumulative effect of successive transfer penalties and missed connections is already reflected in the prediction target itself, making the aggregated chain-level variables unnecessary.

Finally, *route_changed* was removed from the final feature space due to a lack of variability in the dataset. By construction, every transfer event involves a change from one transport service to another, so *route_changed* took the constant value of one for all observations. As a feature with

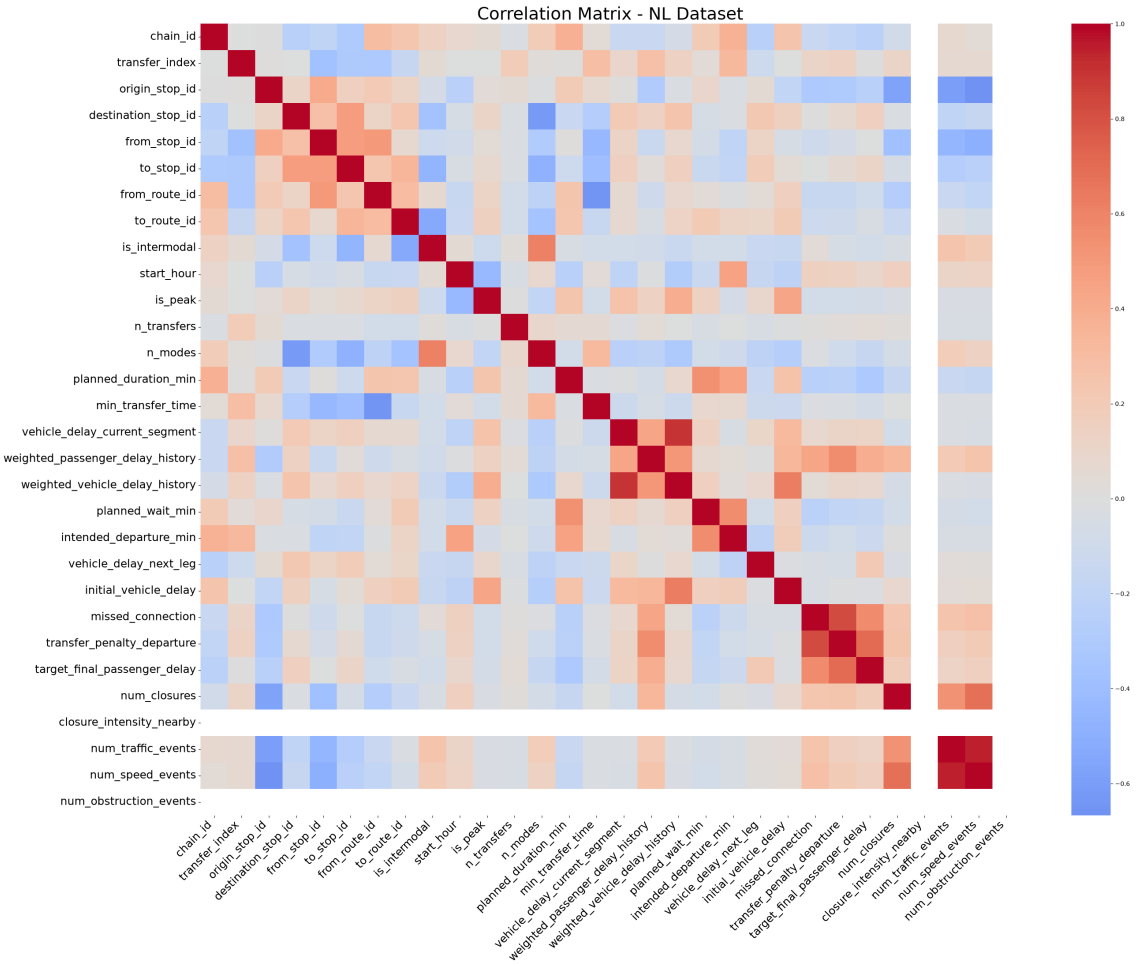


Figure 3.11: Correlation matrix after feature engineering process.

zero variance cannot help distinguish between observations, it carries no predictive information and cannot be exploited by machine learning algorithms; *route_changed* was therefore discarded. This is consistent with the multimodal modelling framework, where each observation explicitly represents a transfer between consecutive journey segments, a route change is therefore an inherent characteristic of every observation, rather than a discriminative feature.

The resulting correlation matrix after the complete feature engineering process is shown in Figure 3.11. Intermediate correlation matrices generated after each feature engineering iteration are provided in Appendix A. Compared with the initial feature space, the final dataset has a considerably cleaner correlation structure, with the most significant sources of multicollinearity and redundant information removed, while keeping the variables that best characterise delay propagation in multimodal journeys.

Feature removal was therefore guided by semantic redundancy rather than by correlation values alone. Variables were discarded only when they described identical operational states or could be directly derived from other retained features, for example, duplicated temporal timestamps, equivalent delay measurements, or aggregated target-related quantities. Correlated variables de-

scribing different transport phenomena, by contrast, were deliberately kept. Features representing vehicle behaviour, passenger delay history, transfer characteristics, network topology, temporal conditions, and contextual disruptions each capture complementary aspects of multimodal transport operations, and so contribute distinct predictive information despite showing moderate correlations. High correlations were removed only when variables represented the same underlying operational information. Correlated variables describing different temporal aspects of delay propagation were deliberately kept. For example, *vehicle_delay_current_segment* represents the instantaneous operational state of the vehicle at the current segment, while *weighted_vehicle_delay_history* summarises the accumulated delay evolution across previous segments [5, 9]. Although highly correlated, these variables capture complementary temporal information relevant for modelling delay propagation. The final feature set thus balances redundancy reduction against information preservation, keeping a compact representation of the multimodal transport system while retaining the operational, temporal, topological, propagation, and contextual information needed to model passenger delay propagation accurately. This refined feature set was used as input for all predictive models evaluated in the following chapter, giving a consistent basis for comparing the XGBoost, LSTM, and GNN approaches [21, 22, 37].

3.11.1.2 Feature Importance Analysis

Following the correlation analysis and feature engineering process, a further feature importance analysis was carried out using the XGBoost model [5, 8]. While correlation analysis identifies linear relationships and potential redundancy between variables, feature importance evaluates each feature's actual contribution to the predictive model during training.

As shown in Figure 3.12, the XGBoost model assigns the highest importance to *transfer_penalty_departure*, accounting for approximately 46.5% of total feature importance. This is expected, as this variable directly quantifies the additional delay introduced by the current transfer when a passenger misses the intended connection and must wait for the next available service [28]. As transfer penalties are the primary mechanism through which delays propagate across successive journey segments, they naturally become the strongest predictor of the final passenger delay. The second most influential feature is *initial_vehicle_delay* (10.8%), showing that the delay accumulated at the start of the journey continues to influence the passenger throughout the rest of the trip. This confirms that delay propagation depends strongly on the operational state established during the initial segment. Similarly, *vehicle_delay_next_leg* (4.3%) also contributes to the prediction, by modelling the operating conditions of the connecting service after the transfer. Temporal and journey-related characteristics also contribute: the binary feature *is_peak* (6.1%) reflects the greater operational variability typically associated with peak-hour traffic [4, 9], while variables such as *planned_duration_min*, *planned_wait_min*, *n_transfers*, and *min_transfer_time* capture structural characteristics of the journey that affect the likelihood of delay accumulation. The historical delay descriptors *weighted_passenger_delay_history* (0.74%) and *weighted_vehicle_delay_history* (0.52%) have relatively modest individual importance values. Nevertheless, they provide a compact representation of how passenger and vehicle delays

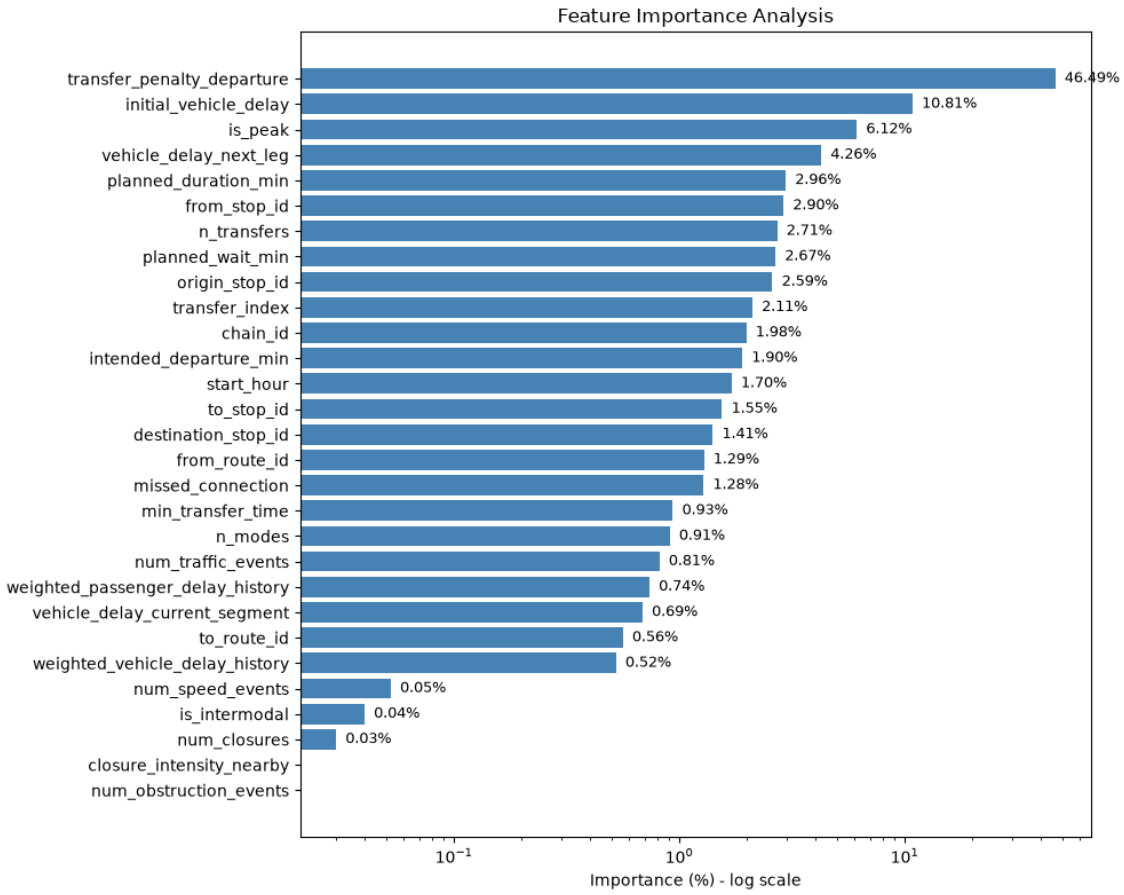


Figure 3.12: Feature importance.

evolve over time, replacing several highly correlated instantaneous delay variables and reducing redundancy in the feature space [9]. Their inclusion therefore reflects a trade-off between predictive contribution and a more robust temporal representation of delay propagation. As expected, the contextual variables continue to show relatively low individual importance, consistent with the screening analysis discussed earlier in Subsubsection 3.10.3. Even so, compared with the single-mode dataset, the retained contextual variables show stronger correlations with the prediction target, suggesting that segment-level aggregation captures external operating conditions more effectively than the earlier stop-level representation.

3.11.2 Final Feature Selection

Following the correlation analysis and feature selection process, the final multimodal dataset consists of 28 predictive features, presented in Table 3.4. The retained variables capture complementary aspects of passenger journeys, including spatial information, transfer characteristics, temporal context, delay propagation, and contextual operating conditions. This final feature set was used as input for the XGBoost, LSTM, and GNN models presented in the following chapter.

Category	Features
Journey identifiers	chain_id, transfer_index
Spatial information	origin_stop_id, destination_stop_id, from_stop_id, to_stop_id, from_route_id, to_route_id
Transfer characteristics	is_intermodal, n_transfers, n_modes, min_transfer_time, planned_wait_min, intended_departure_min
Temporal context	start_hour, is_peak, planned_duration_min
Delay propagation	weighted_passenger_delay_history, weighted_vehicle_delay_history, vehicle_delay_next_leg, initial_vehicle_delay, missed_connection, transfer_penalty_departure
Contextual information	num_closures, closure_intensity_nearby, num_traffic_events, num_speed_events, num_obstruction_events

Table 3.4: Final feature set used for multimodal delay prediction after feature engineering and feature selection.

3.12 Expected Outcomes

The proposed framework is expected to demonstrate that explicitly modelling passenger transfers, missed connections, and accumulated delay propagation provides a more realistic and informative representation of multimodal journey reliability than conventional, vehicle-centred delay prediction. From a methodological perspective, the comparative evaluation of XGBoost, LSTM, and GNN models is expected to clarify which learning paradigm is best suited to each stage of the problem, namely local, single-trip delay prediction and longer-horizon, passenger-centred delay propagation.

From a practical perspective, the resulting methodology is intended to contribute towards passenger-centred intelligent transport systems, supporting more informed travel decisions, improved transfer reliability assessment, and a foundation for future multimodal journey planning applications.

Chapter 4

Prediction Models

4.1 Overview of the Evaluated Predictive Models

The aim of this work is to investigate how different machine learning paradigms model delay propagation in public transport systems, and to assess their suitability for both single-mode and multimodal passenger delay prediction. This study evaluates three fundamentally different learning paradigms, each able to capture complementary characteristics of transport operations: a tree-based ensemble model, a recurrent neural network, and a graph neural network [6, 21].

The evaluated models are Extreme Gradient Boosting (XGBoost), Long Short-Term Memory (LSTM) networks, and Graph Neural Networks (GNNs), each modelling passenger delay propagation from a different perspective. XGBoost operates on structured tabular data and exploits non-linear relationships [4, 5] between engineered operational, temporal, structural, and contextual features. LSTM networks explicitly model temporal dependencies by learning sequential patterns [36, 40] from consecutive transport observations, allowing cumulative delay evolution to be represented over time. Graph Neural Networks, in turn, incorporate the spatial topology of the transport network by modelling stops as graph nodes [6, 21] connected through operational links, enabling information to propagate across neighbouring nodes. These three paradigms were chosen to give a comprehensive evaluation of delay propagation modelling from complementary perspectives: XGBoost provides a strong baseline for structured transport prediction tasks, LSTM investigates the contribution of temporal sequential learning, and GNN tests whether explicitly modelling the underlying network topology improves predictive performance. Together, they allow local operational modelling, temporal dependency learning, and spatial propagation mechanisms to be compared within a single experimental framework.

The same three paradigms were evaluated on both the single-mode and multimodal datasets developed in this work. While the underlying learning algorithms remain unchanged, the input representation differs according to the corresponding feature engineering process: the single-mode models operate on individual vehicle trips, whereas the multimodal models incorporate passenger transfers, multimodal propagation features- Both contain contextual information extracted from

the integrated GTFS and DATEX II datasets. This common evaluation framework allows a direct comparison between conventional single-trip delay prediction and the proposed multimodal passenger delay propagation approach [28].

4.2 Single-mode Prediction Models

4.2.1 Extreme Gradient Boosting (XGBoost)

XGBoost was adopted as the primary tree-based baseline due to its robustness, computational efficiency, and consistently strong performance on structured tabular prediction problems [5, 8]. Public transport delay prediction is characterised by heterogeneous feature types, including numerical operational variables, temporal descriptors, network characteristics, historical delay indicators, and contextual information. XGBoost is well suited to this type of data, as it can efficiently model complex non-linear relationships without extensive feature transformation. It also provides feature importance estimates that aid interpretation of the prediction process [5, 8], making it particularly useful for analysing the relative contribution of the engineered transport features proposed in this work. For these reasons, XGBoost serves both as a competitive predictive baseline and as an interpretable model for assessing the effectiveness of the proposed feature engineering methodology.

4.2.1.1 Model Description

XGBoost is an ensemble learning algorithm based on gradient-boosted decision trees. Rather than building a single predictive model, the algorithm iteratively constructs a sequence of decision trees, where each new tree is trained to minimise the residual errors of the previous ensemble [8]. The final prediction is therefore the sum of the predictions generated by all trees. This iterative boosting strategy enables XGBoost to progressively learn increasingly complex non-linear relationships between transport characteristics and future passenger delays while maintaining high predictive accuracy and strong generalisation capability [8].

4.2.1.2 Input Representation

For both the single-mode and multimodal prediction tasks, XGBoost operates directly on the engineered feature vectors described in Chapter 3. These include temporal descriptors, operational transport variables, delay propagation indicators, network topology characteristics, transfer-related information, and contextual features derived from GTFS and DATEX II data. Although the learning algorithm is identical for both scenarios, the multimodal dataset extends the original feature space with passenger-oriented propagation variables describing transfers, multimodal journeys, and accumulated delay evolution across multiple transport segments. XGBoost can therefore be used to assess whether these additional engineered features improve passenger delay prediction compared with conventional single-trip delay modelling.

Hyperparameter	Value
Number of estimators	100
Maximum tree depth	5
Learning rate	0.1
Random seed	42

Table 4.1: Hyperparameter configuration adopted for the XGBoost model.

4.2.1.3 Model Configuration

The XGBoost configuration was selected considering the trade-off between predictive performance and model complexity. The number of estimators and maximum tree depth were kept moderate to reduce the risk of overfitting, while the learning rate was set to a standard value that allows stable boosting behaviour. This choice is consistent with previous delay prediction studies using XGBoost, where hyperparameter tuning is shown to be an important step for improving predictive performance [8].

4.2.2 Long Short-Term Memory Networks (LSTM)

While XGBoost models each transport observation independently using engineered feature vectors, it does not explicitly capture temporal dependencies between consecutive observations. Delay propagation, however, is inherently sequential: the delay experienced at a given stop is influenced by delays accumulated throughout the preceding journey [6]. To model these temporal relationships explicitly, a LSTM network was adopted.

LSTM networks are a specialised type of recurrent neural network designed to learn long-term temporal dependencies while mitigating the gradient problem commonly seen in conventional recurrent networks. Through their internal memory cells and gating mechanisms, LSTM architectures can selectively retain or discard information from previous observations, making them well suited to modelling cumulative delay propagation in public transport systems.

In this work, the LSTM model was designed to learn how delays evolve over consecutive transport observations, capturing recurrent operational patterns that cannot be directly represented by independent tabular observations.

4.2.2.1 Input Representation

Unlike tree-based models, LSTM networks require sequential rather than independent input observations. Each training sample was therefore represented as an ordered sequence of consecutive transport observations, allowing the model to learn temporal dependencies between successive delay states [5].

For the single-mode prediction task, sequences of length 10 were generated [29] using a sliding-window approach applied to each vehicle trip. Consecutive observations belonging to the same trip were grouped into fixed-length sequences, where each input sequence represents the operational state of the previous stops and the corresponding target is the delay at the next stop. This

Hyperparameter	Value
Sequence length	10
First LSTM layer	64 hidden units
Second LSTM layer	32 hidden units
Output layer	Dense (1 neuron)
Optimizer	Adam
Epochs	100
Batch size	32
Random seed	42

Table 4.2: Hyperparameter configuration adopted for the LSTM model.

formulation allows the network to capture the temporal evolution of delays throughout a journey by drawing on several preceding observations, rather than relying on a single instantaneous state.

Each observation in a sequence is described by the engineered features presented in Section 3.9.2, including temporal descriptors, operational variables, delay propagation features, structural network characteristics, and contextual information. Prior to training, all numerical input features were normalised using Min-Max scaling, to ensure comparable numerical ranges and improve optimisation stability [16, 21]. Target values were normalised independently and converted back to their original units during model evaluation.

4.2.2.2 Model Architecture

The LSTM architecture consists of two stacked recurrent layers followed by a fully connected output layer [27, 42]. The first layer comprises 64 hidden units and returns the complete sequence to a second LSTM layer of 32 hidden units. The resulting hidden representation is then processed by a dense output layer that predicts the passenger delay. This layered architecture allows the network to progressively learn increasingly abstract temporal representations, capturing both short-term operational fluctuations and longer-term delay propagation patterns throughout the transport sequence [5, 6, 40].

.

4.2.3 Graph Neural Networks (GNN)

XGBoost and LSTM operate on tabular and sequential representations of transport observations, respectively, but neither explicitly exploits the underlying topology of the transport network. Delay propagation, however, is not only a temporal phenomenon but also a spatial one, since delays may propagate between neighbouring stops connected through vehicle movements. To investigate whether incorporating network structure improves predictive performance, a Graph Neural Network (GNN) based on Graph Convolutional Network (GCN) [6, 21] was adopted. GNNs extend deep learning to graph-structured data by allowing neighbouring nodes to exchange information during learning [21]. In public transport systems, this is particularly relevant, since stops are naturally connected through routes and trips, allowing operational conditions at one location to

Hyperparameter	Value
First GCN layer	32 hidden units
Second GCN layer	Output layer
Activation function	ReLU
Loss function	Mean Squared Error (MSE)
Optimizer	Adam
Epochs	100
Learning rate	0.001
Random seed	42

Table 4.3: Hyperparameter configuration adopted for the GNN model.

influence nearby parts of the network. By exploiting these spatial dependencies, GNNs provide a perspective that complements conventional tabular and sequential learning models.

4.2.3.1 Graph Representation

The transport network was represented as a directed graph $G = (V, E)$, where each node $v \in V$ corresponds to a transport stop and each edge $e \in E$ represents a direct connection [21] between two consecutive stops within the same trip.

Graph edges were built directly from the GTFS stop sequences [15, 21], connecting every pair of consecutive stops within each trip. The resulting graph therefore preserves the operational topology of the transport network and allows information to propagate between neighbouring stops during graph convolution. Each node was associated with an aggregated feature vector describing the operational characteristics of the corresponding stop, obtained by aggregating the engineered transport variables described in Chapter 3, so that each stop represents its operational behaviour within the network. This representation allows the model to exploit local stop characteristics and the spatial relationships between neighbouring stops jointly, letting delay information propagate across the network during learning.

4.2.3.2 Model Architecture

The Graph Neural Network architecture is based on Graph Convolutional Networks (GCNs), implemented using the PyTorch Geometric framewor [21], and consists of two graph convolutional layers [6, 35]. The first projects the input node features into a 32-dimensional hidden representation, followed by a Rectified Linear Unit (ReLU) activation [35] to introduce non-linearity. A second graph convolution layer then produces the final node-level predictions. During each graph convolution, node representations are updated by aggregating information from neighbouring nodes [21], so the learned representation of each stop depends not only on its own operational characteristics but also on those of adjacent stops connected through the transport network. This mechanism allows the model to capture spatial delay propagation effects that cannot be represented by independent observations alone.

Property	**Next-Segment Delay**	**Final Journey Delay**
Prediction target	*target_passenger_delay_after*	*target_final_passenger_delay*
Prediction	Next journey segment	Entire remaining journey
Represents	Delay after completing the next segment	Delay at the final destination
Main application	Transfer monitoring	Journey arrival prediction
Main uncertainty	Current operational state	Future delay propagation
Difficulty	Lower	Higher

Table 4.4: Differences between the two prediction objectives.

4.2.3.3 Input Representation

Unlike XGBoost and LSTM, which operate on tabular observations and temporal sequences respectively, the GNN receives the entire transport network as input. Each node corresponds to a transport stop described by an engineered feature vector, while edges encode the physical connectivity between consecutive stops extracted from GTFS trip trajectories [15, 21]. This representation lets the model exploit node-level operational information and the spatial relationships encoded by the network topology at the same time. During training, node representations are iteratively updated through graph convolutions, allowing delay information to propagate across neighbouring stops before the final prediction is generated.

4.3 Multimodal Prediction Models

The predictive paradigms described above were then adapted to the proposed multimodal passenger delay prediction problem. Although the underlying learning algorithms remain unchanged, the prediction task differs substantially, due to the passenger-centred representation adopted throughout this work. Unlike the single-mode formulation, multimodal prediction requires modelling delay propagation across multiple transport segments, transfer events, and transport modes. The main differences therefore relate to the prediction objectives, input representation, and model implementation, described in the following sections.

4.3.1 Prediction Objectives

Unlike conventional delay prediction problems, multimodal passenger journeys naturally involve multiple prediction horizons. A passenger may need an estimate of the delay immediately after the next transfer, to determine whether the planned connection can still be made [28], while journey planning systems also need an estimate of the final arrival delay at the destination [2]. Two complementary prediction targets are therefore considered in this work. Table 4.4 summarises the differences between them: the first target estimates the passenger delay immediately after the next journey segment, while the second predicts the total accumulated delay at the final destination. These complementary objectives allow the proposed framework to support both local transfer monitoring and end-to-end journey reliability estimation.

The first prediction target, *target_passenger_delay_after*, represents the accumulated passenger delay immediately after completing the next journey segment. It estimates the passenger delay resulting from the current operational state of the journey, including the effects of vehicle delays, transfer waiting times, and missed connections [28]:

Mathematically,

$$\text{Delay}_{\text{next}} = \text{ActualArrival}_{\text{next}} - \text{PlannedArrival}_{\text{next}} \tag{4.1}$$

Here, the predicted delay corresponds to the accumulated passenger delay after completing the next segment, rather than the isolated delay of a single vehicle.

The second prediction target, *target_final_passenger_delay*, represents the total accumulated delay at the final destination of the complete passenger journey:

$$\text{Delay}_{\text{final}} = \text{ActualArrival}_{\text{destination}} - \text{PlannedArrival}_{\text{destination}} \tag{4.2}$$

This target is inherently more challenging, since it requires estimating the cumulative effect of all remaining journey segments and transfer opportunities after the current prediction instant.

Evaluating both targets together makes it possible to analyse not only local delay propagation immediately after a transfer, but also its longer-term impact on the complete multimodal journey, assessing the proposed framework from both an operational perspective (local delay propagation) and a passenger perspective (final journey reliability).

4.4 XGBoost Model for Multimodal Passenger Delay Propagation

XGBoost was also adopted as the first predictive model for the multimodal passenger delay propagation problem, building on the same rationale presented in Section 4.2.1 for the single-mode formulation. Here, however, the prediction task differs substantially from the single-mode setting. Rather than predicting the delay of the next vehicle at an individual stop, the objective is to estimate how delay propagates throughout an entire passenger journey, across multiple transport modes and transfer events. The prediction problem therefore shifts from vehicle-centred prediction to passenger-centred delay propagation [3], while the underlying learning algorithm and its main advantages, robustness, computational efficiency, and interpretability through feature importance [5, 8], remain unchanged.

4.4.1 Input Representation

The XGBoost model uses the final multimodal dataset obtained through the preprocessing and feature engineering pipeline described in Section 3.10.2. All input variables therefore correspond to information available at the prediction instant, ensuring that no future operational information is used during training. Each observation corresponds to a transfer event within a passenger journey and is represented by the final feature set resulting from the feature selection process. The

Hyperparameter	Value	Purpose
Number of trees	150	Sequential boosted trees
Maximum depth	4	Controls tree complexity
Learning rate	0.08	Gradual boosting updates
Row subsampling	0.9	Reduces overfitting and increases diversity
Feature subsampling	0.9	Reduces feature correlation between trees

Table 4.5: XGBoost model hyperparameter configuration.

model thus receives the passenger's state immediately before boarding the next service, including operational, temporal, multimodal, and contextual information.

4.4.2 Model Implementation

Since XGBoost requires numerical input variables, categorical attributes describing transport modes, mode combinations, and transfer types were converted to numerical representations using one-hot encoding [13]. This allows the model to learn independent contributions from each transport mode and transfer category while preserving the original categorical information. Unlike distance-based or gradient-based learning algorithms, XGBoost does not require feature normalisation, as tree-based ensemble methods are inherently insensitive to differences in feature scale. All numerical variables were therefore used directly during training.

Finally, the selected prediction target (*target_passenger_delay_after* or *target_final_passenger_delay*) was separated from the predictor variables to build the feature matrix used for model fitting.

4.4.3 Hyperparameter Configuration

The XGBoost regressor was configured using the squared error objective function (`reg:squarederror`), appropriate for continuous regression problems [11]. The model consists of 150 gradient-boosted decision trees, each with a maximum depth of four levels. A learning rate of 0.08 was used to incorporate the contribution of each tree into the ensemble gradually, reducing the risk of overfitting while maintaining stable convergence. Row subsampling (subsample = 0.9) and feature subsampling (colsample_bytree = 0.9) introduced additional randomness during tree construction, reducing correlation between successive trees and improving robustness against overfitting [8]. Histogram-based tree construction (tree_method = hist) was used to reduce computational complexity while preserving predictive performance.

The hyperparameter configuration in Table 4.5 gives a moderate model complexity, reducing the risk of overfitting while keeping training efficient on the multimodal passenger delay dataset.

4.4.4 Long Short-Term Memory Model for Multimodal Passenger Delay Propagation

The same rationale presented in Section 4.2.2 for the single-mode LSTM applies here, since the underlying motivation for using a recurrent architecture remains unchanged. Passenger delay propagation is inherently sequential [6, 36], and LSTM networks can retain information from previous observations through their internal memory mechanism [36, 42]. In the multimodal setting, however, the sequential unit is not a series of consecutive stops within a single vehicle trip, but the ordered sequence of transfer events that make up a passenger journey [28]. Each prediction therefore depends not only on the current operational conditions, but also on how delay has evolved across the preceding journey segments, making LSTM networks well suited to modelling passenger delay propagation across multimodal journeys.

4.4.5 Input Representation

The LSTM model uses the same multimodal passenger dataset described in Section 3.10.2. Each observation corresponds to a transfer event within a passenger journey and contains the operational, temporal, multimodal, and contextual features selected during feature engineering. Unlike the XGBoost model, however, individual observations are not processed independently: observations belonging to the same passenger journey are grouped into ordered temporal sequences according to their transfer order, allowing the network to learn how passenger delay evolves throughout the complete journey.

4.4.6 Model Implementation

The proposed architecture consists of a single LSTM layer followed by two TimeDistributed fully connected layers, which generate delay predictions for every transfer event in the journey sequence. The LSTM layer contains 64 hidden units [24] and operates over the complete sequence of transfer events, preserving temporal dependencies through its internal memory state. Since delay propagation depends on operational conditions accumulated throughout the journey, this recurrent memory allows the model to retain relevant historical information across consecutive transfer events. The recurrent layer is followed by a TimeDistributed dense layer of 32 neurons with ReLU activation [6, 33], allowing further non-linear feature transformations to be applied independently at each point in the sequence. Finally, a second TimeDistributed dense layer with a single neuron produces the continuous passenger delay prediction for each transfer event.

4.4.7 Sequence Representation

Unlike tree-based algorithms, LSTM networks operate on ordered temporal sequences rather than independent observations. The multimodal passenger dataset was therefore reorganised before training so that each passenger journey forms one complete sequence. Transfer events sharing

Hyperparameter	Value	Purpose
LSTM units	64	Temporal modelling
Dense neurons	32	Nonlinear feature transformation
Output neurons	1	Delay prediction
Optimizer	Adam	Gradient optimisation
Loss function	Mean Squared Error	Regression objective
Batch size	128	Mini-batch learning
Maximum epochs	100	Training limit
Early stopping patience	10	Overfitting prevention

Table 4.6: LSTM model hyperparameter configuration.

the same *chain_id* were first grouped together, then ordered chronologically according to *transfer_index* [29]. This preserves the natural temporal evolution of the passenger journey, allowing the network to observe how delays accumulate from one transfer to the next. Prior to training, all numerical input variables were standardised [16, 24] using the mean and standard deviation computed exclusively from the training partition of each cross-validation fold.

Passenger journeys contain different numbers of transfer events, resulting in variable-length sequences. As neural networks require tensors of fixed dimension for mini-batch optimisation, all sequences were padded to the maximum sequence length observed in the dataset. Padding values were set to a constant outside the standardised feature range (PAD_VALUE = -999), so that they could be uniquely identified. A masking layer was then applied as the first network component: during both forward and backward propagation, masked positions are ignored by the recurrent network, preventing artificially padded observations from influencing the hidden states, loss computation, or parameter updates. The model therefore learns only from real transfer events, while still benefiting from efficient batch-based training [40].

4.4.8 Hyperparameter Configuration

The network was trained using the Adam optimisation algorithm [29, 33] with Mean Squared Error (MSE) as the loss function, appropriate for continuous regression problems [14, 40]. Training ran for a maximum of 100 epochs, using mini-batches of 128 passenger journeys. Early stopping was used to reduce overfitting by monitoring the validation loss, with training automatically halted if no improvement was observed for ten consecutive epochs. The model weights corresponding to the lowest validation loss were restored before final evaluation. Table 4.6 summarises the main hyperparameters used during training.

4.5 Graph Neural Network Model for Multimodal Passenger Delay Propagation

Unlike the single-mode formulation, where graph nodes represent transport stops, the multimodal framework represents each passenger journey as an independent graph, with nodes corresponding

to transfer events and edges describing the temporal succession of transfers within the journey.

4.5.1 Graph Representation

Each passenger journey was represented as an independent graph whose nodes correspond to consecutive transfer events belonging to the same passenger chain. Nodes were ordered according to the transfer index, and bidirectional edges were created between consecutive transfer events, allowing information to propagate in both temporal directions along the journey. Each node contains the complete engineered feature vector describing the operational, temporal, contextual, and multimodal characteristics of the corresponding transfer event. Prior to training, both input features and target values were standardised using statistics computed exclusively from the training partition [15] of each cross-validation fold. The Graph Neural Network therefore learns a node-level regression problem, predicting the passenger delay associated with every transfer event [11] while exploiting the graph connectivity between consecutive transfers.

4.5.2 Model Implementation

Unlike the LSTM, which models temporal dependencies through recurrent hidden states, the Graph Neural Network represents each passenger journey as a graph, allowing information to be exchanged between consecutive transfer events through message passing [21]. Each passenger chain corresponds to an independent graph, with nodes representing transfer events and edges encoding the temporal order of the journey. Each node carries the engineered feature vector described in Section 3.10.2, including historical delay information, transfer characteristics, contextual disruption indicators, and transport mode attributes. Consecutive transfer events within the same chain are connected through bidirectional edges, allowing information to propagate in both directions along the journey. The architecture consists of two graph convolutional layers (`GCNConv`) followed by a fully connected output layer [6, 35]. The first graph convolution projects the input feature representation into a 64-dimensional latent space, while the second reduces this to 32 latent features. A ReLU activation is applied after each graph convolution [6, 35], and a dropout layer with probability 0.2 is placed between the convolutional layers to reduce overfitting. A linear output layer then predicts the passenger delay associated with each transfer event. Prior to training, node features were standardised using the mean and standard deviation computed exclusively from the training graphs [15] of each cross-validation fold, and target values were normalised in the same way and converted back to their original delay units during evaluation. The network parameters were optimised using the Adam optimiser [29, 33] with a learning rate of 10^{-3}. The Huber loss function was adopted for its robustness to the extreme delay values present in the multimodal dataset, reducing the influence of severe outliers while keeping gradient updates stable. A *ReduceLROnPlateau* scheduler automatically reduced the learning rate whenever the validation loss stagnated, while gradient clipping with a maximum norm of 1.0 reduced the risk of unstable parameter updates. Early stopping with a patience of ten epochs was used to avoid overfitting, and model performance was evaluated using the same chain-aware 10-fold cross-validation protocol

Component	Configuration
Graph type	Passenger-chain graph
Node representation	Engineered feature vector
Edges	Bidirectional consecutive transfers
Graph convolution	GCNConv ($39 \rightarrow 64$)
Graph convolution	GCNConv ($64 \rightarrow 32$)
Output layer	Linear ($32 \rightarrow 1$)
Activation	ReLU
Dropout	0.2
Optimizer	Adam
Learning rate	10^{-3}
Loss function	Huber Loss
Scheduler	ReduceLROnPlateau
Gradient clipping	1.0
Maximum epochs	120
Early stopping patience	10 epochs
Evaluation	10-fold cross-validation

Table 4.7: GNN model configuration.

adopted for the other learning algorithms. Table 4.7 summarises the final GNN architecture and training configuration.

4.6 Evaluation Methodology

4.6.1 Experimental Configurations and Validation

To ensure a fair and consistent comparison between the evaluated models, all experiments followed the same training and evaluation protocol. Separate experiments were run for the single-mode and multimodal prediction tasks, using the corresponding datasets and prediction targets described in the previous chapters.

For the single-mode experiments, the objective was to predict the delay at the next stop of an individual vehicle trip. The multimodal experiments considered two complementary passenger-oriented prediction tasks: (i) predicting the passenger delay after the next transfer event [28] (*next-segment passenger delay*), and (ii) predicting the final passenger delay at the destination [2], after completing the entire multimodal journey (*final passenger delay*).

To obtain statistically robust performance estimates and reduce the variability associated with a single train-test split, all models were evaluated using 10-fold cross-validation [13, 21]. For each fold, the dataset was split into independent training and testing subsets, ensuring that every observation was evaluated exactly once. In the multimodal experiments, partitioning was performed at the passenger-chain level [15], so that all transfer events belonging to the same journey remained within the same fold, preventing information leakage between training and testing data.

For the LSTM and Graph Neural Network models, all feature normalisation was computed exclusively from the training subset of each fold and then applied to the corresponding validation

Metric	Purpose
MAE	Average prediction error (minutes)
R^2	Explained variance of the regression model
Accuracy	Overall classification performance
Precision	Correctness of predicted delay classes
Recall	Detection rate of delay classes
F1-score	Balance between precision and recall
95% CI	Statistical robustness across folds

Table 4.8: Evaluation metrics used in the experimental study.

and testing data [15]. This prevents information leakage while ensuring that model evaluation reflects unseen operational conditions accurately. Early stopping was used for the neural models, based on an independent validation subset drawn from the training data, allowing training to stop automatically once validation performance no longer improved [15]. The reported performance therefore corresponds to the mean and standard deviation obtained across the ten cross-validation folds [4], providing an estimate of both predictive accuracy and performance variability.

4.6.2 Evaluation Metrics

The predictive performance of the evaluated models was assessed using the regression and classification metrics summarised in Table 4.8. As public transport delay prediction is formulated as a regression problem, performance was primarily assessed using the Mean Absolute Error (MAE) and the coefficient of determination (R^2) [2, 3, 40]. The MAE measures the average magnitude of the prediction error in minutes, giving an intuitive measure of practical prediction accuracy. The coefficient of determination (R^2) quantifies the proportion of variance in the observed delays explained by the model, giving an overall measure of regression quality.

To aid operational interpretation, the continuous delay predictions were also discretised into three transport service categories: *early*, *on-time*, and *late* [18]. Based on these categories, the classification metrics Accuracy, Precision, Recall, and F1-score were computed, to evaluate each model's ability to correctly identify the operational service state [23].

Finally, to assess the statistical stability of the results, the mean, standard deviation, and corresponding 95% Confidence Interval (CI) were computed across the ten cross-validation folds for every evaluation metric [4, 6]. This provides further insight into the robustness and generalisation ability of the evaluated predictive models.

Chapter 5

Experimental Results and Discussion

5.1 Qualitative Inspection of the Multimodal Network

The interactive visualisation was used to qualitatively inspect the spatial and modal structure of the generated network. Figure 5.1 shows the interface, where routes can be filtered by transport mode and individual lines selected for closer inspection. This functionality was useful for checking whether the GTFS-derived network preserved the expected spatial structure of the public transport system. A further example of the route-level filtering functionality is provided in Appendix B, where a single tram route is selected and displayed together with its corresponding stops.

The interface also supports the identification of multimodal stops, defined as stops served by more than one transport mode. As shown in Figure 5.2, these stops can be highlighted and inspected together with the routes and modes serving them. This is particularly relevant for the proposed framework, since passenger delay propagation is strongly influenced by transfer opportunities between different services and modes. The visualisation therefore provides a qualitative check of the multimodal structure later used to construct passenger chains and transfer-based delay propagation features.

5.2 Contextual Data Analysis

5.2.1 Contextual Event Coverage

Figure 5.3 shows the distribution of contextual events across the multimodal transport observations. The results reveal a highly sparse contextual environment: the vast majority of transport segments (92.1%) are not associated with any nearby road closure or traffic-related event. Only 7.9% of observations are affected by nearby traffic events, and road closures are present in an even smaller share of the dataset (4.0%). Most passenger journeys therefore occur under normal operating conditions, while external disruptions remain relatively infrequent within the transport network. Although traffic-related events occur more often than road closures, both contextual sources remain highly imbalanced relative to the overall dataset. This is consistent with real-world public transport operations, where significant external disruptions typically occur only in



Figure 5.1: Interactive visualization interface for inspecting the Dutch multimodal public transport network.

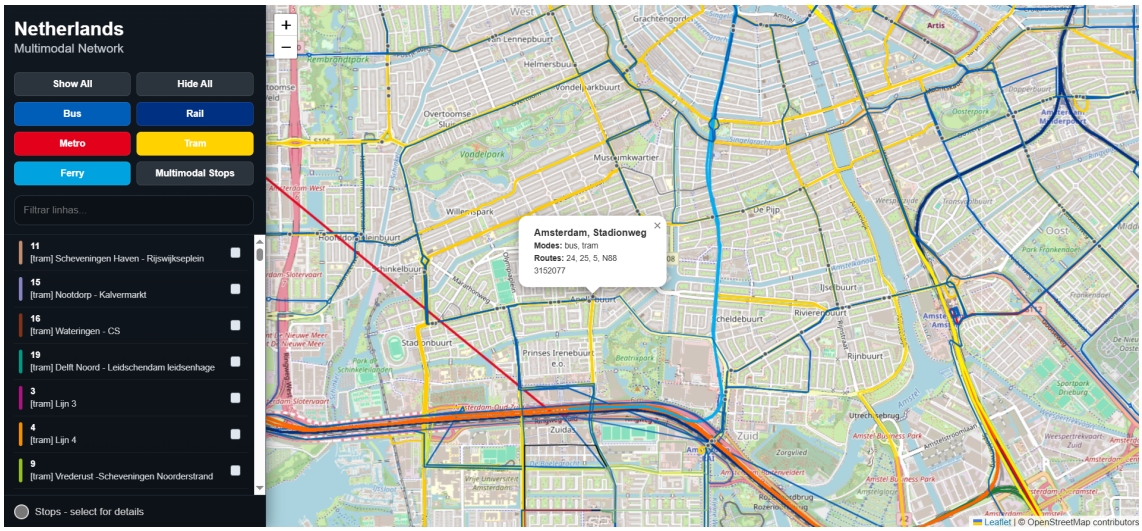


Figure 5.2: Identification of multimodal stops and associated transport services in the visualization interface.

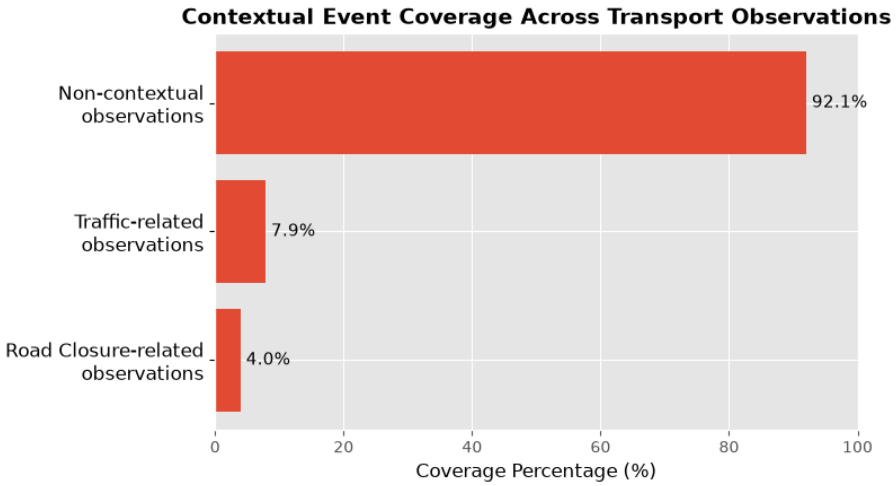


Figure 5.3: Contextual event coverage across observations

specific locations and during limited periods. As a result, contextual information is available for only a relatively small subset of observations, while most samples carry no external contextual signal. This imbalance has important implications for predictive modelling. Since contextual variables are absent from most observations, learning is dominated by operational variables describing vehicle behaviour and delay propagation, including historical delays, transfer penalties, missed connections, and other temporal characteristics. The models therefore have comparatively fewer opportunities to learn robust relationships between contextual disruptions and passenger delay evolution, naturally reducing the influence of contextual variables relative to propagation-related features, which are consistently available throughout the dataset.

This sparse distribution also helps explain the results of the contextual feature screening procedure. The relatively small changes in predictive performance observed after removing contextual variables should not be read as evidence that road closures and traffic events are irrelevant. Rather, they show that the available contextual signal is unevenly distributed and affects only a limited share of observations: under these conditions, contextual variables contribute to prediction only when disruptions are actually present, while the remaining observations rely mainly on operational and delay propagation patterns. Nevertheless, these variables were kept in the final feature set. Although their overall statistical contribution is lower than that of the dominant propagation-related variables, they represent external operational conditions that influence passenger journeys whenever disruptions occur. Retaining them therefore improves the realism, interpretability, and practical applicability of the proposed framework, allowing the models to make use of contextual information whenever it is available.

5.2.2 Contextual Event Intensity

While Figure 5.3 shows that contextual events are relatively uncommon across the dataset as a whole, Figure 5.4 examines only the transport segments associated with at least one nearby contextual event. The percentages in this figure are therefore computed exclusively over the subset

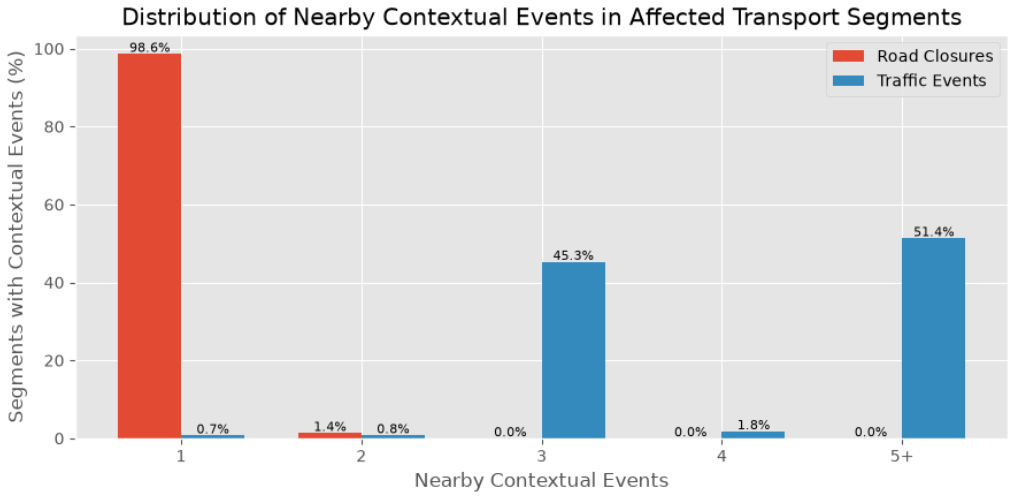


Figure 5.4: Distribution of nearby contextual events in affected transport segments, comparing the number of road closures and traffic events detected around each segment.

of affected observations, allowing the intensity of contextual disruptions to be examined independently of their overall frequency. A clear difference emerges between road closures and traffic-related events. Among segments affected by road closures, 98.6% are associated with a single nearby closure, and only 1.4% with two. No observations contain three or more nearby closures. Road closures therefore generally occur as isolated disruptions, affecting relatively small portions of the network rather than forming clusters of simultaneous events. Traffic-related events behave quite differently: only 0.7% of affected segments contain a single nearby event, while 45.3% are associated with three nearby events and 51.4% with five or more. This suggests that traffic disruptions frequently occur as groups of related events rather than isolated incidents, which is expected in real traffic management systems, where a single operational incident may generate multiple DATEX II records describing speed management measures, congestion, lane restrictions, or other related conditions within the same area. These differences help explain the distinct behaviour of the two contextual feature groups used in this work: although road closures affect fewer segments, they generally correspond to isolated events with relatively low local complexity, whereas traffic-related observations often capture several concurrent events within the same segment, providing a richer description of the surrounding conditions whenever contextual information is available.

Overall, the contextual information incorporated into the proposed framework is characterised not only by a sparse spatial distribution, but also by distinct structural properties: road closures typically appear as isolated events, whereas traffic disruptions tend to occur in clusters. This distinction supports the inclusion of multiple contextual descriptors, such as the number of nearby events and event-specific categories, allowing the models to capture different types of external operational conditions whenever they are present.

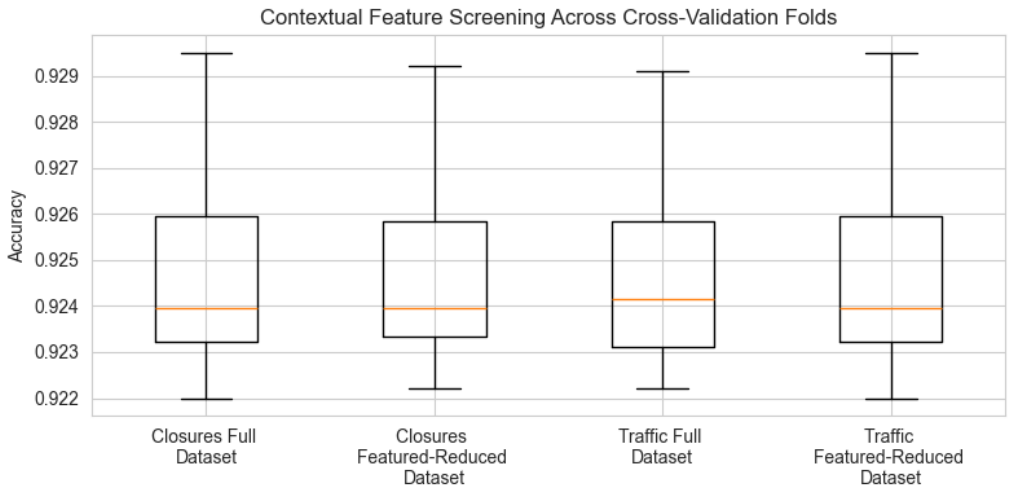


Figure 5.5: Cross-validation distributions after contextual feature screening.

5.3 Contextual Feature Screening

Figure 5.5 shows the cross-validation distributions obtained after selectively removing closure and traffic-related contextual variables from the XGBoost framework. The resulting distributions remain almost entirely overlapping across all evaluated configurations, indicating negligible practical differences in predictive performance, for both accuracy and MAE. This is consistent with the feature correlation and importance analyses, which showed that variables such as current delay, weighted delay history, cumulative delay, and delay variation consistently dominated the predictive process, confirming that future delays are primarily explained by recent delay evolution rather than external disruption information.

The limited predictive contribution of contextual variables can be attributed to three complementary factors: the high sparsity of contextual coverage, with most transport observations containing no nearby traffic events or road closures; the use of proximity-based spatial aggregation, which may not fully capture the operational relationship between traffic disruptions and affected transport services; and the structure of the synthetic delay generation process, which was explicitly designed around temporal propagation mechanisms, naturally increasing the dominance of propagation-related variables during training. Nevertheless, contextual integration remains important from a modelling perspective, as it contributes to a more semantically complete representation of real-world operational environments, and may become increasingly relevant under denser or fully real-world disruption conditions.

5.4 Single-Mode Delay Prediction

5.4.1 Experimental Setup

The single-mode experiments followed the evaluation protocol described in Section 4.6 of Chapter 4. The dataset used was the single-mode dataset described in Chapter 3, where each observa-

Metric	Mean ± Std.	95% CI
MAE	0.414 ± 0.003	[0.412, 0.415]
Accuracy	0.925 ± 0.002	[0.923, 0.926]
Precision	0.924 ± 0.003	[0.923, 0.926]
Recall	0.925 ± 0.002	[0.923, .926]
F1-Score	0.924 ± 0.002	[0.922, 0.925]

Table 5.1: XGBoost performance obtained using 10-fold cross-validation.

tion corresponds to an individual vehicle stop and the prediction target is the delay at the next stop along the same trip. As in the general protocol, performance is reported as the mean across the ten cross-validation folds, together with the standard deviation and 95% CI, for both the regression metric (MAE) and the classification metrics derived from the *early*, *on time*, and *late* delay categories.

5.4.2 XGBoost Results

The predictive performance of the XGBoost model was evaluated using 10-fold cross-validation. Regression performance was assessed using the MAE, while the predicted delays were also converted into qualitative delay classes to compute Accuracy, Precision, Recall, and F1-score. The complete results for each fold are given in Appendix B. The discussion below focuses on the aggregated mean, standard deviation, and 95% confidence intervals.

XGBoost achieved consistently strong predictive performance across all folds, with an average MAE of 0.414 minutes and an overall classification accuracy of 96.4%. Precision, Recall, and F1-score all exceeded 96%, showing that the model accurately predicts both the numerical delay values and their corresponding qualitative categories. Performance variability across folds was limited, as reflected by the low standard deviations for all metrics. The MAE, in particular, had a standard deviation of only 0.009 minutes, while the classification metrics varied by less than 0.5 percentage points. Inspection of the individual fold results in Appendix B confirms this consistency, with no fold showing substantial degradation in performance. The 95% confidence intervals reinforce the robustness of these results: classification accuracy is expected to lie between 96.2% and 96.6%, and the MAE within a narrow interval between 0.412 and 0.415 minutes. The small confidence intervals indicate that the reported performance is stable and unlikely to result from a particularly favourable data partition.

Overall, these results show that XGBoost provides a strong and reliable baseline for public transport delay prediction. Its ability to exploit the engineered operational, temporal, structural, and contextual features enables accurate modelling of local delay propagation patterns while maintaining generalisation to unseen observations. The XGBoost model therefore serves as the reference against which the temporal (LSTM) and spatial (GNN) paradigms are compared in the following sections.

Metric	Mean ± Std.	95% CI
MAE	1.231 ± 0.217	[1.097, 1.366]
Accuracy	0.935 ± 0.013	[0.927, 0.943]
Precision	0.931 ± 0.014	[0.922, 0.940]
Recall	0.935 ± 0.013	[0.927, 0.943]
F1-Score	0.930 ± 0.014	[0.921, 0.939]

Table 5.2: LSTM performance obtained using 10-fold cross-validation.

5.4.3 LSTM Results

The predictive performance of the LSTM model was evaluated using the same 10-fold cross-validation procedure adopted for XGBoost. Regression performance was assessed using the MAE, and classification metrics were computed after converting the predicted delay values into qualitative categories. The complete results for each fold are given in Appendix B. This section discusses the aggregated statistics.

The LSTM model achieved good predictive performance across the ten folds, with an average MAE of 1.231 minutes and an overall classification accuracy of 93.5%. Precision, Recall, and F1-score all remained above 93%, indicating that the network successfully captured the temporal evolution of transport delays and correctly identified the qualitative delay category for most observations. Compared with the XGBoost baseline, however, the LSTM showed lower predictive accuracy and greater variability across folds: the average MAE increased from 0.414 to 1.231 minutes, while classification accuracy fell from 96.4% to 93.5%. The standard deviation of the MAE also increased substantially (0.217, against 0.009 for XGBoost), showing that the recurrent model was considerably more sensitive to the specific training and validation partitions. Although LSTM networks are specifically designed to model temporal dependencies, these results suggest that the engineered delay history features already provide a strong representation of temporal information. The additional sequential modelling capability of the LSTM therefore does not translate into improved performance for the single-mode delay prediction task. Instead, the tree-based XGBoost model appears to exploit the engineered feature space more effectively, achieving both higher accuracy and greater robustness. Even so, recurrent networks remain suitable for modelling longer sequential dependencies, so the same architecture is evaluated again in the multimodal passenger journey setting, where considerably longer temporal sequences are available.

5.4.4 GNN Results

The Graph Neural Network was evaluated using the same 10-fold cross-validation procedure adopted for the previous models. The complete results for each fold are given in Appendix B, while this section presents the aggregated metrics and their implications in Table 5.3. The GNN achieved an average MAE of 0.775 minutes and an overall classification accuracy of 99.2%; Precision, Recall, and F1-score also exceeded 98.9%, apparently outperforming both XGBoost and the LSTM. These classification results, however, should be interpreted with caution. A detailed

Metric	Mean ± Std.	95% CI
MAE	0.775 ± 0.074	[0.729, 0.8205]
Accuracy	0.992 ± 0.006	[0.988, 0.995]
Precision	0.991 ± 0.006	[0.987, 0.995]
Recall	0.992 ± 0.006	[0.988, 0.995]
F1-score	$.990 \pm 0.008$	[0.985, 0.995]

Table 5.3: GNN performance obtained using 10-fold cross-validation.

inspection of the individual folds and the corresponding confusion patterns (Appendix B) reveals a severe class imbalance, with several validation folds containing almost exclusively observations from the *late* category and, in some cases, only a single observation from the remaining classes. Under these conditions, a classifier can achieve near-perfect performance simply by predicting the dominant class, so the observed accuracy, precision, recall, and F1-score are substantially shaped by the underlying class distribution and do not exclusively reflect the model's ability to learn meaningful spatial delay propagation patterns.

Overall, the results suggest that the adopted GNN architecture can exploit the transport network topology to produce stable predictions. However, the severe class imbalance in the single-mode dataset substantially inflates the classification metrics, making direct comparison with the other models difficult. These observations motivated the redesign of the graph representation and the move to the multimodal passenger propagation framework presented in the following sections, where richer graph structures and more informative propagation mechanisms are introduced. Although Graph Neural Networks are naturally suited to transport networks, the graph adopted in the single-mode dataset represents only consecutive stop connections within individual vehicle trips, providing relatively limited spatial information compared with the richer multimodal graph developed later in this work.

5.4.5 Single-Mode Model Performance Comparison

The predictive performance of the three learning paradigms is summarised in Figures 5.6 and 5.7, with the complete numerical results reported in Table 5.4. The analysis compares not only the average performance of each model, but also its robustness across the ten folds.

XGBoost consistently achieved the strongest overall performance, obtaining the lowest prediction error (MAE = 0.414 minutes) while maintaining high classification performance, with an average accuracy of 96.4% and an F1-score of 96.3%. The compact distributions observed in both

Model	MAE	Accuracy	Precision	Recall	F1-Score
XGBoost	$\mathbf{0.414 \pm 0.003}$	0.925	0.924	0.925	0.924
LSTM	1.231 ± 0.217	0.935	0.931	0.935	0.930
GNN	0.775 ± 0.074	**0.992**	**0.991**	**0.992**	**0.990**

Table 5.4: Comparison of the predictive performance obtained by the evaluated single-mode models using 10-fold cross-validation.

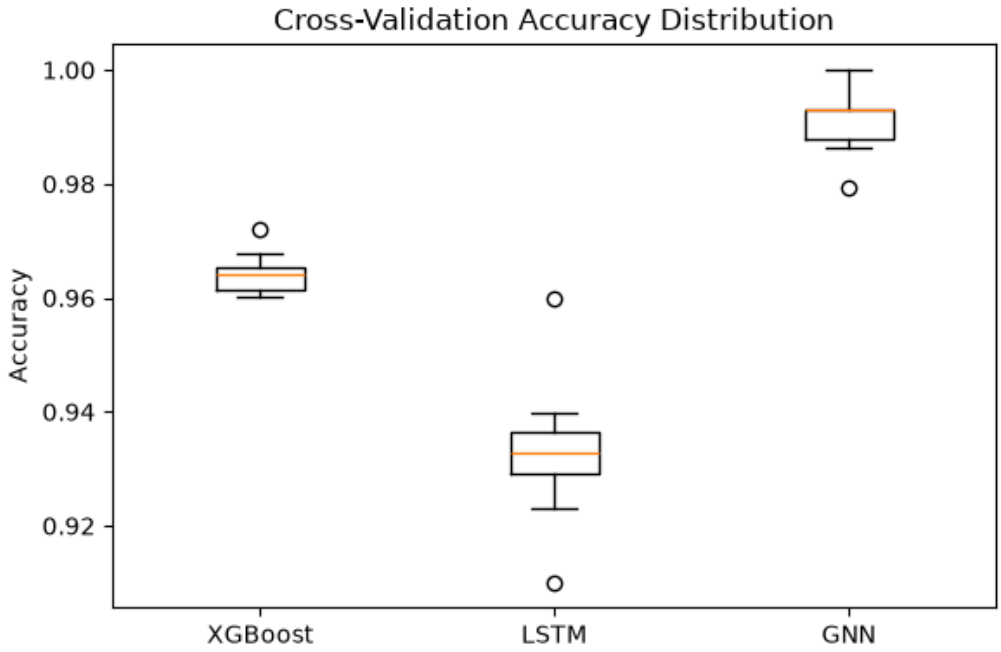


Figure 5.6: Cross-validation accuracy distributions

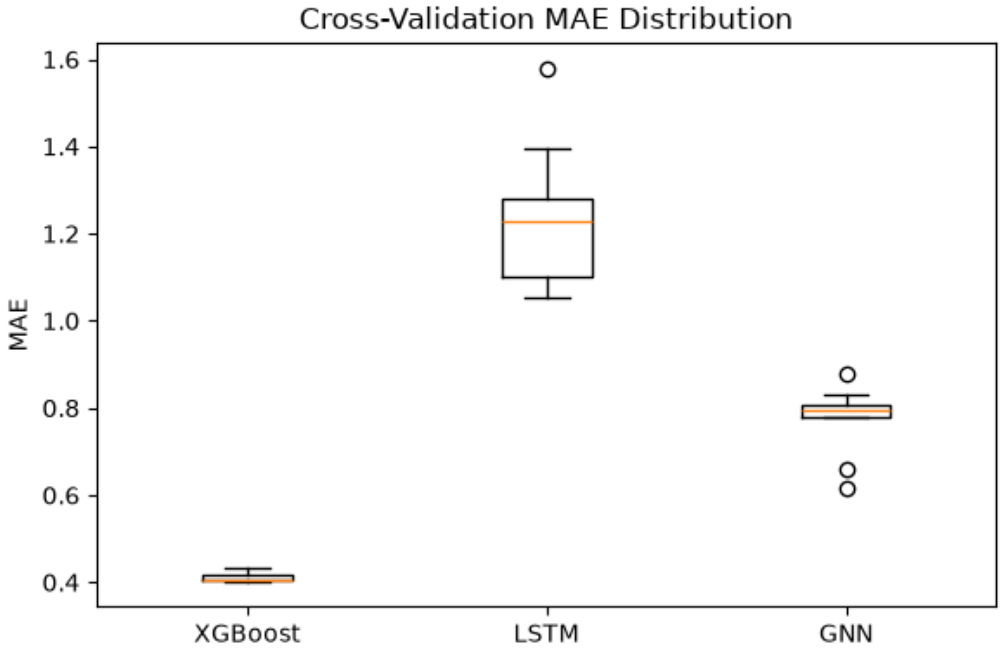


Figure 5.7: Cross-validation MAE distributions.

the accuracy and MAE boxplots indicate stability across folds, suggesting that the model generalises consistently regardless of the particular training-validation partition. This shows that the engineered feature space successfully captures the dominant operational and propagation-related factors driving delay evolution, allowing the tree-based algorithm to exploit the available information effectively.

The LSTM model achieved the lowest predictive performance among the three approaches. Although recurrent networks are specifically designed to model temporal dependencies, the results indicate that their additional sequential modelling capability provided only limited benefit under the adopted experimental setting. The model produced the highest average prediction error (MAE = 1.231 minutes) together with the largest variability across folds, indicating lower robustness and greater sensitivity to the training partition. One possible explanation is that much of the temporal information required to predict future delays had already been explicitly encoded into the feature space through propagation-aware variables such as weighted delay history, cumulative delay, and delay variation. The sequential memory mechanism of the LSTM therefore contributed relatively little additional predictive information beyond what was already captured by the input features.

The Graph Neural Network obtained the highest classification metrics, with an average accuracy of 99.2% and an F1-score of 99.0%, but, as discussed in Section 5.4.4, these values are inflated by the severe class imbalance present in the single-mode dataset and should not be directly compared with the other models on that basis. The regression results offer a more reliable comparison: despite its near-perfect classification performance, the GNN produced a substantially higher MAE (0.775 minutes) than XGBoost (0.414 minutes), confirming that its numerical delay estimates remain less accurate than those of the tree-based approach. Together with the LSTM results discussed above, this indicates that, for the single-mode delay prediction problem, the engineered propagation-aware feature space already captures most of the relevant temporal and operational information, allowing XGBoost to outperform both deep learning architectures evaluated in this work.

Overall, XGBoost provides the best balance between predictive accuracy, numerical precision, and robustness among the three evaluated models, consistently achieving the lowest prediction error while maintaining stable performance across all folds. Additional visual summaries of the average performance metrics are provided in Appendix B, where bar charts compare the MAE, accuracy, precision, recall, and F1-score obtained by each model.

5.5 Multimodal Passenger Delay Prediction

5.5.1 Problem Formulation

The multimodal prediction problem extends conventional vehicle delay prediction by estimating the delay experienced by passengers throughout complete multimodal journeys. Unlike the single-mode scenario, predictions depend not only on the operational state of individual vehicles, but also on transfer availability, missed connections, delay propagation across successive journey

Property	Single-mode	Multimodal
Prediction target	Next stop delay	Final passenger delay
Prediction scope	Single vehicle trip	Complete passenger journey
Transport network	Single route	Multiple interconnected routes
Transfers considered	No	Yes
Delay propagation	Within a single vehicle	Across successive vehicles and transfers
Passenger perspective	Vehicle-centred	Passenger-centred
Spatial representation	Individual stops	Journey segments and transfer points
Context integration	Stop-level contextual events	Segment-level contextual events
Prediction complexity	Lower	Higher

Table 5.5: Comparison between the single-mode and multimodal prediction problems.

segments, and external contextual disruptions. The multimodal formulation therefore introduces additional sources of uncertainty, while providing a more realistic representation of passenger travel behaviour.

5.5.2 Experimental Setup

Model performance was evaluated using 10-fold cross-validation, to obtain robust estimates while reducing the influence of any particular train-test split. The same folds were used for all models, to ensure a fair comparison.

Three prediction models were evaluated: XGBoost, LSTM, and GNN. Hyperparameters were selected through preliminary experiments and kept fixed across all folds. The XGBoost model used 300 estimators with a maximum depth of 8 and a learning rate of 0.05. The LSTM consisted of two stacked LSTM layers (128 and 64 units), followed by dropout regularisation and a dense output layer. The GNN used two graph convolutional layers with 64 hidden units and was trained for 120 epochs using the Adam optimiser.

Performance was assessed using the MAE as the primary regression metric. To facilitate comparison with previous studies, continuous predictions were also converted into delay categories (early, on-time, and late), from which Accuracy, Precision, Recall, and F1-score were computed. To ensure reproducibility, a fixed random seed was used throughout data splitting, model initialisation, and training, wherever applicable.

5.5.3 XGBoost Results

5.5.3.1 Predictive Performance

Table 5.6 summarises the predictive performance of the XGBoost model for both passenger delay prediction objectives, using ten-fold cross-validation. Performance is reported as the average across folds, together with the corresponding 95% confidence intervals. The complete results for each fold are given in Appendix B. The relatively narrow confidence intervals show that the model behaves consistently across different subsets of passenger journeys, indicating good generalisation and limited sensitivity to the particular partitioning used during cross-validation.

Metric	Next-Segment Delay	Final Journey Delay
MAE	3.6861 [3.6647-3.7074]	8.1113 [8.0129-8.2097]
R^2	0.8902 [0.8844-0.8961]	0.7356 [0.7275-0.7436]
Accuracy	0.9474 [0.9464-0.9484]	0.9213 [0.9190-0.9236]
Precision	0.9315 [0.9296-0.9334]	0.9046 [0.9018-0.9075]
Recall	0.9474 [0.9464-0.9484]	0.9213 [0.9190-0.9236]
F1-score	0.9375 [0.9361-0.9389]	0.9115 [0.9091-0.9139]

Table 5.6: XGBoost predictive performance obtained using 10-fold cross-validation. Values are reported as mean [95% CI].

Prediction of Immediate Passenger Delay

For predicting the passenger delay immediately after the next journey segment (*target_ passenger_delay_after*), the model achieved a mean absolute error of 3.69 minutes, while correctly classifying passenger delay states with an average accuracy of 94.74% and an F1-score of 93.75%. The corresponding coefficient of determination ($R^2 = 0.890$) shows that the model explains approximately 89% of the observed variance in passenger delays. These results show that the proposed feature representation successfully captures the operational conditions that most directly affect passenger delay propagation, including vehicle delays, transfer characteristics, historical delay evolution, and contextual disruption information.

Prediction of Final Passenger Delay

Predicting the final accumulated passenger delay at the destination (*target_final_passenger_ delay*) is a more complex regression problem, since the model must estimate the cumulative effect of all remaining journey segments and transfer opportunities after the current prediction instant. The prediction error accordingly increases to a mean absolute error of 8.11 minutes, with a coefficient of determination of 0.736 and an F1-score above 91%. Despite the longer prediction scope and the greater uncertainty associated with future operational events, the model maintains a high level of predictive performance and stable behaviour across all folds.

Comparative Analysis

Overall, XGBoost provides accurate predictions for both multimodal targets while remaining stable across folds. Although prediction accuracy naturally decreases for the final passenger delay, due to the longer prediction scope, the results show that the proposed multimodal feature representation successfully captures the operational and contextual factors driving passenger delay propagation.

5.5.3.2 Prediction Analysis

Figures 5.8 and 5.9 compare the predicted and observed passenger delays for both prediction targets. The diagonal dashed line represents the ideal prediction, where estimated and observed delays coincide. Most observations are concentrated around this reference line, indicating that the XGBoost model provides accurate predictions for the majority of passenger journeys.

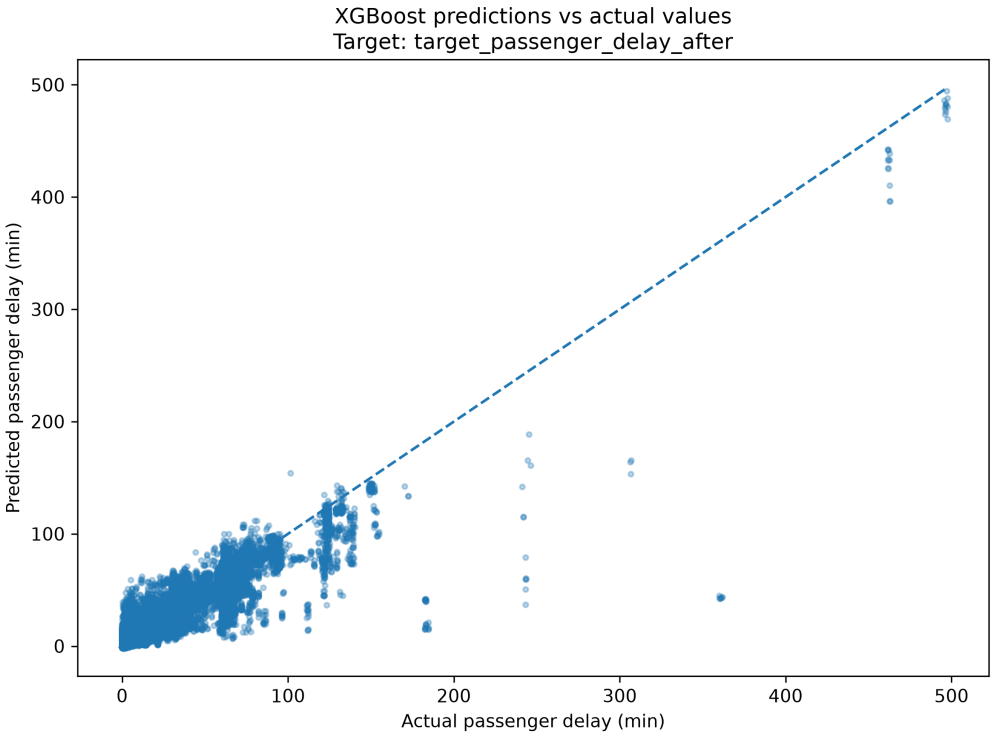


Figure 5.8: XGBoost prediction scatter plot for *target_passenger_delay_after*, showing strong alignment for most low-to-moderate passenger delays and larger deviations for rare extreme delay cases.

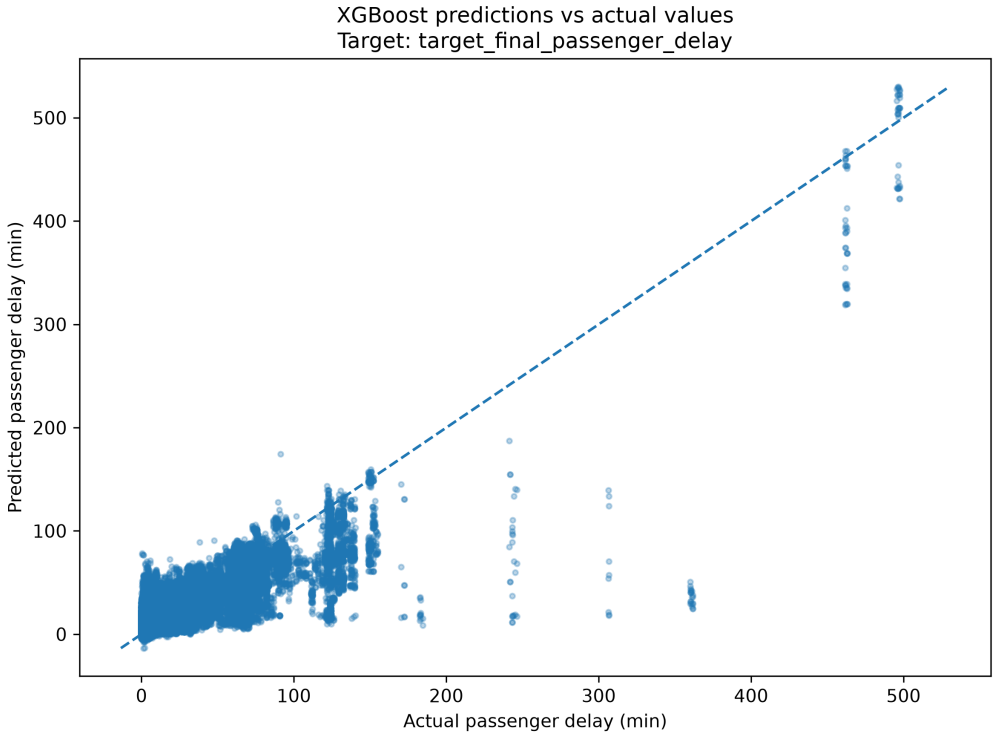


Figure 5.9: XGBoost prediction scatter plot for *target_final_passenger_delay*, showing the relationship between actual and predicted final passenger delays. The dashed line represents the ideal prediction line.

Figure 5.8 compares the predicted passenger delays with the corresponding ground-truth values. Most observations cluster around the diagonal reference line, indicating that the model accurately predicts the majority of passenger delay values. The highest prediction accuracy is observed for journeys with low and moderate delays, which make up the largest share of the generated multimodal dataset. As passenger delay increases, the prediction variance also grows, reflecting the increasing uncertainty associated with long chains of propagated delay. For extremely large propagated delays, the model tends to underestimate the observed values. These cases correspond to rare operational scenarios where missing a connection results in exceptionally long waiting times before the next scheduled departure, producing propagated delays substantially larger than those typically observed in the dataset.

Figure 5.9 corresponding to *target_final_passenger_delay* shows greater dispersion, particularly for journeys with large accumulated delays. This is expected, since the model must estimate the combined effect of all remaining journey segments, making the prediction substantially more uncertain. Nevertheless, the overall distribution remains concentrated around the ideal prediction line for the vast majority of observations, confirming that the model generalises well to realistic multimodal passenger journeys.

Feature	Next-Segment Delay	Final Journey Delay
Weighted passenger delay history	0.2379	0.2075
Mode pair (bus → bus)	0.1231	0.1229
Planned waiting time	0.0953	0.0595
Current vehicle delay	0.0669	0.0466
Initial vehicle delay	0.0473	0.0838
Number of traffic events	0.0475	0.0353
Destination mode: rail	0.0432	0.0017
Planned journey duration	0.0402	0.0785
Weighted vehicle delay history	0.0375	0.0396
Intended departure time	0.0287	0.0320
Transfer index	0.0282	0.0239
Number of transport modes	0.0239	0.0706
Peak-hour indicator	0.0230	0.0366
Start hour	0.0182	0.0358
Number of transfers	0.0108	0.0606

Table 5.7: Average XGBoost feature importance obtained over the 10 cross-validation folds for both prediction targets. Only the fifteen most relevant features are presented.

5.5.3.3 Feature Importance

Table 5.7 presents the average feature importance across the ten cross-validation folds for both prediction targets. The reported values correspond to the average importance assigned by XGBoost during tree construction, reflecting the relative contribution of each feature to the reduction in prediction error throughout the ensemble.

For both tasks, *weighted_passenger_delay_history* is consistently the most influential predictor, confirming that the recent temporal evolution of passenger delays carries substantially more predictive information than a single instantaneous delay observation. Rather than relying solely on the current operational state, the model benefits from a compact representation of the delay propagation process experienced throughout the previously completed journey segments.

For *target_passenger_delay_after*, the most influential variables after passenger delay history are the current transport mode, planned waiting time, current vehicle delay, and contextual traffic information, variables that directly describe the operational conditions immediately preceding the next transfer, and strongly influence the passenger delay accumulated after the following segment. Predicting the final accumulated passenger delay, by contrast, requires modelling the evolution of the entire remaining journey, so variables describing the overall journey structure become considerably more important: planned journey duration, number of transport modes, number of transfers, and the initial vehicle delay all show substantially higher importance than for the immediate delay prediction task, reflecting their influence on the long-term propagation of delays across multiple segments.

Contextual disruption indicators extracted from DATEX II, particularly the number of traffic events and road speed restrictions, also contribute to both tasks, though with lower relative importance than the operational delay variables. This is expected, since contextual events influence

Feature	Mean |SHAP|
Planned journey duration	9.586
Weighted passenger delay history	4.728
Planned waiting time	4.079
Intended departure time	2.290
Initial vehicle delay	1.980
Transfer index	1.937
Number of transport modes	1.869
Number of transfers	1.773
Start hour	1.324
Current vehicle delay	1.167
Peak-hour indicator	0.505
Weighted vehicle delay history	0.318
Number of traffic events	0.215
Number of road closures	0.164
Mode pair (bus $\rightarrow$ bus)	0.109

Table 5.8: Average SHAP feature importance obtained for the final journey delay prediction task. Values correspond to the mean absolute SHAP values computed over 30,000 sampled observations across the ten cross-validation folds. Only the fifteen most relevant features are presented.

passenger delays indirectly through their impact on vehicle operations. Several variables show negligible or zero importance, including obstruction events, closure intensity, and some infrequent multimodal transfer combinations. This does not necessarily mean these variables are irrelevant. Rather, it reflects how infrequently these situations occur in the dataset, which limits their contribution during model optimisation.

Overall, the obtained feature importance rankings are consistent with the proposed passenger-centred modelling framework. Historical passenger delay remains the dominant predictor, while operational transfer characteristics, vehicle delays, and journey-level descriptors become progressively more important as the prediction scope extends from the next journey segment to the complete multimodal trip.

5.5.3.4 SHAP Analysis of Feature Contributions

To complement the gain-based feature importance reported in Table 5.7, a SHapley Additive exPlanations (SHAP) analysis was conducted for the XGBoost model predicting the final journey delay, using the same final feature set described in Table 3.4. SHAP values were computed using TreeExplainer on 30,000 instances sampled across the ten cross-validation folds, and the mean absolute SHAP value of each feature was used as a global importance measure.

Table 5.8 and Figure 5.10 present the fifteen most relevant features. The ranking is consistent with the gain-based importance discussed in Section 5.5.3.3 and with the preliminary feature selection analysis of Section 3.11.1.1: journey-level structural variables, namely the planned journey duration, the weighted passenger delay history, and the planned waiting time, dominate the

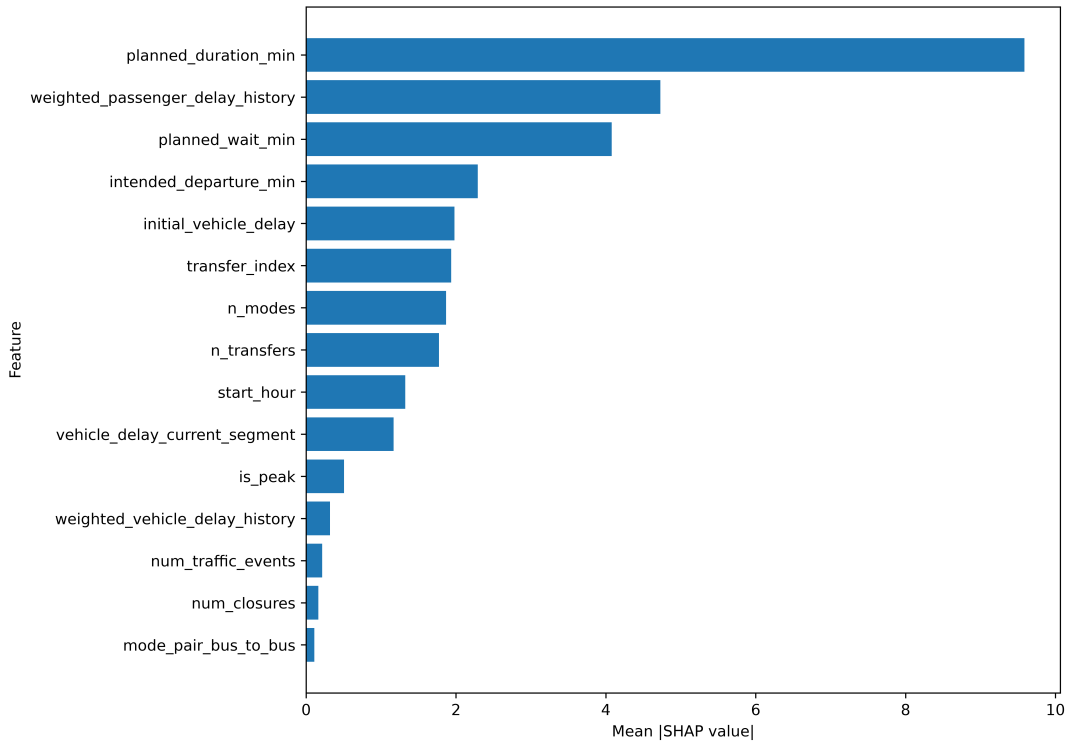


Figure 5.10: SHAP feature importance summary for the final journey delay prediction model, showing the mean absolute SHAP value for each feature across the sampled test instances.

prediction of the final accumulated delay, confirming that the overall structure of the remaining journey is the primary factor explaining long-horizon delay propagation.

Contextual disruption indicators extracted from DATEX II, namely the number of nearby traffic events (mean $|\text{SHAP}| = 0.215$) and the number of nearby road closures (mean $|\text{SHAP}| = 0.164$), rank considerably lower than the dominant propagation and journey-structure variables, and the remaining DATEX II-derived features contribute negligibly to the global explanation. This result is consistent with the contextual event coverage reported in Section 5.2.1, where 92.1% of the observations in the dataset are not associated with any nearby road closure or traffic event. Because SHAP values are averaged over the entire evaluation set, features that are only defined or non-zero for a small subset of observations necessarily obtain a low global importance score, even if they carry meaningful information for the specific instances in which a disruption is actually present. Low mean $|\text{SHAP}|$ values for contextual features should therefore be interpreted as a consequence of data sparsity rather than as direct evidence that DATEX II information is uninformative whenever a disruption occurs.

5.5.3.5 Error Analysis

Although the XGBoost model achieves high predictive performance for the vast majority of passenger journeys, the largest prediction errors are concentrated in a relatively small number of observations with exceptionally large accumulated delays. Manual inspection of the corresponding

Metric	Next-Segment Delay	Final Journey Delay
MAE	2.855 [2.798-2.913]	6.129 [6.059-6.200]
R^2	0.930 [0.930-0.933]	0.815 [0.809-0.822]
Accuracy	0.931 [0.930-0.935]	0.932 [0.928-0.935]
Precision	0.933 [0.930-0.940]	0.905 [0.902-0.908]
Recall	0.931 [0.926-0.935]	0.932 [0.928-0.935]
F1-Score	0.931 [0.928-0.934]	0.915 [0.913-0.917]

Table 5.9: LSTM predictive performance obtained using 10-fold cross-validation. Values are reported as mean [95% CI].

passenger chains showed that these cases are mainly associated with successive missed connections and unusually long waiting times caused by low service frequencies, rather than modelling inconsistencies. These extreme situations result when a passenger misses one or more planned connections and, under the modelling assumptions adopted in this work, remains on the originally planned journey: the passenger then waits for the next available service on the same route, which may only depart several hours later. As delay propagation continues across later segments, the accumulated passenger delay can increase substantially, producing the largest regression errors observed during evaluation. Importantly, these observations correspond to realistic operational scenarios rather than anomalies or data quality issues, and were therefore intentionally preserved during dataset construction, rather than removed as statistical outliers or artificially truncated. Retaining these rare but operationally meaningful situations allows the proposed framework to represent realistic multimodal delay propagation behaviour, even though such cases remain considerably harder to predict than typical passenger journeys.

5.5.4 LSTM Results

5.5.4.1 Prediction Performance

Table 5.9 summarises the predictive performance of the LSTM model. The complete results for each fold are given in Appendix B. The network achieved high prediction accuracy for both prediction objectives while remaining consistent across folds, indicating stable generalisation to previously unseen passenger journeys.

For the accumulated passenger delay immediately after the next transfer, the model achieved a mean MAE of approximately 2.86 minutes, substantially improving on the XGBoost baseline. The corresponding R^2 exceeded 0.93, showing that the recurrent architecture captures most of the temporal variability associated with passenger delay propagation.

Predicting the final passenger delay remains more demanding, since the model must estimate the cumulative effect of all remaining journey segments after the current transfer. Even so, the LSTM achieved a mean MAE of 6.13 minutes, with an accuracy of 93.16% and an F1-score of 91.48%, outperforming the corresponding XGBoost model on every reported metric.

The relatively narrow confidence intervals across the ten folds indicate consistent predictive performance and good generalisation to unseen passenger journeys.

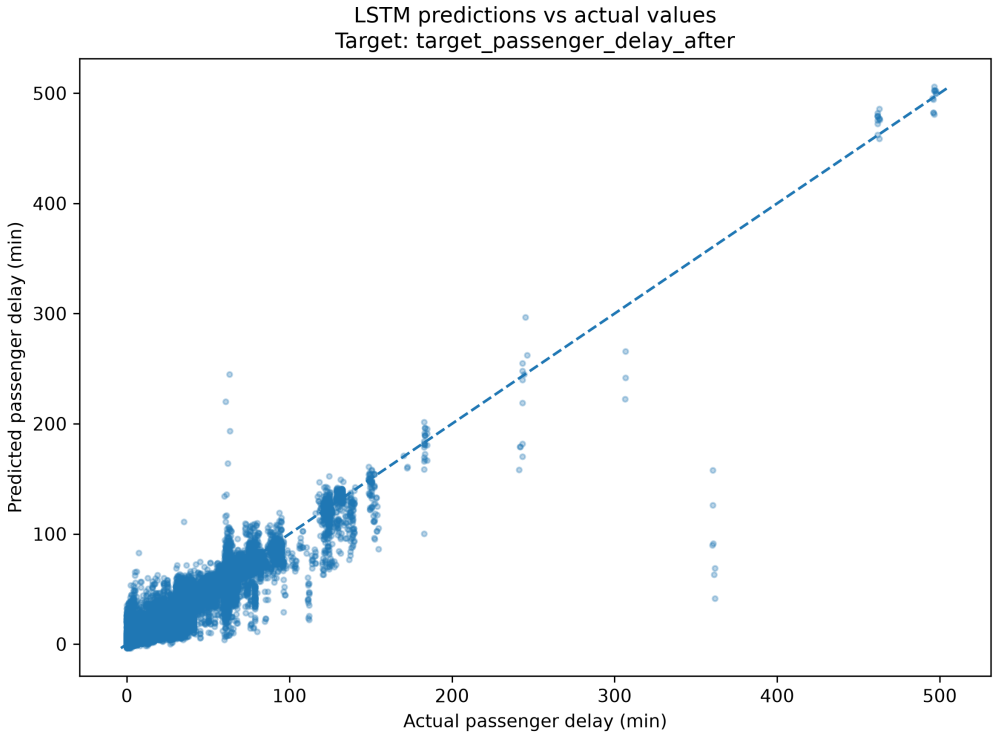


Figure 5.11: Predicted versus observed next-segment passenger delays obtained by the LSTM model.

5.5.4.2 Prediction Analysis

Figure 5.11 shows the relationship between predicted and observed passenger delays after the next journey segment. Compared with the XGBoost baseline, the predictions are noticeably more concentrated around the identity line, indicating better alignment between predicted and actual delay values, consistent with the lower MAE and higher R^2 reported in Table 5.9.

The recurrent architecture effectively captures the temporal evolution of passenger delays across successive transfer events, giving more accurate predictions throughout the most common delay ranges. Although the largest prediction errors are still associated with rare severe disruption events involving multiple missed connections and prolonged waiting, the LSTM predicts substantially larger delay values than XGBoost in these cases, showing an improved representation of long-term delay propagation.

Figure 5.12 shows the corresponding results for the final passenger delay. Despite the greater difficulty of estimating the cumulative effect of future transfers and delay propagation until the destination, most observations remain closely distributed around the identity line, indicating that the LSTM successfully models long-term temporal dependencies throughout multimodal journeys.

Compared with XGBoost, the LSTM produces a noticeably tighter prediction distribution across the most common delay ranges, while also giving more accurate estimates for severe disruption events. Although the largest accumulated delays are still systematically underestimated, the recurrent architecture captures the overall growth trend considerably better than the tree-based

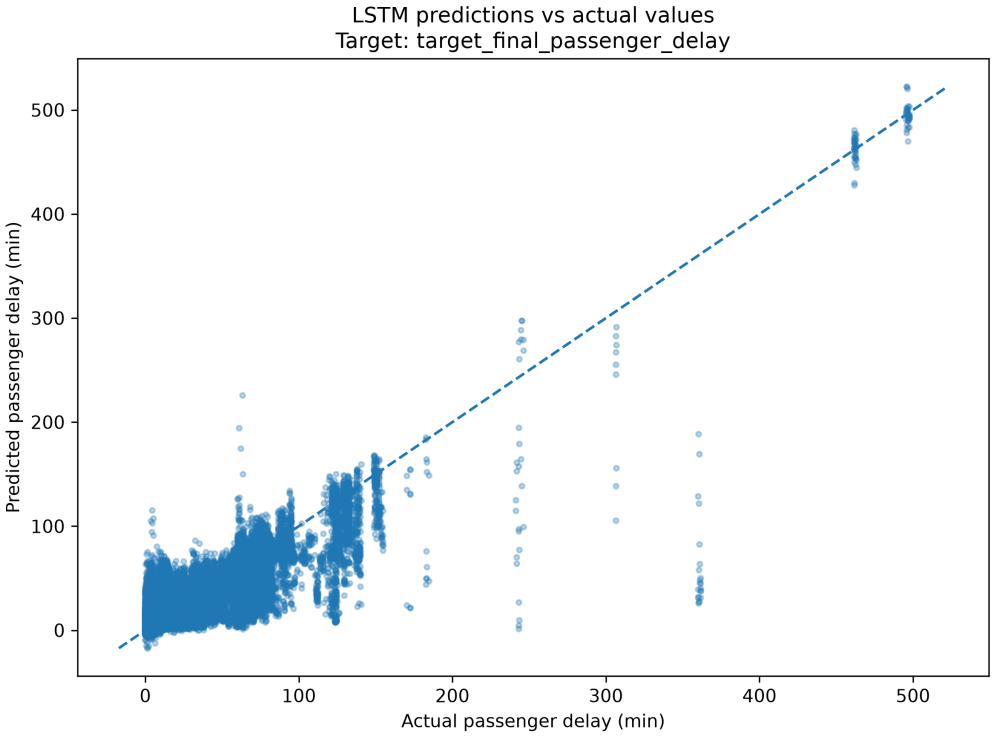


Figure 5.12: Predicted versus observed final passenger delays obtained by the LSTM model.

baseline, explaining the substantial reduction in MAE from 8.11 to 6.13 minutes. Overall, the scatter plot confirms that explicitly modelling the temporal evolution of passenger delays allows more accurate prediction of cumulative delay propagation across multimodal journeys. The remaining errors are mainly associated with rare extreme disruption events and are analysed further in Section 4.4.6.

5.5.5 GNN Results

5.5.5.1 Predictive Performance

The proposed GNN achieved strong predictive performance for the next-segment passenger delay task. As shown in Table 5.10, the model obtained an average MAE of 4.084 minutes, with an over-

Metric	Next-Segment Delay	Final Journey Delay
MAE	4.0841 [4.0267-4.1414]	8.5655 [8.4394-8.6917]
R^2	0.7605 [0.7474-0.7735]	0.6201 [0.6092-0.6310]
Accuracy	0.9477 [0.9456-0.9497]	0.9355 [0.9336-0.9374]
Precision	0.9350 [0.9316-0.9384]	0.9018 [0.9001-0.9034]
Recall	0.9477 [0.9456-0.9497]	0.9355 [0.9336-0.9374]
F1-Score	0.9385 [0.9359-0.9411]	0.9148 [0.9129-0.9168]

Table 5.10: GNN predictive performance obtained using 10-fold cross-validation. Values are reported as mean [95% CI].

all classification accuracy of 94.8%; Precision, Recall, and F1-score all exceeded 93%, indicating that the graph representation successfully captures the spatial relationships between consecutive transfer events. The complete results for each fold are given in Appendix B

Performance remained consistent across folds, with relatively small standard deviations and narrow confidence intervals for all metrics: the MAE varied by less than 0.1 minutes across folds, while the classification metrics varied by less than one percentage point, indicating stable generalisation across data partitions. Although competitive, the GNN does not outperform XGBoost on this task. Its regression error remains considerably higher than the tree-based baseline, suggesting that the engineered propagation-aware features already capture much of the information required to estimate the passenger delay after the next transfer, so the additional spatial aggregation performed by the graph convolutions provides only limited improvement.

The GNN was then evaluated using the final passenger delay as the prediction target. Table 5.10 summarises the aggregated performance over the ten folds. Compared with the next-segment task, the final-delay prediction problem proved considerably harder: the average MAE increased from 4.084 to 8.566 minutes, while R^2 decreased from 0.761 to 0.620. The classification metrics, however, remained remarkably stable, with an average accuracy of 93.6% and an F1-score of 91.5%. This indicates that the GNN continues to correctly identify the qualitative delay category for most passenger journeys, even though predicting how much delay has accumulated becomes significantly harder as it propagates across multiple transfers.

5.5.5.2 Prediction Analysis

Figure 5.13 shows the relationship between predicted and observed passenger delays after the next journey segment. Most observations remain concentrated around the identity line, indicating that the GNN successfully captures the general delay propagation behaviour across multimodal passenger journeys. However, compared with both XGBoost and the LSTM, a larger dispersion is observed for medium and high delay values, leading to higher prediction errors as the size of the accumulated delay increases.

A similar pattern is observed for the final passenger delay (Figure 5.14). Although the GNN preserves the relationship between predicted and observed delays, the prediction dispersion becomes noticeably larger for severe disruption events, leading to systematic underestimation of the highest accumulated delays. These observations are consistent with the lower R^2 and higher MAE reported in Table 5.10. Overall, the scatter plots indicate that the graph representation successfully models the overall propagation process, but is less effective than the LSTM at estimating how much delay accumulates over long-term passenger journeys. This suggests that, under the proposed feature representation, explicitly modelling temporal dependencies provides greater predictive benefit than relying solely on graph connectivity.

A more detailed comparison of the error distributions across all models is presented in Section 4.4.6.

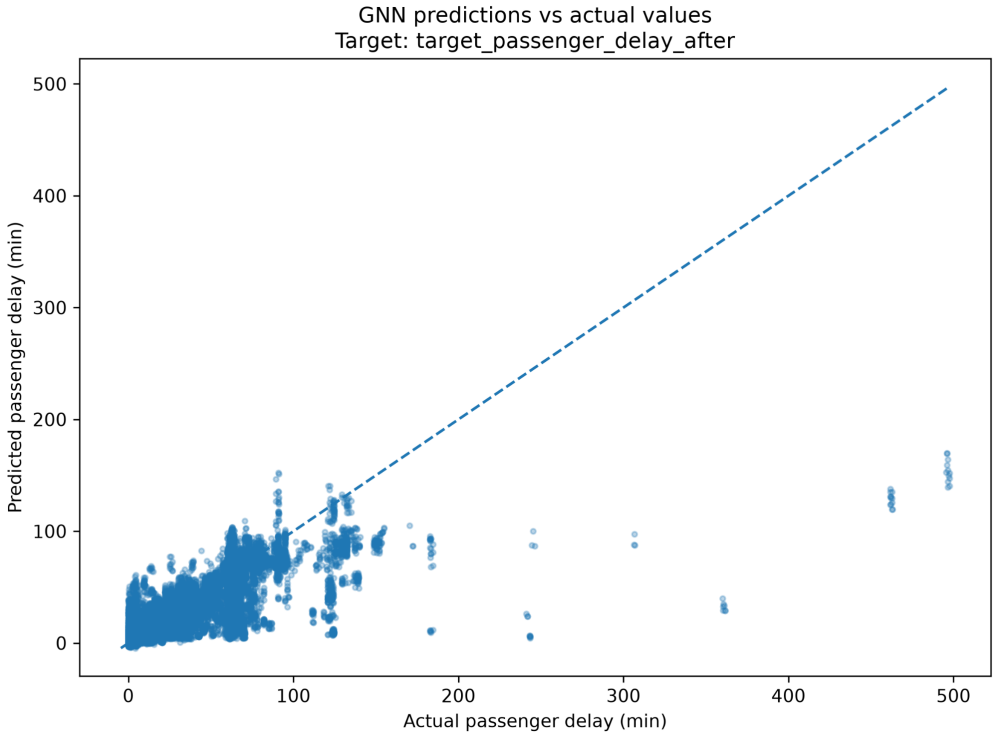


Figure 5.13: Predicted versus observed next-segment passenger delays obtained by the GNN model.

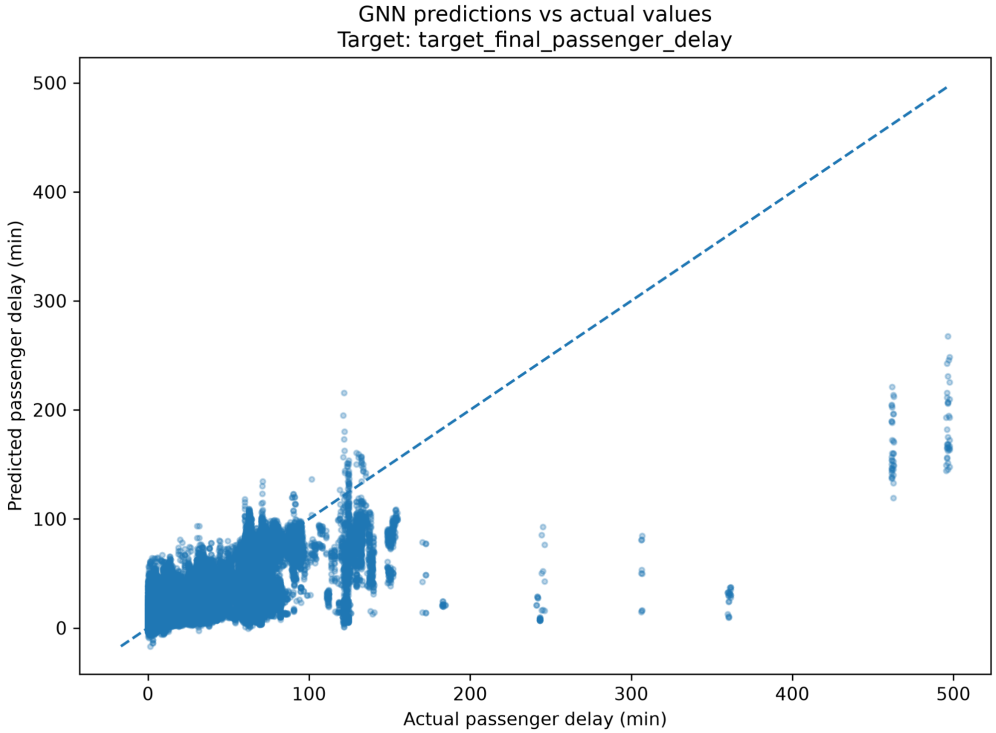


Figure 5.14: Predicted versus observed final passenger delays obtained by the GNN model.

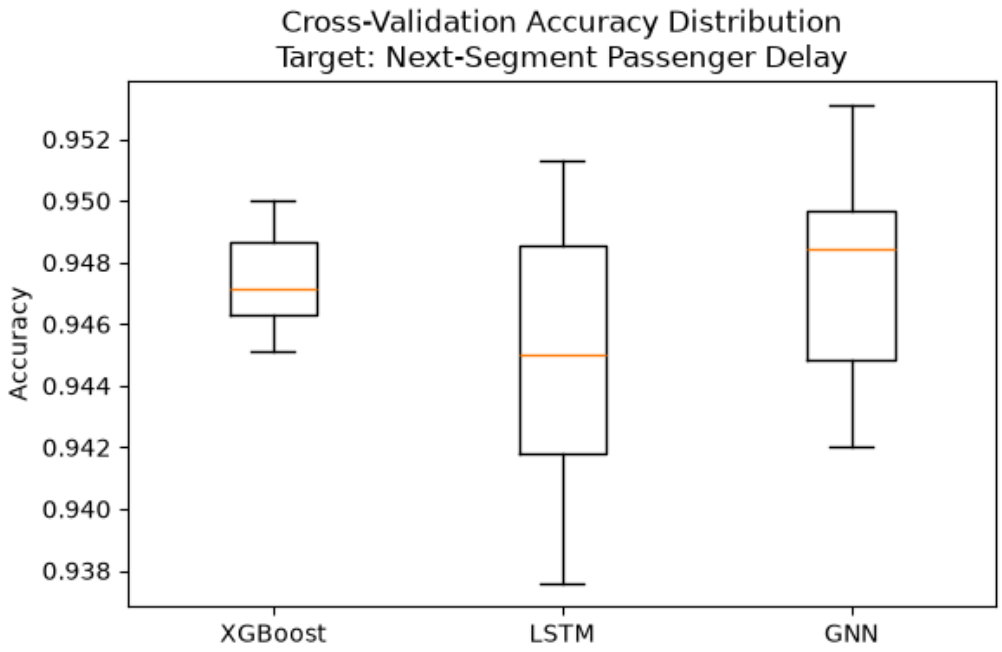


Figure 5.15: Cross-validation accuracy distribution for the next-segment passenger delay target across the XGBoost, LSTM, and GNN models.

5.5.6 Multimodal Model Performance Comparison

Figures 5.15 and 5.16 show the cross-validation accuracy distributions obtained by the three models for the two multimodal prediction tasks. While the complete quantitative results are given in Table 5.11, the boxplots offer further insight into the robustness and consistency of each model across the ten folds. The corresponding MAE cross-validation distributions are included in Appendix B.

For the next-segment delay task (Figure 5.15), all three models achieve remarkably similar classification performance, with median accuracies close to 95%. The compact interquartile ranges indicate stable behaviour across folds, suggesting that the multimodal feature representation provides sufficient information for reliable short-term delay prediction regardless of the chosen algorithm. Although the median accuracies are similar across models, the LSTM shows the largest variability between folds, indicating greater sensitivity to the training partition, while both XGBoost and the GNN show narrower distributions, with the GNN achieving the highest median accuracy among the three. This suggests that modelling the multimodal transport network as a graph offers a slight advantage when predicting delay propagation immediately after a transfer, where spatial relationships between network elements become particularly relevant.

A similar analysis applies to the final passenger delay task (Figure 5.16), where clearer differences emerge between approaches. The GNN again achieves the highest median accuracy, while keeping a relatively compact distribution despite a small outlier. The LSTM consistently outperforms XGBoost and shows moderate variability across folds, whereas XGBoost produces the lowest median accuracy of the three.

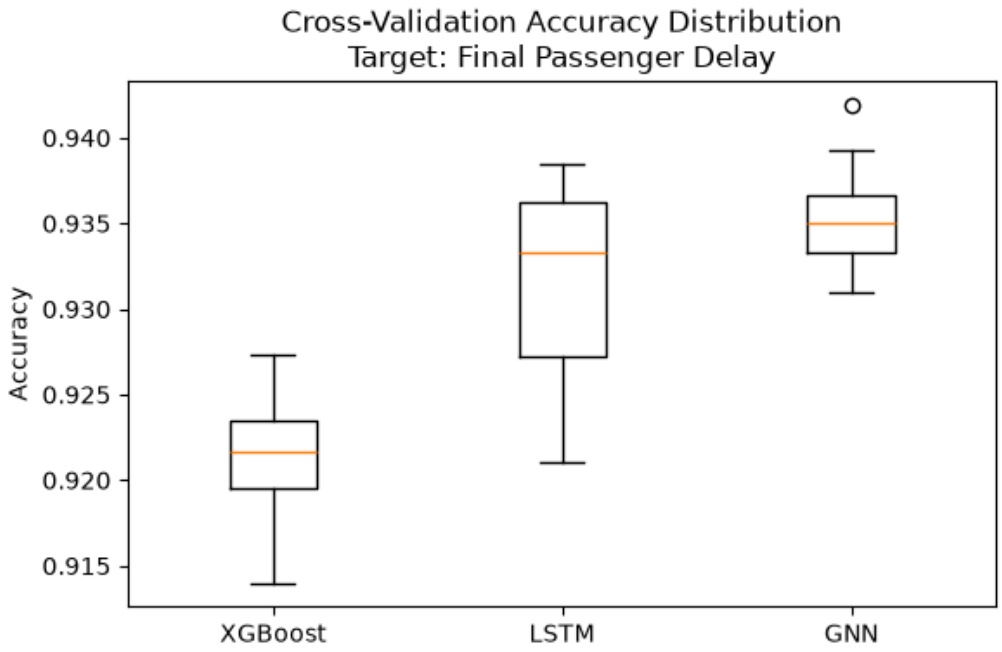


Figure 5.16: Cross-validation accuracy distribution for the final passenger delay target, showing consistently high classification performance across the multimodal models, with the GNN and LSTM achieving higher median accuracy than XGBoost.

Overall, the cross-validation distributions show that all three models provide stable predictive performance across folds. The observed differences are consistent with the average metrics in Table 5.11, with the LSTM achieving the lowest regression error and the GNN the highest classification scores. These patterns are examined in more detail below.

5.5.7 Comparative Error Analysis

To better understand how prediction errors vary across operating conditions, Figures 5.17 and 5.18 show the distribution of absolute prediction errors for each model across increasing passenger delay intervals.

Target	Model	MAE	R^2	Accuracy	Precision	Recall	F1-score
Next-segment delay	XGBoost	3.686	0.890	0.948	0.932	0.948	0.938
	LSTM	**2.856**	**0.930**	0.931	0.9328	0.931	0.931
	GNN	4.084	0.761	**0.948**	**0.935**	**0.948**	**0.939**
Final passenger delay	XGBoost	8.111	0.736	0.921	0.905	0.921	0.912
	LSTM	**6.129**	**0.815**	0.932	**0.905**	0.932	**0.915**
	GNN	8.566	0.620	**0.936**	0.902	**0.936**	**0.915**

Table 5.11: Average predictive performance of the evaluated multimodal models obtained using 10-fold cross-validation for both passenger delay prediction targets. Best results for each target are highlighted in bold.

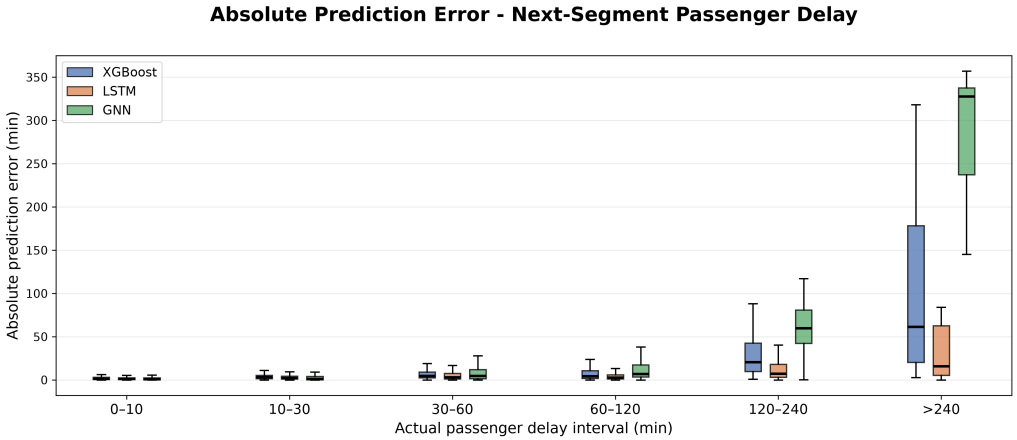


Figure 5.17: Distribution of absolute prediction errors for the next-segment passenger delay target across actual delay intervals, comparing XGBoost, LSTM, and GNN.

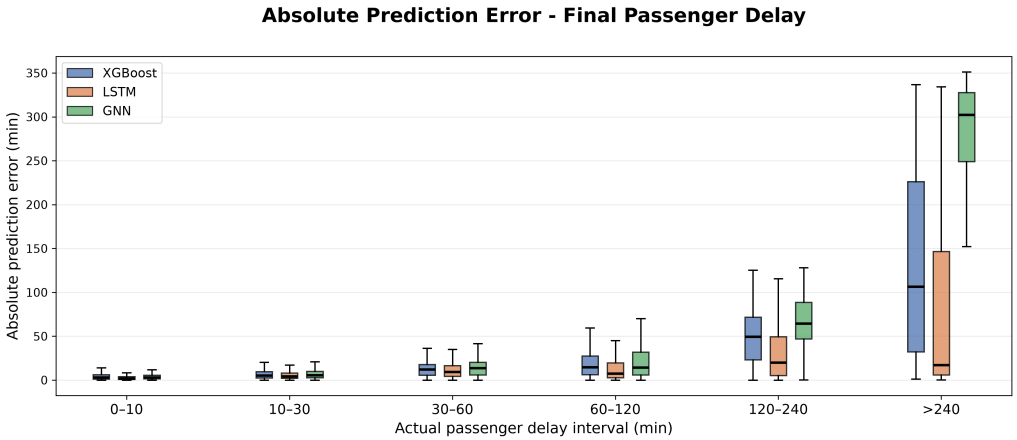


Figure 5.18: Absolute prediction error distribution for the final passenger delay target across actual delay intervals, showing increasing error dispersion for longer accumulated delays, with the LSTM maintaining lower errors than XGBoost and GNN in high-delay journeys.

For both prediction targets, all models achieve low absolute errors for delays below 30 minutes, with narrow interquartile ranges and a median error of only a few minutes. Error variability gradually increases for delays between 30 and 120 minutes, reflecting the greater uncertainty associated with larger delay propagation effects. A different behaviour appears for extreme delays beyond 240 minutes, where prediction errors increase considerably for all models, showing that severe disruptions remain the hardest cases to predict. Such events are relatively rare in the dataset and typically correspond to complex propagation scenarios involving missed transfers and accumulated delays over several segments. For the next-segment target (Figure 5.17), the LSTM shows the smallest median error across nearly all delay intervals, particularly for large delays, while XGBoost shows greater error dispersion as delay magnitude increases. The GNN shows the widest error distribution for extreme delays, suggesting that, although the graph representation captures network connectivity, it struggles to model very large delay propagation events accurately. A similar pattern appears for the final passenger delay (Figure 5.18): prediction errors remain relatively low for moderate delays but increase at the largest delay interval. The LSTM again has the lowest median error and the narrowest interquartile range, whereas XGBoost shows higher variability and the GNN produces the largest errors for the most severe disruption scenarios. Additional plots in Appendix B (Figures B.6 and B.5) provide a sample-level view of these errors, confirming that low and moderate delays are generally associated with small prediction errors, while the largest errors are concentrated in a limited number of rare, high-delay events.

Overall, these results show that prediction difficulty grows with the magnitude of passenger delay regardless of the modelling approach. While all three models provide accurate predictions under normal operating conditions, extreme disruption events remain challenging, since they result from complex interactions between accumulated vehicle delays, transfer penalties, and cascading delay propagation throughout multimodal journeys. The superior robustness of the LSTM at the highest delay intervals suggests that its sequential modelling capability gives a more effective representation of temporal delay accumulation than either the tree-based XGBoost model or the graph-based GNN.

5.6 Single-Mode vs Multimodal Prediction

Tables 5.12 and 5.13 summarise the overall regression and classification performance of the evaluated models across the single-mode and multimodal prediction scenarios.

A direct comparison between the single-mode and multimodal experiments shows that the multimodal task is substantially more complex. While the single-mode models operate on individual vehicle trips, the multimodal framework requires modelling the accumulation and propagation of passenger delays across multiple journey segments, where transfer waiting times, missed connections, and contextual events jointly create the final prediction. All models accordingly show higher MAE values in the multimodal setting, particularly for the final passenger delay, which reflects the cumulative effect of the entire journey. Despite this added complexity, classification performance remains consistently high across all models, with accuracy generally exceeding

Dataset	Target	Model	MAE	R^2
Single-mode	Vehicle delay	XGBoost	**0.414 ± 0.003**	–
	Vehicle delay	LSTM	1.231 ± 0.217	–
	Vehicle delay	GNN	0.775 ± 0.074	–
Multimodal	Next-segment delay	XGBoost	3.686 ± 0.035	0.890 ± 0.010
		LSTM	**2.855 ± 0.092**	**0.930 ± 0.005**
		GNN	4.084 ± 0.093	0.761 ± 0.021
	Final passenger delay	XGBoost	8.111 ± 0.159	0.736 ± 0.013
		LSTM	**6.129 ± 0.114**	**0.815 ± 0.011**
		GNN	8.566 ± 0.204	0.620 ± 0.018

Table 5.12: Regression performance comparison between the evaluated single-mode and multimodal prediction models using 10-fold cross-validation. Results are reported as mean ± standard deviation. Lower MAE and higher R^2 indicate better predictive performance.

92%, indicating that the proposed multimodal feature engineering strategy successfully preserves the overall delay categories even as numerical prediction becomes harder. The relative behaviour of the models also changes considerably between the two settings. In the single-mode scenario, XGBoost achieved the lowest prediction error, outperforming both deep learning approaches, suggesting that the propagation-aware features already capture most of the temporal information required to predict delays along individual vehicle trips, allowing the tree-based ensemble to model the underlying relationships very effectively.

In the multimodal experiments, by contrast, the LSTM architecture shows a clear advantage. For both prediction targets, it consistently achieved the lowest MAE and the highest R^2, indicating that sequential modelling becomes considerably more beneficial once passenger journeys involve multiple transfer events. The ability of recurrent networks to model temporal dependencies across consecutive transfers allows the LSTM to better capture the accumulation of delays throughout the journey, particularly when predicting the final passenger delay. The GNN produced the highest classification metrics in most multimodal experiments, confirming that graph representations effectively capture the structural relationships between consecutive transfer events. Its regression performance, however, remained below that of the LSTM, particularly for larger delay values, suggesting that the relatively simple graph architecture adopted here is less effective at modelling extreme delay propagation than recurrent sequence modelling.

Overall, these results indicate that the most appropriate model depends on the complexity of the prediction task. XGBoost provides an excellent solution for conventional single-mode delay prediction, due to its simplicity, robustness, and low prediction error. When passenger delay propagation across multimodal journeys is explicitly considered, however, the LSTM offers the best overall compromise between regression accuracy and predictive robustness. Its superior performance across both multimodal targets shows that sequential modelling is particularly well suited to representing the temporal evolution of passenger delays through successive transfers, making it the recommended approach for context-aware multimodal passenger delay prediction.

These findings also support the central hypothesis of this work: explicitly modelling passenger journeys, transfer dependencies, and contextual disruptions provides a richer representation of

Dataset	Target	Model	Accuracy	Precision	Recall	F1-score
Single-mode	Vehicle delay	XGBoost	0.964 ± 0.004	0.963 ± 0.005	0.964 ± 0.004	0.963 ± 0.004
	Vehicle delay	LSTM	0.935 ± 0.013	0.931 ± 0.014	0.935 ± 0.013	0.930 ± 0.014
	Vehicle delay	GNN	$\mathbf{0.992 \pm 0.006}$	$\mathbf{0.991 \pm 0.006}$	$\mathbf{0.992 \pm 0.006}$	$\mathbf{0.990 \pm 0.008}$
Multimodal	Next-segment delay	XGBoost	0.947 ± 0.002	0.932 ± 0.003	0.947 ± 0.002	0.938 ± 0.002
		LSTM	0.931 ± 0.007	0.933 ± 0.005	0.931 ± 0.007	0.931 ± 0.004
		GNN	$\mathbf{0.948 \pm 0.003}$	$\mathbf{0.935 \pm 0.006}$	$\mathbf{0.948 \pm 0.003}$	$\mathbf{0.939 \pm 0.004}$
	Final passenger delay	XGBoost	0.921 ± 0.004	0.905 ± 0.005	0.921 ± 0.004	0.912 ± 0.004
		LSTM	0.932 ± 0.006	$\mathbf{0.905 \pm 0.005}$	0.932 ± 0.006	$\mathbf{0.915 \pm 0.003}$
		GNN	$\mathbf{0.936 \pm 0.003}$	0.902 ± 0.003	$\mathbf{0.936 \pm 0.003}$	$\mathbf{0.915 \pm 0.003}$

Table 5.13: Classification performance comparison between the evaluated single-mode and multimodal prediction models using 10-fold cross-validation. Results are reported as mean $\pm$ standard deviation. Higher Accuracy, Precision, Recall and F1-score indicate better predictive performance.

delay propagation than conventional vehicle-based prediction. The transition from single-mode vehicle delay prediction to context-aware multimodal passenger delay prediction is therefore not only a more realistic formulation of the problem, but also one that allows sequence-based learning models to fully exploit the temporal dependencies inherent to multimodal transport systems.

Figures 5.20 and 5.19 provide a visual comparison of the prediction error and classification performance obtained by each model.

Among the evaluated approaches, the LSTM shows the smallest degradation in prediction error, achieving the lowest MAE for both multimodal targets (2.86 minutes for next-segment delay and 6.13 minutes for final passenger delay). XGBoost, which achieved the best performance in the single-mode experiments, by contrast shows an increase in prediction error, particularly for the final passenger delay task. The GNN consistently produces the highest MAE under the multimodal setting, suggesting that the adopted graph representation is not yet expressive enough to accurately capture long-range delay propagation across complex passenger journeys.

A different pattern appears for the classification metrics in Figure 5.19. Although accuracy decreases slightly for all models in the multimodal datasets, the reduction is considerably smaller than that observed for the regression metrics, indicating that, despite the increased difficulty of estimating how much delay has accumulated, the models remain able to correctly identify the corresponding delay category in most cases.

Overall, these results highlight a clear trade-off between the two prediction settings. The single-mode problem is considerably easier, allowing all models to achieve very low prediction errors. The multimodal formulation introduces substantially greater uncertainty, due to transfer interactions and accumulated delay propagation, leading to higher regression errors while largely preserving classification performance. This confirms that multimodal passenger delay prediction is a significantly more complex learning problem, and better reflects the operational complexity found in real public transport networks.

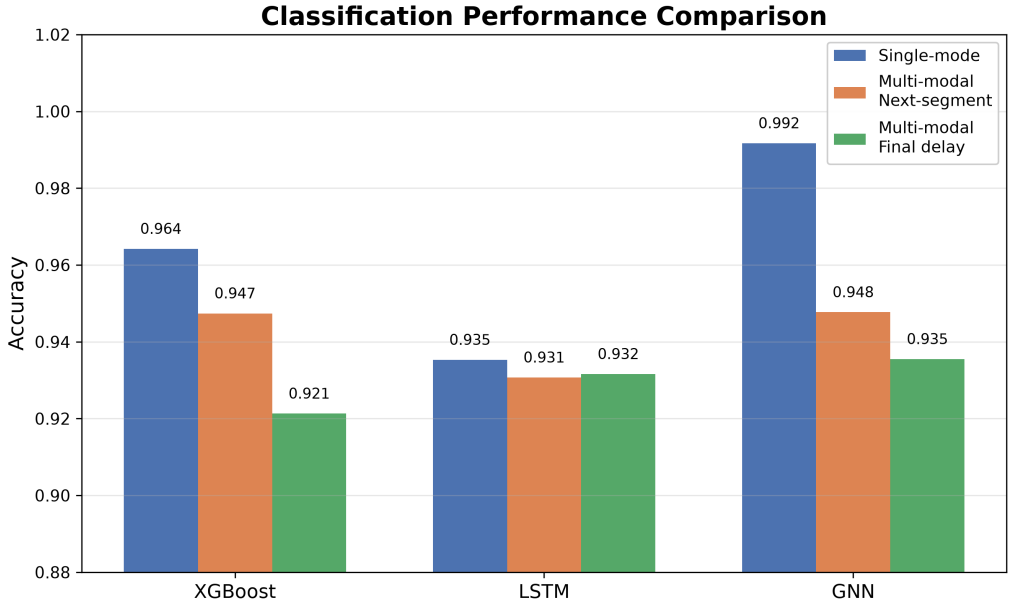


Figure 5.19: Classification accuracy comparison between the single-mode prediction task and the two multimodal passenger delay targets across XGBoost, LSTM, and GNN.

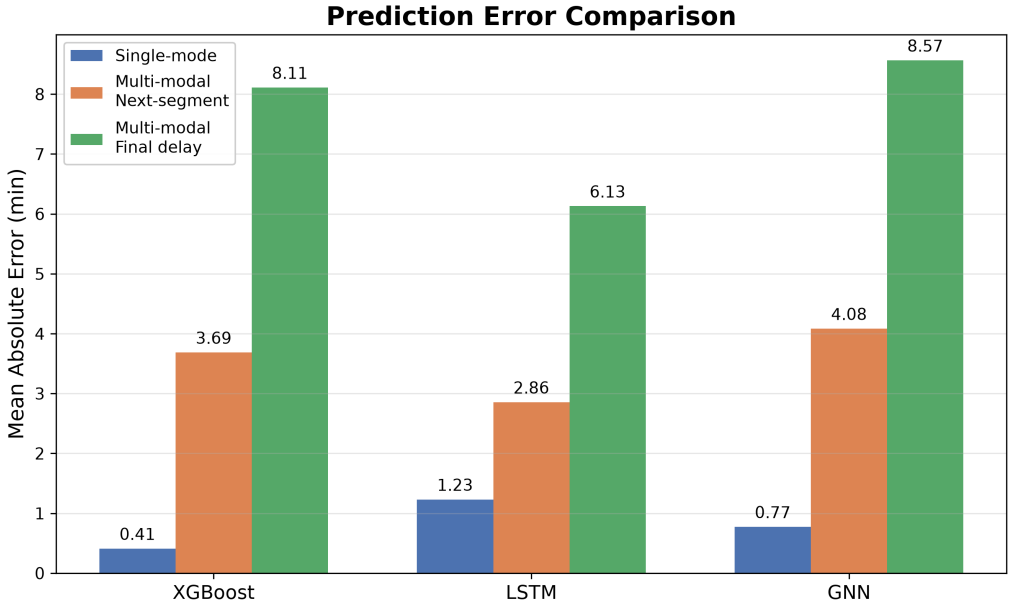


Figure 5.20: Mean absolute error comparison between the single-mode prediction task and the two multimodal passenger delay targets across XGBoost, LSTM, and GNN.

5.7 Discussion

The experimental evaluation shows that predictive performance depends not only on the chosen learning algorithm, but also on how well the model architecture matches the characteristics of the underlying prediction problem.

The feature engineering process proved to be a key component of the proposed framework. Historical propagation-aware variables consistently made the greatest predictive contribution, indicating that delay evolution is primarily driven by temporal dependencies between successive observations. Contextual information from DATEX II, including traffic events and road closures, acted instead as complementary information, improving prediction during disruption scenarios without replacing the predictive value of historical operational data. These findings highlight the importance of combining operational history with external contextual information to obtain robust delay prediction models.

The experimental comparison further shows that the suitability of each learning paradigm depends on the characteristics of the prediction task. When temporal propagation behaviour is already encoded through engineered features, tree-based ensemble methods efficiently capture the remaining non-linear relationships. As the prediction problem moves from individual vehicle delays to passenger delay propagation across multiple journey segments, however, explicitly modelling temporal dependencies becomes increasingly advantageous, explaining the superior performance of the LSTM in the multimodal setting. Although Graph Neural Networks naturally exploit the spatial structure of transport networks, passenger delay propagation depends on more than network topology alone: transfer timing, waiting behaviour, and the sequential accumulation of delays also play a fundamental role. Consequently, the GNN achieved lower regression performance than the LSTM, despite explicitly modelling the network's spatial structure.

Another important finding is that prediction uncertainty increases during severe disruption scenarios. While all evaluated models produced accurate predictions under normal operating conditions, large accumulated delays remained considerably harder to estimate. These rare events typically result from cascading propagation effects involving multiple missed connections and prolonged waiting times, making their evolution inherently less predictable than conventional operational delays.

Overall, the results show that passenger-oriented delay prediction requires both an appropriate multimodal journey representation and a learning architecture capable of exploiting its temporal characteristics. The proposed framework successfully combines propagation-aware feature engineering, contextual operational information, and sequential modelling, providing an effective approach for passenger-centric delay prediction in multimodal public transport networks.

5.8 Final Model Selection

The primary objective of this dissertation is passenger-oriented multimodal delay prediction, rather than conventional vehicle delay estimation. The final model selection is therefore based not only

Scenario	Selected Model	Selection rationale
Single-mode vehicle delay prediction	XGBoost	Highest regression accuracy, lowest MAE, robust cross-validation performance, and computational efficiency.
Multimodal passenger delay prediction	**LSTM**	Lowest MAE for both multimodal prediction targets and superior modelling of temporal delay accumulation across complete passenger journeys.

Table 5.14: Summary of the selected predictive models for each prediction scenario.

on predictive accuracy, but also on the ability of each learning architecture to represent the temporal dynamics of delay propagation throughout complete passenger journeys.

The overall comparison in Table 5.14 shows that different modelling approaches are preferable depending on the prediction scenario. For single-mode vehicle delay prediction, XGBoost achieved the highest regression performance while maintaining excellent robustness across folds. For multimodal passenger delay prediction, however, the LSTM consistently obtained the lowest prediction error for both evaluated targets, with average MAE values of 2.86 minutes for next-segment delay and 6.13 minutes for final passenger delay. Although the GNN explicitly models the transport network topology, its regression performance remained below that of the LSTM, indicating that, under the proposed passenger-chain representation, accurately modelling the temporal evolution of delays across successive segments is more beneficial than exploiting spatial connectivity alone. The LSTM is therefore selected as the final prediction model adopted in this dissertation. Combined with the proposed multimodal feature engineering framework, it offers the best balance between predictive accuracy, robustness, and the ability to model the temporal accumulation of delays across multimodal passenger journeys. The resulting framework provides a complete, context-aware solution for estimating passenger delays under realistic multimodal operating conditions.

5.9 Practical Prediction Examples

To complement the aggregate evaluation metrics, two practical prediction examples were selected to illustrate the output of the selected multimodal LSTM model. The examples correspond to representative passenger journeys involving bus, metro, and rail services. The first journey is completed without missed connections, while the second includes missed connections that introduce additional transfer penalties and increase the accumulated passenger delay.

Table 5.15 summarises the final passenger delay predictions for both journeys. The journey identifier corresponds to the internal *chain_id* used in the generated multimodal dataset. The columns *Actual final delay* and *Predicted final delay* correspond, respectively, to the observed passenger delay and the value estimated by the final LSTM model.

To give a more detailed view of the prediction process, Table 5.16 reports the delay evolution after each transfer for the same two journeys. This transfer-level representation shows how

Journey ID	Origin stop	Destination stop	Modes	Transfers	Missed conn.	Actual final delay	Predicted final delay
31492	3890023	3153078	Bus–Metro–Rail	3	0	3.34 min	3.09 min
22469	3890023	3917277	Bus–Metro–Rail	3	2	42.25 min	42.57 min

Table 5.15: Practical examples of final passenger delay predictions obtained with the selected multimodal LSTM model.

Journey ID	Transfer	Mode change	Missed conn.	Transfer penalty	Actual delay after transfer	Predicted delay after transfer
31492	1	Bus–Bus	0	0.00 min	2.22 min	1.47 min
31492	2	Bus–Rail	0	0.00 min	0.48 min	1.04 min
31492	3	Rail–Metro	0	0.00 min	3.34 min	2.83 min
22469	1	Bus–Bus	0	0.00 min	9.08 min	6.83 min
22469	2	Bus–Rail	1	30.00 min	34.63 min	24.05 min
22469	3	Rail–Metro	1	40.00 min	42.25 min	42.84 min

Table 5.16: Transfer-level passenger delay predictions for a successful multimodal journey without missed connections (Journey ID = 31492) and transfer-level passenger delay predictions for a successful multimodal journey with missed connections (Journey ID = 22469).

passenger delay accumulates along the journey and how missed connections can amplify the final delay experienced by passengers.

In the successful journey, all transfers are completed without additional transfer penalties, and the final passenger delay remains low. In the disrupted journey, by contrast, two missed connections introduce transfer penalties of 30 and 40 minutes, increasing the passenger-level delay from 9.08 minutes after the first transfer to 42.25 minutes by the end of the journey. The selected LSTM model estimates a final delay of 42.57 minutes for this case, illustrating the practical relevance of explicitly modelling transfer feasibility and missed connections in multimodal passenger delay prediction.

Chapter 6

Conclusions

6.1 Summary of Work

This dissertation presented a context-aware framework for passenger-oriented delay prediction in multimodal public transport systems, integrating operational schedule data from the General Transit Feed Specification (GTFS) with contextual disruption information extracted from DATEX II. Unlike conventional approaches that focus primarily on predicting delays for individual vehicles or single transport modes, the proposed framework explicitly models complete passenger journeys, capturing delay propagation across multiple transport segments, transfer events, missed connections, accumulated delays, and the influence of external contextual disruptions. This passenger-centred perspective provides a more realistic representation of multimodal travel, in which a delay experienced in one segment may propagate throughout the remainder of the journey.

To support this objective, a complete data processing and modelling pipeline was developed. The methodology was structured incrementally, beginning with a single-mode, stop-level formulation used to validate the data integration pipeline, the feature engineering strategy, and the predictive modelling approach. This baseline was then extended to a multimodal, transfer-based formulation, in which passenger journeys are represented as chains of connected transport segments linked by transfer events. Since synchronised real-time delay observations were not available for the chosen case study, a stochastic delay simulation mechanism was developed to generate realistic vehicle delay patterns, which were subsequently propagated through passenger journeys by evaluating transfer feasibility, missed connections, and the resulting transfer penalties.

The framework integrates three primary data sources: the Netherlands OpenOV GTFS feed, providing the operational schedule for a national, multimodal network comprising bus, tram, metro, rail, and ferry services; and two DATEX II datasets, describing road closures and traffic events, respectively, obtained from the Dutch National Data Warehouse for Traffic Information (NDW). An extensive feature engineering process was conducted for both the single-mode and multimodal datasets, capturing temporal, operational, topological, propagation-related, and contextual characteristics. Correlation analysis, feature importance evaluation, and systematic contextual feature screening were performed to optimise the feature space, replacing redundant

instantaneous delay observations with weighted temporal history representations and removing variables that captured the same underlying information, while preserving features that described complementary aspects of delay propagation.

Three complementary machine learning paradigms were comparatively evaluated across both the single-mode and multimodal prediction tasks: Extreme Gradient Boosting (XGBoost), Long Short-Term Memory (LSTM) networks, and Graph Neural Networks (GNNs). Models were assessed using ten-fold, chain-aware cross-validation, combining regression metrics (MAE, R^2) with classification metrics (Accuracy, Precision, Recall, F1-score) derived from discretised delay categories, and reported with 95% confidence intervals to ensure statistical robustness.

6.2 Addressing the Research Questions

RQ1 - Data integration. The first research question asked how heterogeneous open data sources could be integrated into a unified, passenger-oriented representation suitable for modelling multimodal delay propagation. This was addressed through a spatial integration strategy combining GTFS operational data with DATEX II contextual data via Haversine-distance matching within a 2 km radius, applied first at stop level and subsequently at segment level for the multimodal formulation. The resulting unified dataset, comprising 50,000 transfer-level observations with no missing values or duplicates, demonstrates that open, standardised data sources can be successfully combined into a consistent, passenger-oriented representation, despite originating from entirely independent providers and formats.

RQ2 - Explicit transfer and propagation modelling. The second research question examined how explicitly modelling passenger transfers, missed connections, and cumulative delay propagation, rather than predicting delays at the level of individual vehicles, affects the accuracy and realism of delay prediction. The comparative results in Section 5.6 confirm that the multimodal, passenger-centred formulation is a substantially more complex prediction problem than conventional single-mode delay prediction, with higher MAE values across all models, particularly for the final passenger delay target. However, the feature importance analysis (Section 5.5.3.3) showed that transfer-specific variables, namely *transfer_penalty_departure* and *initial_vehicle_delay*, dominate the prediction of final passenger delay, confirming that explicitly modelling transfer feasibility and missed connections captures the dominant mechanism through which small vehicle delays escalate into substantially larger passenger delays. This supports the central premise of the dissertation: passenger-level delay prediction requires explicit transfer modelling, and cannot be adequately approximated by aggregating independent, vehicle-level predictions.

RQ3 - Comparison of predictive paradigms. The third research question asked which of the three complementary paradigms, tabular, sequential, and graph-based, was best suited to capturing delay propagation at the single-mode and multimodal levels. The results showed that no single paradigm was universally superior. Instead, model suitability depended on the structure of the prediction task. For single-mode, vehicle-level delay prediction, XGBoost achieved the lowest

prediction error (MAE = 0.414 minutes) and the most stable performance across folds, indicating that, once propagation-aware features such as weighted delay history are available, a tree-based ensemble can exploit this engineered temporal information as effectively as, or more effectively than, models with explicit sequential or spatial inductive biases. For multimodal, passenger-level delay prediction, by contrast, the LSTM consistently achieved the lowest MAE for both prediction targets (2.86 minutes for next-segment delay and 6.13 minutes for final passenger delay), confirming that explicit sequential modelling becomes increasingly beneficial as the prediction horizon extends across multiple transfer events. The GNN achieved the highest classification metrics in the multimodal setting, reflecting its ability to exploit the connectivity between consecutive transfers, but its regression performance remained below that of the LSTM, suggesting that the static graph representation adopted in this work was not sufficiently expressive to outperform sequential modelling for this task.

RQ4 - Contribution of contextual information. The fourth research question investigated the extent to which contextual traffic disruption information improves multimodal delay propagation prediction, relative to operational and historical delay features alone. The contextual feature screening experiments (Section 5.3) showed that removing road closure and traffic-event variables produced negligible changes in predictive performance, and the corresponding feature importance analyses consistently ranked contextual variables below propagation-related variables. This was explained by the sparse distribution of contextual events across the dataset, with 92.1% of transport segments unaffected by any nearby disruption (Section 5.2.1). Despite their limited statistical contribution, contextual variables were retained in the final framework, since they represent real external operating conditions and were found to become more informative when aggregated at the segment level, rather than the stop level, as reflected in their stronger correlations with the prediction target in the multimodal dataset.

6.3 Validation of Hypotheses

Following the expected outcomes formulated in Section 3.12 of Chapter 3, three working hypotheses underpinned the design of this dissertation. Each is assessed below in light of the experimental results.

H1 - Explicit transfer modelling improves the realism of passenger delay representation. It was hypothesised that explicitly modelling passenger transfers, missed connections, and accumulated delay propagation would provide a more realistic and informative representation of multimodal journey reliability than conventional, vehicle-centred delay prediction. This hypothesis is supported by the results. The dominance of transfer-related features in the multimodal feature importance analysis, together with the practical prediction examples in Section 5.9, illustrating how two missed connections increased a passenger's accumulated delay from 9.08 to 42.25 minutes, demonstrates that vehicle-level delay alone would substantially underestimate the disruption actually experienced by the passenger.

H2 - The most suitable learning paradigm depends on the prediction scope. It was hypothesised that the comparative evaluation of XGBoost, LSTM, and GNN models would clarify which learning paradigm is best suited to each stage of the problem, namely local, single-trip delay prediction and longer-scope, passenger-centred delay propagation. This hypothesis is confirmed: XGBoost was the most effective model for single-mode prediction, while the LSTM was the most effective for multimodal, passenger-level prediction. This finding indicates that paradigm selection should not be treated as a fixed methodological choice, but should instead be guided by the temporal and structural characteristics of the specific prediction task.

H3 - Graph-based modelling captures complementary structural information. It was hypothesised that incorporating network topology, through Graph Neural Networks, would improve predictive performance by exploiting the spatial structure of the transport network. This hypothesis was only partially supported. While the GNN achieved competitive, and at times superior, classification performance, particularly for the multimodal final passenger delay target, its regression accuracy remained below that of the LSTM in both the single-mode and multimodal settings. This suggests that, under the static graph representation adopted in this work, spatial topology alone is insufficient to outperform explicit temporal modelling, although it does provide complementary value, particularly for correctly identifying qualitative delay categories.

6.4 Overall Contributions

This dissertation makes the following contributions to the study of delay prediction in multimodal public transport systems:

1. A passenger-centred problem formulation that shifts the unit of prediction from individual vehicle stops to transfer events within complete multimodal journeys, explicitly representing transfer feasibility, missed connections, and accumulated passenger delay.

2. An integrated data pipeline combining GTFS operational schedules with DATEX II contextual disruption data through spatial matching, applied consistently at both stop level and journey-segment level.

3. A realistic delay simulation and propagation mechanism, modelling vehicle delays as a cumulative stochastic process and propagating these delays through passenger journeys via explicit transfer feasibility evaluation, in the absence of complete real-time delay observations.

4. A systematic feature engineering methodology, including the development of weighted temporal history representations that reduce redundancy between consecutive delay observations while preserving predictive information, applied consistently across both the single-mode and multimodal datasets.

5. A comparative evaluation of three complementary machine learning paradigms, tree-based ensembles, recurrent neural networks, and graph neural networks, across two prediction

scopes (single-mode and multimodal) and two prediction targets (next-segment and final passenger delay), providing empirical evidence on which paradigm is best suited to each scenario.

6. Empirical evidence on the limited, but non-negligible, predictive contribution of contextual traffic disruption data under conditions of data sparsity, together with a demonstration that segment-level contextual aggregation is more informative than stop-level aggregation for multimodal prediction.

7. **Research dissemination**. An article entitled "A Data-Driven Approach to Predicting Itinerary Delays in Multi-Modal Transit Networks", based on the work developed in this dissertation, was accepted for publication at the IEEE International Smart Cities Conference (ISC2 2026).

6.5 Limitations and Future Work

Although the proposed framework achieved encouraging predictive performance, several limitations should be acknowledged, each of which suggests a corresponding direction for future research.

Reliance on simulated delay data. The experimental evaluation was conducted using simulated delay propagation rather than real operational delay records, primarily due to the limited availability of complete passenger trajectory data containing both multimodal journeys and transfer outcomes. The proposed simulation incorporates realistic mechanisms, including accumulated vehicle delays, missed connections, transfer penalties, and contextual disruptions, providing a controlled environment for modelling passenger-oriented delay propagation. Nevertheless, simulated delays cannot fully reproduce the complexity and variability of real transport operations, which are influenced by factors such as passenger demand, operational incidents, and network-wide interactions. The integration of real-time operational data from the Dutch transport network was investigated during this work through the available OpenOV Application Programming Interface (API). However, due to time constraints and the complexity of integrating these heterogeneous sources with the proposed framework, this extension could not be completed within the scope of the dissertation. Adapting the proposed pipeline to real operational data would require three main methodological changes. First, the delay simulation stage (Section 3.7) would be replaced by real-time Automatic Vehicle Location (AVL) feeds, with the trend and noise decomposition no longer needed, since actual delay values would be observed directly at each stop. Second, the BFS-based chain construction and transfer feasibility logic would need to operate incrementally, evaluating transfer feasibility as vehicles report their positions rather than reconstructing complete passenger chains retrospectively. Third, the contextual integration stage, developed in Section 3.9, would need to consume live DATEX II event streams instead of static snapshots, with a rolling aggregation window replacing the fixed 2 km spatial threshold used in this work. In a

deployment setting, a transport authority could run the trained LSTM model as a streaming service, updating passenger delay predictions incrementally as new vehicle position and disruption reports arrive, and triggering transfer-risk alerts when predicted delays exceed the minimum transfer time at upcoming connections. The main engineering challenge in this scenario would not be model inference, which is computationally lightweight, but the synchronisation of heterogeneous, asynchronously updated data streams (AVL, GTFS-realtime, and DATEX II) within a consistent temporal reference frame. Future work should integrate real AVL data together with passenger journey information to validate the proposed methodology under real operating conditions, capturing operational phenomena that are difficult to reproduce through simulation, such as irregular service patterns and unforeseen incidents.

Sparsity of contextual disruption data. The contextual information extracted from DATEX II is relatively sparse compared with the predominance of normal operating conditions throughout the dataset, with only a small proportion of observations associated with nearby road closures or traffic-related events. Consequently, although contextual variables improve the operational realism of the framework, their predictive contribution remains limited compared with propagation-related features. This raises a legitimate question regarding the cost-effectiveness of contextual integration: given that DATEX II data accounted for only a marginal share of predictive performance, while requiring substantial engineering effort for spatial aggregation, coordinate matching, and stop-to-chain propagation (Sections 3.9 and 3.11), it could be argued that this integration is not justified on purely predictive grounds. However, three considerations support its retention. First, the 92.1% sparsity of contextual events (Section 5.2.1) means that the current evaluation window under-represents disruption scenarios. The practical value of contextual variables is concentrated precisely in the 7.9% of segments where they are informative, and this value cannot be observed in aggregate performance metrics that are dominated by normal-condition observations. Second, the screening analysis (Section 5.3) evaluated contextual removal only under the current sparse-event regime; under denser or fully real-world disruption reporting, the relative contribution of these variables would be expected to increase. Third, retaining contextual variables preserves the operational realism and extensibility of the framework, allowing it to be deployed in networks or periods with higher disruption frequency without architectural changes. The trade-off is therefore not between predictive gain and engineering cost in the present dataset, but between short-term marginal accuracy and long-term generalisability of the framework to more disruption-prone operating conditions. Future work could enrich the contextual representation by incorporating additional external data sources, such as weather conditions, passenger demand, and real-time operational information, to provide a more comprehensive description of the transport environment, particularly during abnormal operating conditions.

Static graph representation. The Graph Neural Network adopted in this work is based on a static Graph Convolutional Network (GCN) representation, in which transport stops or transfer events are represented as graph nodes connected through fixed edges. Although this representation successfully captures the spatial or sequential topology of the transport network, it is less suited to modelling the temporal evolution of passenger delays across complete multimodal journeys,

since delay propagation is inherently a spatio-temporal process. Future work should investigate Temporal Graph Neural Networks, Dynamic Graph Neural Networks, or heterogeneous graph representations incorporating different transport modes and transfer relationships, which may provide a richer representation of multimodal delay propagation than the static graph adopted in this work.

Fixed-route passenger behaviour. Passenger behaviour was simplified by assuming that travellers continue to follow their originally planned itinerary after a missed connection, boarding the next available service on the same route according to the GTFS schedule. In practice, passengers frequently adapt their journeys by selecting alternative routes, transfer locations, or transport modes to minimise their final arrival delay. This assumption is also one of the main sources of the extreme prediction errors identified in Section 5.5.3.4, where a small number of passenger chains experienced unusually long waiting times due to low service frequencies on the originally planned route. Future work should integrate dynamic route replanning algorithms, capable of evaluating multiple feasible itineraries after a disruption event and selecting the alternative with the lowest predicted arrival delay, providing a more realistic representation of passenger adaptation strategies and potentially reducing the magnitude of these extreme prediction errors.

Prediction uncertainty under severe disruption. Across all three models, prediction errors increased considerably for rare, severe disruption scenarios involving multiple missed connections and prolonged waiting times. Rather than further optimising prediction under normal operating conditions, where all models already achieve low error, future work should prioritise improving the representation of these rare but operationally significant events, for instance through dedicated modelling of extreme delay distributions or targeted data augmentation strategies for underrepresented disruption scenarios.

Taken together, these limitations do not undermine the central contribution of this dissertation, namely that explicit, passenger-centred modelling of transfer feasibility and delay propagation provides a more realistic and informative basis for multimodal delay prediction than conventional vehicle-centred approaches. Rather, they define a clear and well-motivated research agenda for extending the proposed framework towards real-world deployment, encompassing real operational data, richer contextual integration, temporally-aware graph learning, and adaptive passenger decision-making, ultimately contributing to more resilient and passenger-centred intelligent transport systems.

References

[1] Volodymyr Agafonkin. *Leaflet*.

[2] V.A. Boateng and B. Yang. "A Global Modeling Pruning Ensemble Stacking With Deep Learning and Neural Network Meta-Learner for Passenger Train Delay Prediction". In: *IEEE Access* 11 (2023), pp. 62605–62615. DOI: 10.1109/ACCESS.2023.3287975.

[3] W. Chen et al. "A Deep Learning Model with Conv-LSTM Networks for Subway Passenger Congestion Delay Prediction". In: *Journal of Advanced Transportation* 2021 (2021). DOI: 10.1155/2021/6645214.

[4] X. Chen, Z. Ma, and W. Sun. "Incident Delay Prediction in Urban Railway Systems: Methodology Review and Exploratory Comparative Analysis". In: *Transportation Research Record* 2678.12 (2024), pp. 1629–1638. DOI: 10.1177/03611981241252831.

[5] S. Chtioui et al. "XGBoost in Public Transportation for Multi-Attribute Data: Delay Prediction in Railway Systems in Real-Time". In: *IEEE Access* 12 (2024), pp. 143327–143342. DOI: 10.1109/ACCESS.2024.3463022.

[6] D. Zhang et al. "A Multi-Source Dynamic Temporal Point Process Model for Train Delay Prediction". In: *IEEE Transactions on Intelligent Transportation Systems* 25.11 (Nov. 2024), pp. 17865–17877. ISSN: 1558-0016. DOI: 10.1109/TITS.2024.3430031.

[7] Samenwerkingsverband DOVA. *Occupancy of Public Transportation Vehicles*. July 2025.

[8] T. Gao, J. Chen, and H. Xu. "Data-driven train delay prediction incorporating dispatching commands: An XGBoost-metaheuristic framework". In: *IET Intelligent Transport Systems* 18.10 (2024), pp. 1777–1796. DOI: 10.1049/itr2.12461.

[9] H. Rodriguez-Deniz and M. Villani. "Robust Real-Time Delay Predictions in a Network of High-Frequency Urban Buses". In: *IEEE Transactions on Intelligent Transportation Systems* 23.9 (Sept. 2022), pp. 16304–16317. ISSN: 1558-0016. DOI: 10.1109/TITS.2022.3149656.

[10] H. Zhang et al. "A Prediction Model for Bus Arrival Time at Bus Stop Considering Signal Control and Surrounding Traffic Flow". In: *IEEE Access* 8 (2020), pp. 127672–127681. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.3004856.

[11] J. Pineda-Jaramillo et al. "Short-Term Arrival Delay Time Prediction in Freight Rail Operations Using Data-Driven Models". In: *IEEE Access* 11 (2023), pp. 46966–46978. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2023.3275022.

[12] J. Jalaney and G.R. Ganesh. "IoT based bus arrival time prediction using Artificial Neural Network (ANN) for smart public transport system (SPTS)". In: *International Journal of Intelligent Engineering and Systems* 13.1 (2020), pp. 312–323. DOI: 10.22266/ijies2020.0229.29.

[13] S. Jeong, C. Oh, and J. Jeong. "BAT-Transformer: Prediction of Bus Arrival Time with Transformer Encoder for Smart Public Transportation System". In: *Applied Sciences (Switzerland)* 14.20 (2024). DOI: 10.3390/app14209488.

[14] K. -F. Chu et al. "Deep Encoder Cross Network for Estimated Time of Arrival". In: *IEEE Access* 11 (2023), pp. 76095–76107. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2023.3294345.

[15] O. Kaya and M. Utku Kalay. "Spatio-Temporal Forecasting of Bus Arrival Times Using Context-Aware Deep Learning Models in Urban Transit Systems". In: *IEEE Access* 13 (2025), pp. 161423–161435. DOI: 10.1109/ACCESS.2025.3609530.

[16] Y. Lai et al. "Bus Arrival Time Prediction Using Wavelet Neural Network Trained by Improved Particle Swarm Optimization". In: *Journal of Advanced Transportation* 2020 (2020). DOI: 10.1155/2020/7672847.

[17] P. Lapamonpinyo, S. Derrible, and F. Corman. "Real-Time Passenger Train Delay Prediction Using Machine Learning: A Case Study With Amtrak Passenger Train Routes". In: *IEEE Open Journal of Intelligent Transportation Systems* 3 (2022), pp. 539–550. DOI: 10.1109/OJITS.2022.3194879.

[18] C. Lee and Y. Yoon. "A Novel Bus Arrival Time Prediction Method Based on Spatio-Temporal Flow Centrality Analysis and Deep Learning". In: *Electronics (Switzerland)* 11.12 (2022). DOI: 10.3390/electronics11121875.

[19] Philip Lemaitre, Michael Riis Andersen, and Jes Frellsen. "When Did the Train Arrive? A Bayesian Approach to Enrich Timetable Information Using Smart Card Data". In: *IEEE Open Journal of Intelligent Transportation Systems* 2 (2021), pp. 160–172. ISSN: 2687-7813. DOI: 10.1109/OJITS.2021.3094620.

[20] H. Liu et al. "Bus Arrival Time Prediction Based on LSTM and Spatial-Temporal Feature Vector". In: *IEEE Access* 8 (2020), pp. 11917–11929. DOI: 10.1109/ACCESS.2020.2965094.

[21] Pedro P Lopes et al. "Towards Forecasting Bus Arrival Thorough A Model Based On GNN+LSTM Using GTFS and Real-time Data". In: *Proceedings of the 4th International Conference on AI-ML Systems*. AIMLSystems '24. New York, NY, USA: Association for Computing Machinery, 2025. ISBN: 979-8-4007-1161-9. DOI: 10.1145/3703412.3703417.

[22] M. A. Ghamdi, G. Parr, and W. Wang. "Heterogeneous Machine Learning Ensembles for Predicting Train Delays". In: *IEEE Transactions on Intelligent Transportation Systems* 25.6 (June 2024), pp. 5138–5153. ISSN: 1558-0016. DOI: 10.1109/TITS.2023.3337858.

[23] Ratna Mandal et al. "Exploiting Multi-modal Contextual Sensing for City-bus's Stay Location Characterization: Towards Sub-60 Seconds Accurate Arrival Time Prediction". In: *ACM Trans. Internet Things* 4.1 (Feb. 2023). DOI: 10.1145/3549548.

[24] G. Mustafa, Y. Hwang, and S.-J. Cho. "Predicting Bus Travel Time in Cheonan City through Deep Learning Utilizing Digital Tachograph Data". In: *Electronics (Switzerland)* 13.9 (2024). DOI: 10.3390/electronics13091771.

[25] NDW. *DATEXII Data Formats*. Apr. 2026.

[26] NDW. *XML structure - closures*. Oct. 2024.

[27] D. Panovski and T. Zaharia. "Long and Short-Term Bus Arrival Time Prediction with Traffic Density Matrix". In: *IEEE Access* 8 (2020), pp. 226267–226284. DOI: 10.1109/ACCESS.2020.3044173.

[28] A. Parslov, N.C. Petersen, and F. Rodrigues. "Short-term bus travel time prediction for transfer synchronization with intelligent uncertainty handling". In: *Expert Systems with Applications* 232 (2023). DOI: 10.1016/j.eswa.2023.120751.

[29] Q. Han et al. "A Bus Arrival Time Prediction Method Based on Position Calibration and LSTM". In: *IEEE Access* 8 (2020), pp. 42372–42383. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2976574.

[30] Michael Redmond, Ann Melissa Campbell, and Jan Fabian Ehmke. "Data-driven planning of reliable itineraries in multi-modal transit networks". In: *Springer* (Dec. 2019). DOI: https://doi.org/10.1007/s12469-019-00221-0. (Visited on 10/10/2025).

[31] N. SHANTHI et al. "Analysis on the Bus Arrival Time Prediction Model for Human-Centric Services Using Data Mining Techniques". In: *Computational Intelligence and Neuroscience* 2022 (2022). DOI: 10.1155/2022/7094654.

[32] S. Shanthi et al. "Optimizing City Transit: IoT and Gradient Boosting Algorithms for Accurate Bus Arrival Predictions". In: *IEEE* (Mar. 2024). DOI: 10.1109/IITCEE59897.2024.10467430. (Visited on 10/10/2025).

[33] C. Shi et al. "Enhancing Travel Time Prediction for Intelligent Transportation Systems: A High-Resolution Origin–Destination-Based Approach with Multi-Dimensional Features". In: *Sustainability (Switzerland)* 17.5 (2025). DOI: 10.3390/su17052111.

[34] Nishtha Srivastava, Bhavesh N. Gohil, and Suprio Ray. "Rail transit delay forecasting with Causal Machine Learning". In: *Proceedings of the 1st ACM SIGSPATIAL International Workshop on Spatiotemporal Causal Analysis*. STCausal '24. event-place: Atlanta, GA, USA. New York, NY, USA: Association for Computing Machinery, 2024, pp. 1–10. ISBN: 979-8-4007-1154-1. DOI: 10.1145/3681778.3698784.

[35] T. Qiu et al. "FEN-MRMGCN: A Frontend-Enhanced Network Based on Multi-Relational Modeling GCN for Bus Arrival Time Prediction". In: *IEEE Access* 13 (2025), pp. 5296–5307. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2024.3525357.

[36] D. Tatsui et al. "Prediction method of train delay using deep learning technique". In: *Quarterly Report of RTRI (Railway Technical Research Institute) (Japan)* 62.4 (2021), pp. 263–268. DOI: 10.2219/rtriqr.62.4_263.

[37] H. Wang. "A Two-Stage Train Delay Prediction Method Based on Data Smoothing and Multimodel Fusion Using Asymmetry Features in Urban Rail Systems". In: *Wireless Communications and Mobile Computing* 2022 (2022). DOI: 10.1155/2022/5188105.

[38] NDW (Nationaal Dataportaal Wegverkeer). *Current image*. Logistics, Traffic Safety, Traffic Management. Apr. 2026. (Visited on 04/24/2026).

[39] NDW (Nationaal Dataportaal Wegverkeer). *Temporary traffic measures - Closures*. Logistics, Traffic Safety, Traffic Management. Apr. 2026. (Visited on 04/23/2026).

[40] J. Wu et al. "A Hybrid LSTM-CPS Approach for Long-Term Prediction of Train Delays in Multivariate Time Series". In: *Future Transportation* 1.3 (2021), pp. 765–776. DOI: 10.3390/futuretransp1030042.

[41] R. Xu, F. Wang, and F. Zhou. "Metro Train Operation Plan Analysis Based on Station Travel Time Reliability". In: *Journal of Advanced Transportation* 2021 (2021). DOI: 10.1155/2021/8813461.

[42] B. Zhang et al. "Bus Arrival Time Prediction Based on the Optimized Long Short-Term Memory Neural Network Model With the Improved Whale Algorithm". In: *Journal of Advanced Transportation* 2024.1 (2024). DOI: 10.1155/2024/6997338.

[43] D. Zhang et al. “Train Time Delay Prediction for High-Speed Train Dispatching Based on Spatio-Temporal Graph Convolutional Network”. In: *IEEE Transactions on Intelligent Transportation Systems* 23.3 (2022), pp. 2434–2444. DOI: 10.1109/TITS.2021.3097064.

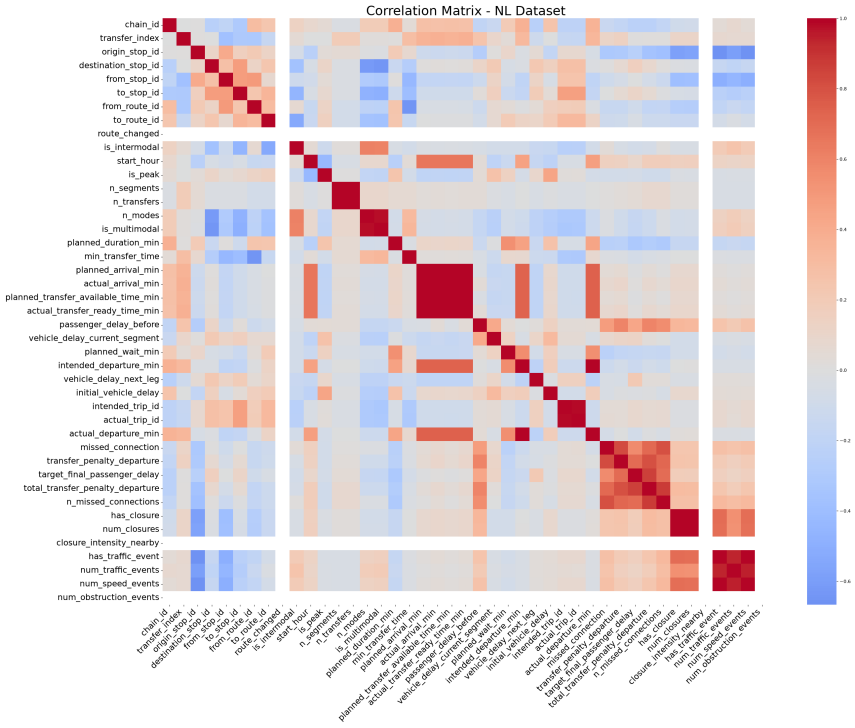


Figure A.1: Multimodal correlation matrix after M1.

Appendix A

Feature Engineering and Correlation Analysis

This appendix documents the intermediate stages of the feature engineering process described in Chapter 3. Correlation matrices are presented after each modification to illustrate how redundant variables were progressively removed while preserving features required to represent temporal, operational, topological, and delay propagation phenomena. Table A.1 summarizes the sequential feature engineering process applied to the multimodal dataset. Each modification was guided by the correlation analysis presented throughout this appendix, with the corresponding correlation matrix provided for reference.

Table A.1: Summary of the feature engineering process based on the correlation analysis.

Step	Modification	Motivation	Figure
M1	Remove *arrival_delay_at_transfer* and *ready_delay_at_transfer*	Redundant with planned and actual arrival/ready time variables.	Fig. A.1
M2	Replace *passenger_delay_before* by *weighted_passenger_delay_history*	Represent cumulative passenger delay propagation instead of the current transfer state.	Fig. A.2
M3	Add *weighted_vehicle_delay_history*	Capture the temporal evolution of vehicle delays across previously travelled journey segments.	Fig. A.3
M4	Remove *n_segments*; retain *n_transfers*	Both variables describe the same journey complexity. The number of transfers provides the most direct operational interpretation.	Fig. A.4
M5	Remove *is_multimodal*; retain *n_modes*	The number of transport modes contains richer information than a binary indicator.	Fig. A.5
M6	Remove planned/actual arrival and transfer-ready time variables	Redundant with *start_hour*, *planned_wait_min* and the propagation-aware delay history features.	Fig. A.6
M7	Remove *actual_departure_min*	Highly correlated with *planned_wait_min* and scheduled departure information.	Fig. A.7
M8	Remove *actual_trip_id* and *intended_trip_id*	Trip identifiers uniquely identify services but do not convey predictive information.	Fig. A.8
M9	Remove *has_closure* and *has_traffic_event*	Binary indicators were redundant with the corresponding event-count variables while providing less information.	Fig. A.9
M10	Remove *total_transfer_penalty_departure* and *n_missed_connections*	The represented information was already captured by propagation-aware delay features and transfer penalty variables.	Fig. A.10
M11	Remove *route_changed*	Binary information already captured by multimodal transfer and route descriptors.	Fig. 3.11 in Section 3.11.1.1 and Chapter 3

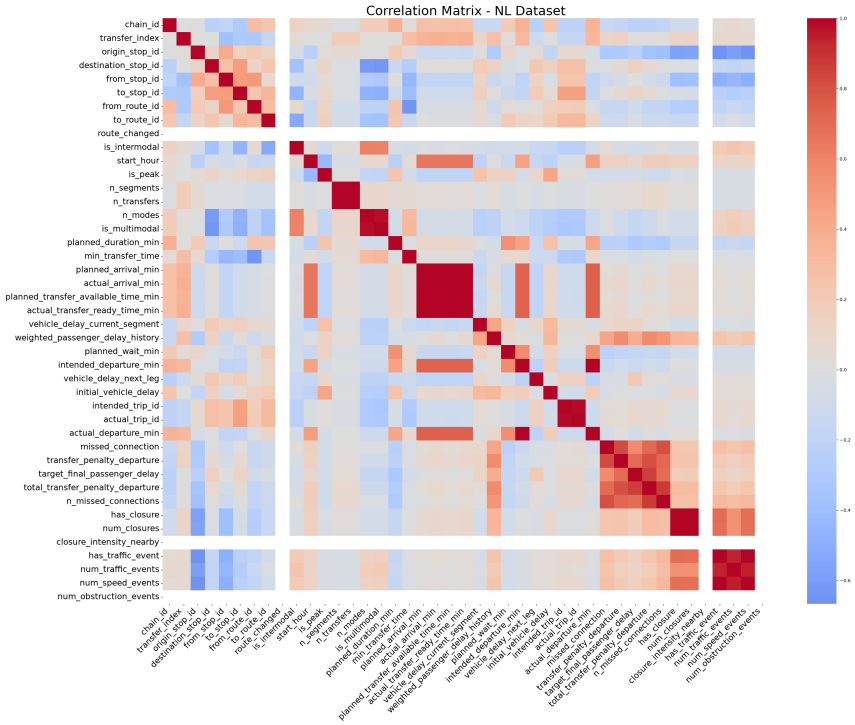


Figure A.2: Multimodal correlation matrix after M2.

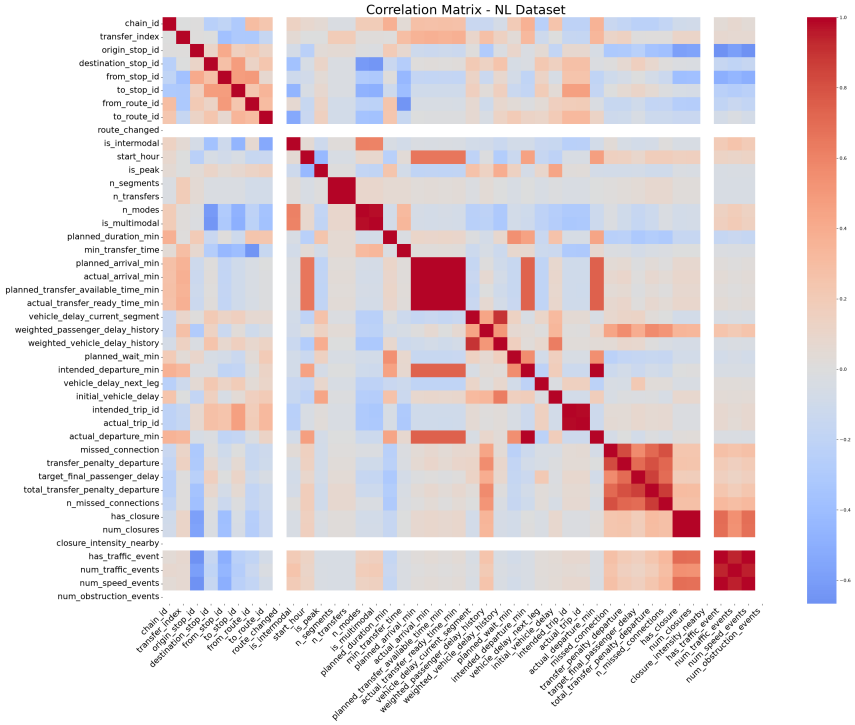


Figure A.3: Multimodal correlation matrix after M3.

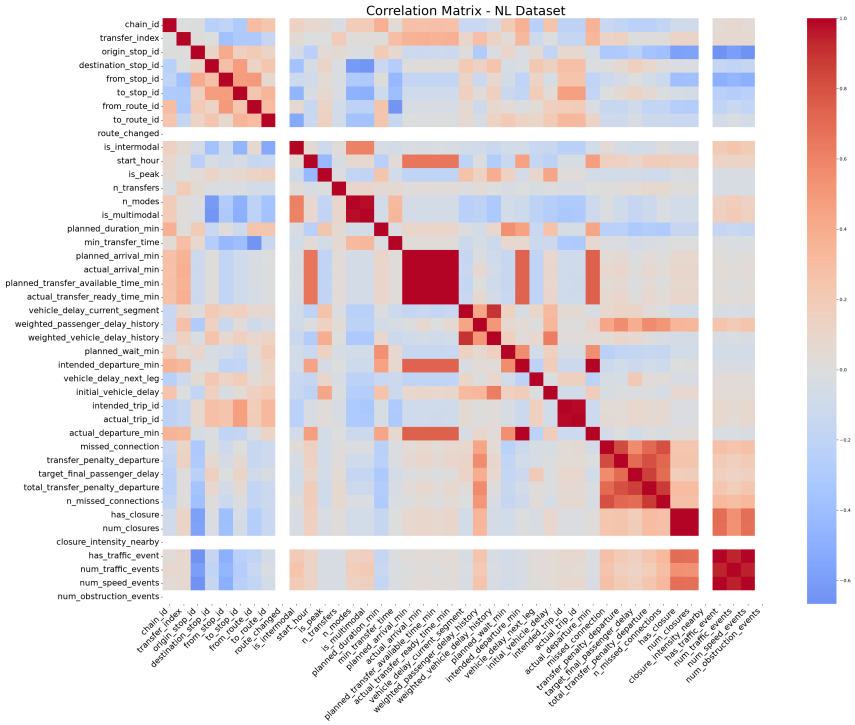


Figure A.4: Multimodal correlation matrix after M4.

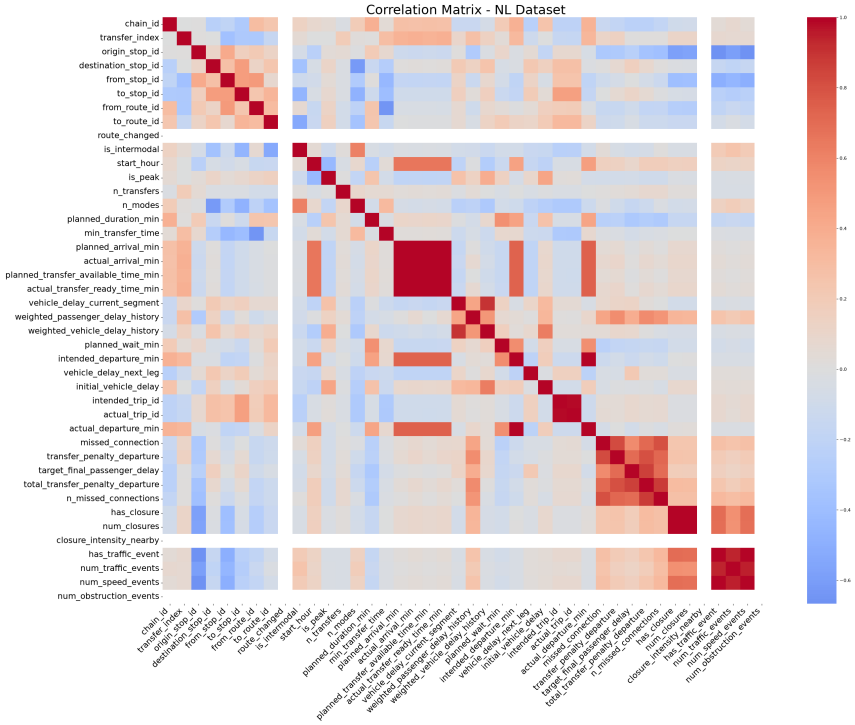


Figure A.5: Multimodal correlation matrix after M5.

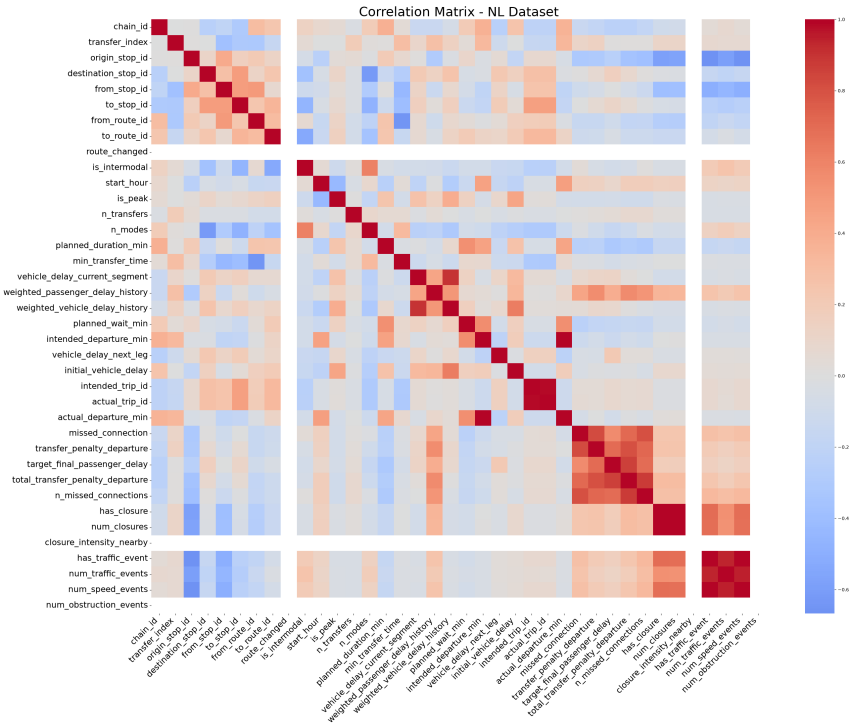


Figure A.6: Multimodal correlation matrix after M6.

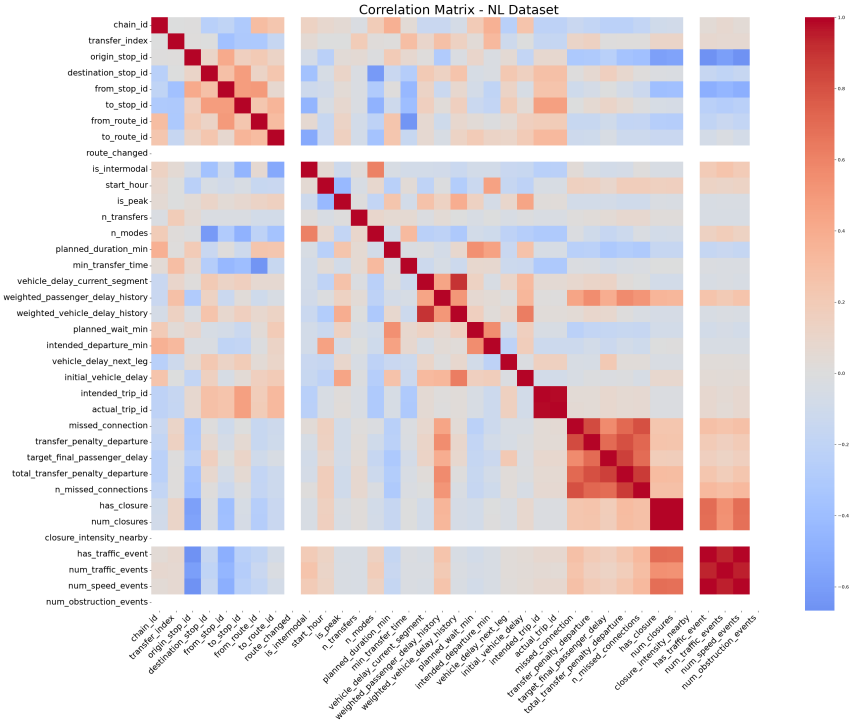


Figure A.7: Multimodal correlation matrix after M7.

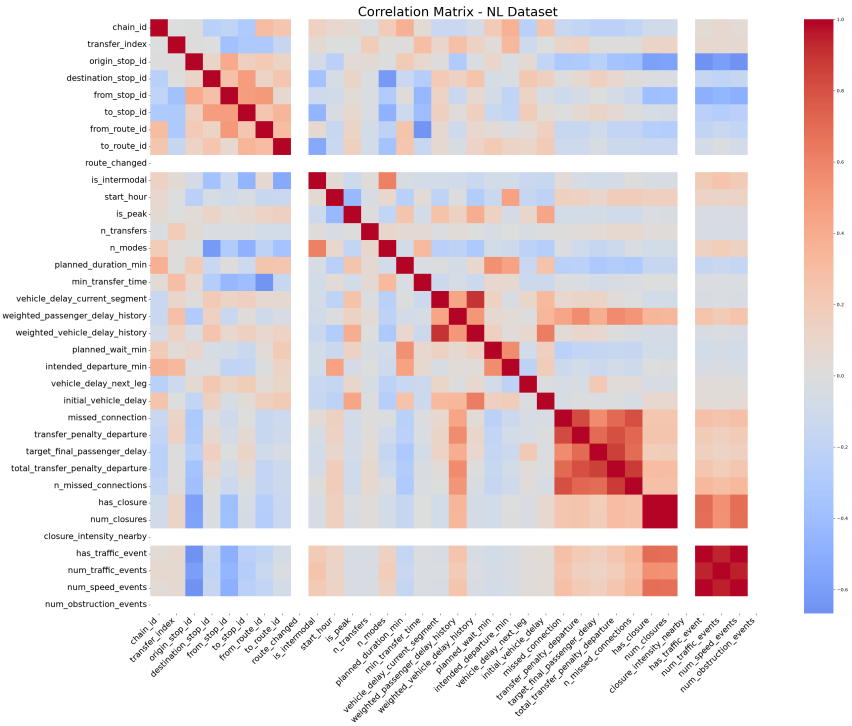


Figure A.8: Multimodal correlation matrix after M8.

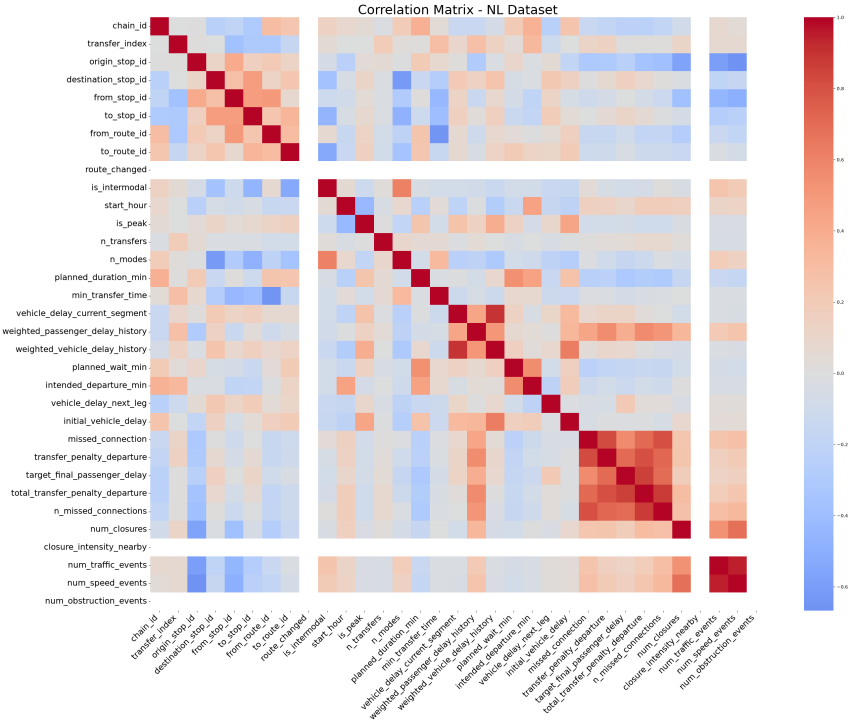


Figure A.9: Multimodal correlation matrix after M9.

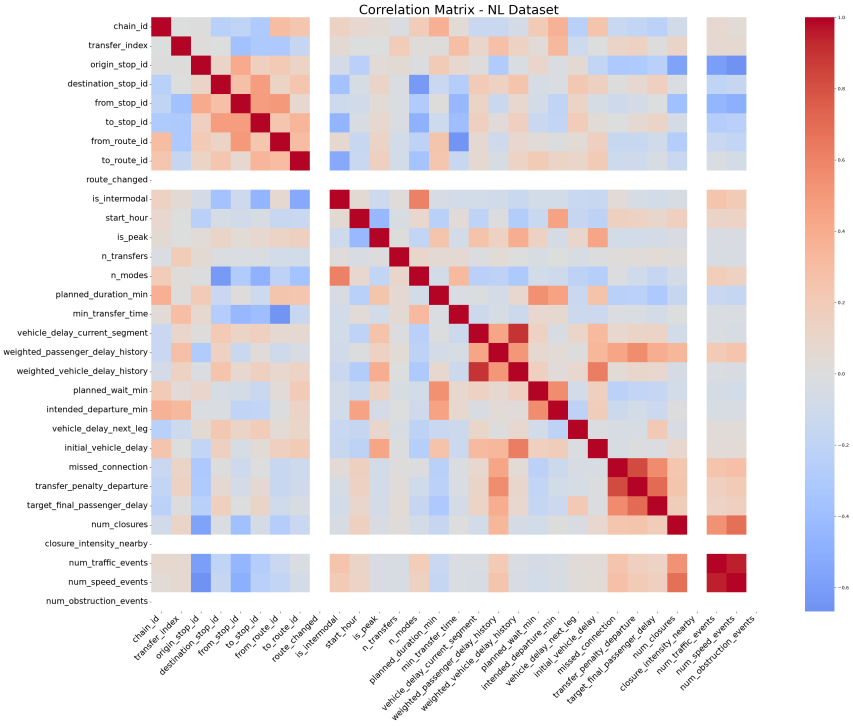


Figure A.10: Multimodal correlation matrix after M10.

Appendix B

Supplementary Experimental Results

B.1 Network Visualization Examples

Figure B.1 shows an example of the route-level filtering functionality implemented in the interactive visualisation tool. Here, a single tram route is selected from the side panel, and only its corresponding geometry is displayed on the map. This functionality supports the visual inspection of individual services and helps verify that GTFS route, trip, shape, and stop information were correctly integrated into the generated network representation.

B.2 Complete Cross-Validation Results

Fold	MAE	R^2	Accuracy	F1	Precision	Recall
1	0.4136	0.9871	0.9292	0.9285	0.9298	0.9292
2	0.4126	0.9869	0.9223	0.9208	0.9211	0.9223
3	0.4099	0.9872	0.9262	0.9248	0.9246	0.9262
4	0.4125	0.9872	0.9248	0.9239	0.9240	0.9248
5	0.4173	0.9870	0.9238	0.9229	0.9237	0.9238
6	0.4188	0.9869	0.9222	0.9215	0.9217	0.9222
7	0.4153	0.9869	0.9239	0.9229	0.9241	0.9239
8	0.4141	0.9865	0.9232	0.9220	0.9219	0.9232
9	0.4115	0.9870	0.9267	0.9256	0.9260	0.9267
10	0.4126	0.9873	0.9240	0.9228	0.9231	0.9240
Mean	**0.4138**	**0.9870**	**0.9246**	**0.9236**	**0.9240**	**0.9246**
Std.	**0.0027**	**0.0002**	**0.0022**	**0.0023**	**0.0025**	**0.0022**

Table B.1: 10-fold cross-validation results for the XGBoost model in the single-mode setting.

B.3 Additional Performance Comparisons

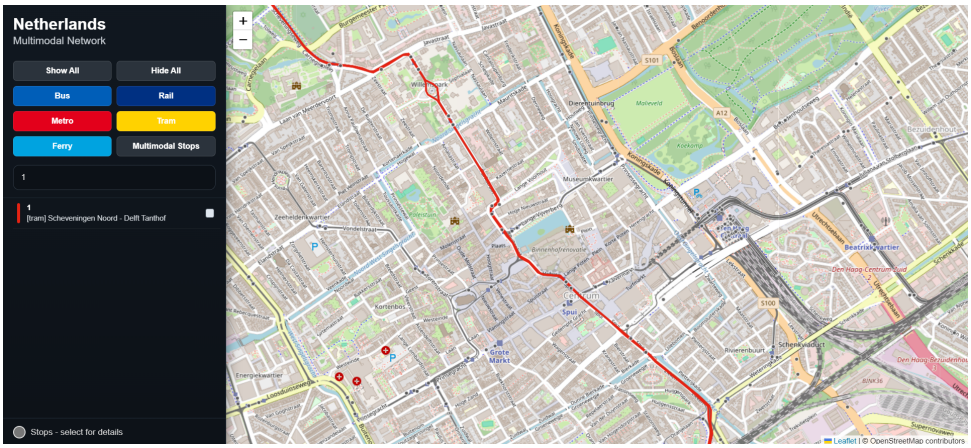


Figure B.1: Example of an individual tram route selected in the interactive network visualization interface.

Fold	MAE	R^2	Accuracy	F1	Precision	Recall
1	1.2048	–	0.9291	0.9179	0.9159	0.9291
2	1.0549	–	0.9297	0.9331	0.9382	0.9297
3	1.2561	–	0.9397	0.9289	0.9288	0.9397
4	1.3951	–	0.9314	0.9135	0.9158	0.9314
5	1.2880	–	0.9344	0.9315	0.9300	0.9344
6	1.1374	–	0.9355	0.9330	0.9332	0.9355
7	1.0556	–	0.9367	0.9326	0.9346	0.9367
8	1.0867	–	0.9598	0.9529	0.9547	0.9598
9	1.2533	–	0.9231	0.8963	0.8967	0.9231
10	1.5778	–	0.9101	0.9162	0.9246	0.9101
Mean	**1.2312**	–	**0.9353**	**0.9301**	**0.9312**	**0.9353**
Std.	**0.2170**	–	**0.0128**	**0.0144**	**0.0140**	**0.0128**

Table B.2: 10-fold cross-validation results for the LSTM model in the single-mode setting.

Fold	MAE	R^2	Accuracy	F1	Precision	Recall
1	0.6606	–	0.9793	0.9732	0.9797	0.9793
2	0.8025	–	0.9931	0.9920	0.9932	0.9931
3	0.8308	–	0.9931	0.9897	0.9863	0.9931
4	0.8785	–	1.0000	1.0000	1.0000	1.0000
5	0.7778	–	0.9862	0.9862	0.9862	0.9862
6	0.6166	–	0.9931	0.9920	0.9932	0.9931
7	0.8043	–	0.9931	0.9897	0.9863	0.9931
8	0.8077	–	0.9931	0.9936	0.9948	0.9931
9	0.7892	–	1.0000	1.0000	1.0000	1.0000
10	0.7780	–	0.9862	0.9828	0.9864	0.9862
Mean	**0.7746**	–	**0.9917**	**0.9899**	**0.9906**	**0.9917**
Std.	**0.0741**	–	**0.0060**	**0.0075**	**0.0063**	**0.0060**

Table B.3: 10-fold cross-validation results for the GNN model in the single-mode setting.

Target	Fold	MAE	R^2	Accuracy	Precision	Recall	F1
target_passenger_delay_after	1	3.6446	0.9049	0.9462	0.9302	0.9462	0.9356
	2	3.7449	0.8774	0.9471	0.9298	0.9471	0.9360
	3	3.7002	0.8875	0.9451	0.9281	0.9451	0.9344
	4	3.6971	0.9062	0.9465	0.9298	0.9465	0.9361
	5	3.7360	0.8849	0.9456	0.9301	0.9456	0.9362
	6	3.6950	0.8881	0.9482	0.9361	0.9482	0.9410
	7	3.6559	0.8819	0.9498	0.9351	0.9498	0.9405
	8	3.6802	0.8835	0.9472	0.9338	0.9472	0.9391
	9	3.6338	0.8878	0.9500	0.9351	0.9500	0.9400
	10	3.6729	0.9000	0.9488	0.9270	0.9488	0.9361
	Mean	**3.6861**	**0.8902**	**0.9474**	**0.9315**	**0.9474**	**0.9375**
	Std.	**0.0345**	**0.0095**	**0.0016**	**0.0031**	**0.0016**	**0.0023**
target_final_passenger_delay	1	7.9265	0.7411	0.9212	0.9018	0.9212	0.9095
	2	8.3148	0.7140	0.9273	0.9155	0.9273	0.9203
	3	8.2576	0.7425	0.9222	0.9019	0.9222	0.9103
	4	8.0099	0.7664	0.9140	0.8992	0.9140	0.9054
	5	8.3366	0.7326	0.9171	0.9034	0.9171	0.9086
	6	8.2028	0.7291	0.9236	0.9061	0.9236	0.9125
	7	8.1530	0.7260	0.9230	0.8992	0.9230	0.9093
	8	8.0984	0.7319	0.9199	0.9062	0.9199	0.9122
	9	7.9042	0.7315	0.9194	0.9055	0.9194	0.9114
	10	7.9093	0.7407	0.9255	0.9076	0.9255	0.9154
	Mean	**8.1113**	**0.7356**	**0.9213**	**0.9046**	**0.9213**	**0.9115**
	Std.	**0.1587**	**0.0130**	**0.0037**	**0.0046**	**0.0037**	**0.0039**

Table B.4: 10-fold cross-validation results for the XGBoost model in the multimodal setting.

Target	Fold	MAE	R^2	Accuracy	Precision	Recall	F1
target_passenger_delay_after	1	2.6911	0.9409	0.9296	0.9317	0.9296	0.9300
	2	2.8220	0.9227	0.9276	0.9289	0.9276	0.9277
	3	2.9639	0.9328	0.9214	0.9215	0.9214	0.9212
	4	2.9602	0.9310	0.9319	0.9296	0.9319	0.9300
	5	2.7173	0.9307	0.9385	0.9348	0.9385	0.9355
	6	2.9240	0.9253	0.9386	0.9359	0.9386	0.9363
	7	2.7896	0.9353	0.9371	0.9319	0.9371	0.9332
	8	2.9128	0.9263	0.9378	0.9380	0.9378	0.9363
	9	2.9081	0.9275	0.9277	0.9332	0.9277	0.9304
	10	2.8651	0.9287	0.9167	0.9413	0.9167	0.9288
	Mean	**2.8554**	**0.9301**	**0.9307**	**0.9327**	**0.9307**	**0.9309**
	Std.	**0.0923**	**0.0050**	**0.0072**	**0.0052**	**0.0072**	**0.0044**
target_final_passenger_delay	1	6.0361	0.8117	0.9355	0.9054	0.9355	0.9139
	2	6.3893	0.7913	0.9279	0.9141	0.9279	0.9197
	3	6.2425	0.8203	0.9366	0.9050	0.9366	0.9149
	4	6.0417	0.8307	0.9261	0.9041	0.9261	0.9123
	5	6.1924	0.8250	0.9364	0.8963	0.9364	0.9111
	6	6.1117	0.8077	0.9270	0.9000	0.9270	0.9116
	7	6.0935	0.8199	0.9312	0.9016	0.9312	0.9130
	8	6.1534	0.8126	0.9210	0.9108	0.9210	0.9141
	9	6.0373	0.8093	0.9385	0.9047	0.9385	0.9180
	10	5.9942	0.8226	0.9357	0.9077	0.9357	0.9189
	Mean	**6.1292**	**0.8151**	**0.9316**	**0.9050**	**0.9316**	**0.9148**
	Std.	**0.1139**	**0.0106**	**0.0055**	**0.0049**	**0.0055**	**0.0029**

Table B.5: 10-fold cross-validation results for the LSTM model in the multimodal setting.

Target	Fold	MAE	R^2	Accuracy	Precision	Recall	F1
target_passenger_delay_after	1	3.9961	0.7828	0.9443	0.9333	0.9443	0.9361
	2	4.1163	0.7641	0.9440	0.9268	0.9440	0.9332
	3	4.2539	0.7248	0.9483	0.9287	0.9483	0.9328
	4	3.9930	0.7715	0.9420	0.9387	0.9420	0.9391
	5	4.2459	0.7329	0.9486	0.9301	0.9486	0.9358
	6	4.1035	0.7490	0.9506	0.9352	0.9506	0.9391
	7	4.0341	0.7616	0.9492	0.9377	0.9492	0.9414
	8	4.0659	0.7494	0.9498	0.9380	0.9498	0.9409
	9	3.9938	0.7706	0.9465	0.9346	0.9465	0.9384
	10	4.0381	0.7979	0.9531	0.9466	0.9531	0.9478
	Mean	**4.0841**	**0.7605**	**0.9477**	**0.9350**	**0.9477**	**0.9385**
	Std.	**0.0925**	**0.0211**	**0.0032**	**0.0055**	**0.0032**	**0.0042**
target_final_passenger_delay	1	8.5226	0.6328	0.9335	0.9020	0.9335	0.9117
	2	8.7474	0.6192	0.9331	0.9068	0.9331	0.9187
	3	8.7962	0.6101	0.9332	0.9031	0.9332	0.9134
	4	8.3301	0.6327	0.9310	0.8973	0.9310	0.9104
	5	8.7089	0.6097	0.9354	0.8980	0.9354	0.9111
	6	8.7382	0.6043	0.9361	0.9031	0.9361	0.9154
	7	8.7322	0.5955	0.9346	0.9033	0.9346	0.9135
	8	8.5690	0.6122	0.9419	0.8997	0.9419	0.9190
	9	8.2648	0.6244	0.9368	0.9013	0.9368	0.9164
	10	8.2461	0.6605	0.9393	0.9029	0.9393	0.9188
	Mean	**8.5655**	**0.6201**	**0.9355**	**0.9018**	**0.9355**	**0.9148**
	Std.	**0.2035**	**0.0176**	**0.0031**	**0.0027**	**0.0031**	**0.0031**

Table B.6: 10-fold cross-validation results for the GNN model in the multimodal setting.

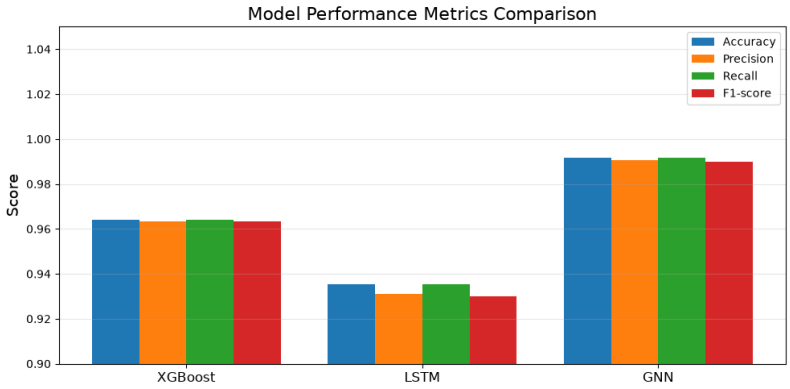


Figure B.2: Comparison of classification performance metrics across XGBoost, LSTM and GNN in the single-mode setting.

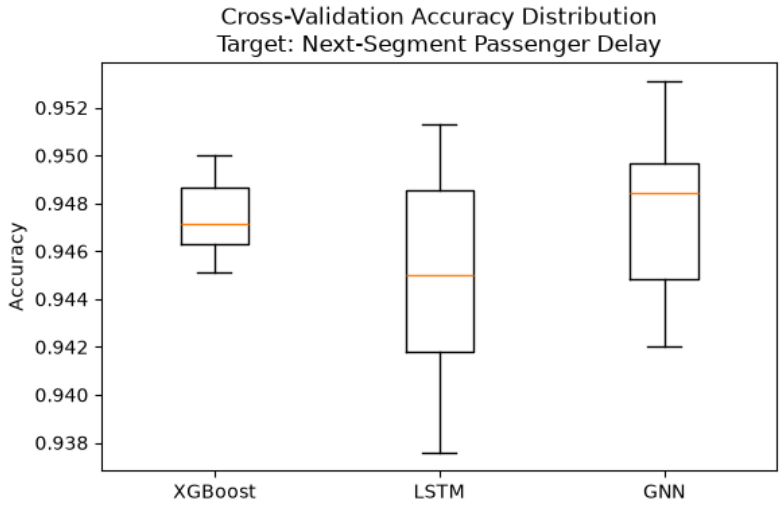


Figure B.3: Cross-validation MAE distribution for next-segment passenger delay prediction in the multimodal setting.

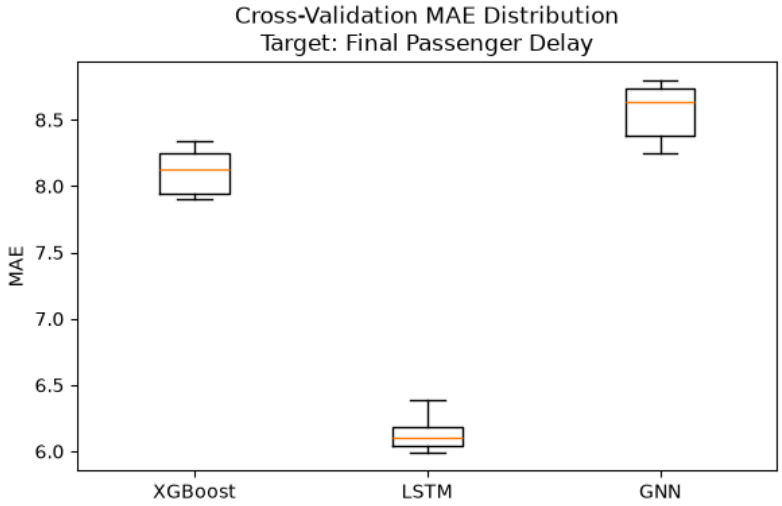


Figure B.4: Cross-validation MAE distribution for final passenger delay prediction in the multimodal setting.

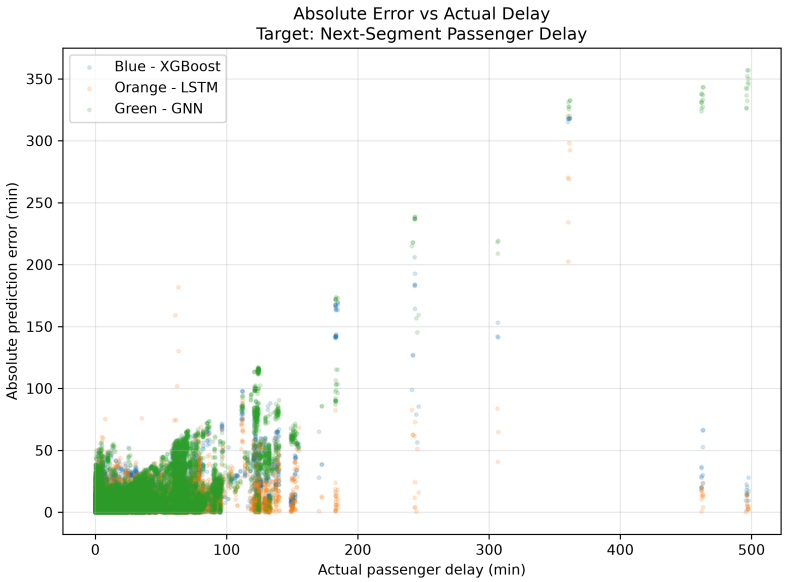


Figure B.5: Absolute prediction error as a function of actual passenger delay for the final passenger delay target.

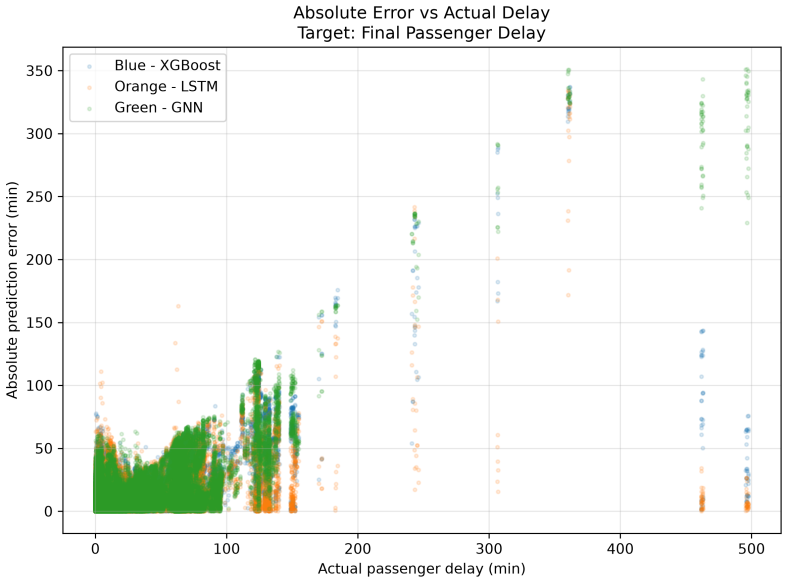


Figure B.6: Absolute prediction error as a function of actual passenger delay for the next-segment passenger delay target.