FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



Event Detection on Multivariate Time Series from CNC Machining Equipment using Machine Learning

Tiago Daniel Almeida Tavares

Mestrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Ricardo Jorge Teixeira de Sousa

July 30, 2023

© Tiago Daniel Almeida Tavares, 2023

Abstract

The Industry 4.0 paradigm and the Industrial Internet of Things growth have enabled manufacturing factories to develop real-time monitoring capabilities. Anomalies are deviations from the expected system operation, and their causes are often unknown due to complex system dynamics.

Networked sensors generate massive amounts of multivariate time series data, capturing valuable insights about the machining operations and tool performance. However, only a subset of the available time series data exhibits the relevant patterns essential for specific industrial analyses.

The research strives to advance the field by providing reliable and efficient approaches for pattern verification in the collected time series data, ultimately leading to improved efficiency, quality control, and overall performance in the manufacturing process.

An adequate system capable of detecting and extracting desired patterns from the vast data is necessary to address this challenge. The approaches proposed in this research utilise machine learning techniques and discriminatory features, specifically Linear Frequency Cepstral Coefficients, well-known for capturing critical information in the frequency domain. One-Class Classification or Deep Learning models are employed for detecting and classifying patterns.

A reproducible benchmarking framework demonstrates the proposed methodology's effectiveness and reliability. This research offers practical insights and solutions to monitor tool wear and offset compensation in Computer Numerical Control machining processes. The object of study is a real-world dataset gathered, in an industrial environment, from turning machines that contain time series with specific patterns related to wear offset.

The preliminary results show that the applied models effectively differentiate the target pattern in the time series data from the rest of the information. In particular, the Isolation Forest model attains an Accuracy of 95.11% $\pm$ 0.67 and an F1-score of 96.27% $\pm$ 0.52.

Keywords: Industry 4.0, Quality Control, CNC Turning Machining, Time Serie, Anomaly Detection, Machine Learning, Linear Frequency Cepstral Coefficient.

Resumo

O paradigma da Indústria 4.0 e o crescimento da Internet Industrial das Coisas permitiram que as fábricas desenvolvessem capacidades de monitorização em tempo real. As anomalias são desvios do funcionamento esperado do sistema, e as suas causas são frequentemente desconhecidas devido à dinâmica complexa do sistema.

Os sensores em rede geram enormes quantidades de dados de séries temporais multivariadas, captando informações valiosas sobre as operações de maquinagem e o desempenho das ferramentas. No entanto, apenas um subconjunto dos dados de séries temporais disponíveis apresenta os padrões relevantes essenciais para análises industriais específicas.

A investigação procura fazer avançar este campo, fornecendo abordagens fiáveis e eficientes para a verificação de padrões nos dados de séries temporais recolhidos, conduzindo, em última análise, a uma maior eficiência, controlo de qualidade e desempenho global no processo de fabrico.

Para responder a este desafio, é necessário um sistema adequado capaz de detetar e extrair os padrões desejados dos vastos dados. As abordagens propostas nesta investigação utilizam técnicas de inteligência computacional e características discriminatórias, especificamente Coeficientes Cepstrais de Frequência Linear, bem conhecidos por captarem informações críticas no domínio da frequência. Os modelos de "One-Class Classification" ou "Deep Learning" são utilizados para detetar e classificar padrões.

Um quadro de avaliação comparativa demonstra a eficácia e a fiabilidade da metodologia proposta. Esta investigação oferece perspectivas e soluções práticas para monitorizar o desgaste das ferramentas e o desvio de compensação nos processos de maquinagem por Controlo Numérico Computorizado. O objeto de estudo é um conjunto de dados reais recolhidos, num ambiente industrial, a partir de máquinas de torneamento que contém séries temporais com padrões específicos relacionados com a compensação do desgaste.

Os resultados preliminares mostram que os modelos aplicados diferenciam eficazmente o padrão alvo na série temporal de dados do resto da informação. Em particular, o modelo "Isolation Forest" atinge uma precisão de $95.11\% \pm 0.67$ e um F1-score de $96.27\% \pm 0.52$.

Palavras-chave: Indústria 4.0, Controlo de Qualidade, Maquinação por Torneamento CNC, Série Temporal, Deteção de Anomalia, Inteligência Computacional, Coeficiente Cepstral de Frequência Linear.

Acknowledgments

I am writing to express my sincere gratitude to the individuals whose contributions made this dissertation possible.

I am genuinely grateful to my supervisor, Ricardo Sousa, for his invaluable guidance, support, and encouragement throughout my research.

I would also like to thank my co-supervisor, João Silva, for his unwavering dedication and commitment to this project. His tireless efforts, constant availability, and insightful feedback have been essential in assuring the success of this work.

I want to acknowledge all the teachers who have contributed to my education, from my earliest years in school to my university studies. Their knowledge, passion, and dedication have inspired me to pursue my academic goals and helped shape me into who I am today.

I am deeply grateful to my family, especially my parents and sister, for their unwavering love, support, and encouragement throughout my academic journey. Their sacrifices, advice, and firm belief in me have been a constant origin of strength and inspiration.

Finally, I would like to thank my friends, Diogo Andrade, Gonçalo Roque, José Marinho, José Pedro, Pedro Ramos, and Ricardo Terrinha, for their support, incentive, and friendship throughout my studies. Their presence, humour, and camaraderie have made this journey all the more enjoyable and memorable.

Tiago Tavares

"Dream as if you'll live forever.
Live as if you'll die today."

James Dean

Contents

Abstract i

Resumo iii

Abbreviations xv

1 Introduction 1
- 1.1 Overview . . . 1
- 1.2 Motivation . . . 1
- 1.3 Problem definition and objectives . . . 1
- 1.4 Research question, hypothesis and method . . . 2
- 1.5 Main contribution and limitations . . . 2
- 1.6 Document structure . . . 3

2 Background 5
- 2.1 Industry 4.0 and Quality 4.0: control concepts for manufacturing processes . . . 5
- 2.2 Machine learning concepts and their relevance in event detection for CNC machining 6
- 2.3 CNC machining and offset adjustment in turning: ensuring precision in manufacturing . . . 7
- 2.4 Feature extraction: linear frequency cepstral coefficients . . . 8
- 2.5 One-class classification . . . 9
- 2.6 Concepts and models in deep learning . . . 10
- 2.7 Generative adversarial network . . . 11
- 2.8 Summary . . . 11

3 Anomaly detection: related work 13
- 3.1 Research methodology . . . 13
- 3.2 Multivariate anomaly detection with generative adversarial networks . . . 15
- 3.3 Time series anomaly detection using generative adversarial networks . . . 16
- 3.4 Hybrid prototype selection-based deep learning approach . . . 16
- 3.5 Fault detection using bispectral features and one-class classifiers . . . 17
- 3.6 Fault detection based on one-class deep learning for manufacturing applications . 17
- 3.7 Summary . . . 18

4 Methodology 19
- 4.1 Hardware and software description . . . 19
  - 4.1.1 Hardware . . . 20
  - 4.1.2 Software . . . 20
- 4.2 Big picture . . . 21

4.3 Data collection . . . 22
4.4 Feature engineering . . . 23
4.5 Data preparation . . . 24
4.6 Selection and design of ML models . . . 24
4.7 Model fine-tuning . . . 27
4.8 Evaluation of models . . . 27
4.9 Offset monitoring . . . 28
4.10 Summary . . . 29

5 Results and discussion **31**
5.1 Results . . . 31
5.2 Performance comparison and visualisation . . . 32
5.3 Statistical analysis . . . 33
5.4 Advantages and disadvantages . . . 36
5.5 Summary . . . 38

6 Conclusions and future work **41**
6.1 Problem and objectives review . . . 41
6.2 Research findings . . . 41
6.3 Limitations . . . 42
6.4 Future work . . . 42
6.5 Conclusion . . . 43

A Additional graphical representations **45**
A.1 Spider chart . . . 45
A.2 Box plot . . . 46

References **49**

List of Figures

2.1 High-precision turning centre. 8
2.2 Illustration of the wear offset on tool T03. 9

3.1 Network visualisation of top 20 keywords extracted from references. 15

4.1 Big picture of the methodology. 21
4.2 Database composed of several variables of the Computer Numerical Control (CNC) turning production process. 22
4.3 Time series of characteristics that resulted from the feature extraction process with Linear Frequency Cepstral Coefficients (LFCCs). 23
4.4 Wear offset from the x-axis of tool T03 during product P1 manufacturing. 28

5.1 Comparative analysis of performance metrics using bar plot. 32
5.2 Critical difference diagram for Accuracy, Precision, Recall, F1-score and Area Under the Curve (AUC). 35
5.3 Categorical scatterplot of Autoencoder Reconstruction for Anomaly Detection (ARAD) across all metrics. 36

A.1 Spider chart comparison for Accuracy, Precision, Recall, F1-score and AUC. . . 45
A.2 Box plot for comparison of Isolation Forest (IF) and One-Class Support Vector Machine (OCSVM) models. 46

List of Tables

4.1 Manufacturing comparison of 2 products across 3 CNC machines. 29

5.1 Performance metrics of tested models for pattern verification. 31

Abbreviations

AE	Autoencoder
AI	Artificial Intelligence
ANN	Artificial Neural Network
ARAD	Autoencoder Reconstruction for Anomaly Detection
AUC	Area Under the Curve
BPNN	Backpropagation Neural Network
CNC	Computer Numerical Control
CNN	Convolution Neural Network
DBN	Deep Belief Network
DCT	Discrete Cosine Transform
DDGAN	Deviation Distance with Generative Adversarial Network
DFT	Discrete Fourier Transform
DL	Deep Learning
FN	False Negative
FP	False Positive
GAN	Generative Adversarial Network
GPU	Graphics Processing Unit
HOS	Higher Order Statistics
IDE	Integrated Development Environment
IF	Isolation Forest
IoT	Internet of Things
ISO	International Organization for Standardization
LFCC	Linear Frequency Cepstral Coefficient
LOF	Local Outlier Factor

LSTM	Long Short-Term Memory
LSTM-RNN	Long Short-Term Memory Recurrent Neural Network
MAD-GAN	Multivariate Anomaly Detection Generative Adversarial Network
MFCC	Mel-Frequency Cepstral Coefficient
ML	Machine Learning
MLP	Multilayer Perceptron
MSE	Mean Squared Error
NLP	Natural Language Processing
NN	Nearest Neighbor
OCC	One-Class Classification
OCSVM	One-Class Support Vector Machine
PC	Principal Component
PCA	Principal Component Analysis
QC	Quality Control
RMSE	Root Mean Square Error
ReLU	Rectified Linear Unit
ResNet	Residual Network
ROC	Receiver Operating Characteristic
SPC	Statistical Process Control
SSD	Solid-State Drive
SVDD	Support Vector Data Description
SVM	Support Vector Machine
TadGAN	Time Series Anomaly Detection Generative Adversarial Network
TL	Transfer Learning
TN	True Negative
TP	True Positive
TPR	True Positive Rate

Chapter 1

Introduction

1.1 Overview

Industry 4.0 and Quality Control (QC) represent a new era of manufacturing characterised by intelligent, interconnected systems that aim to enhance efficiency and productivity. To tackle the challenges posed by this rapidly evolving landscape, the combination of Internet of Things (IoT) and Machine Learning (ML) techniques has emerged as a powerful approach [1]. This synergy enables the processing and analysis of massive volumes of data generated by IoT devices while uncovering complex relationships and patterns that were previously difficult to discern [2].

1.2 Motivation

This work aims to address the challenges posed by the Industry 4.0 paradigm. With the sensing of CNC machines, gathering massive volumes of data to study the productive process is possible. The complexity of the collected signals poses significant challenges that demand advanced analytical techniques for effective processing and analysis. ML techniques can process and analyse this data, learning the patterns identified in order to make predictions. Jasil, a Portuguese company specialising in CNC technology for precise metallic workpieces, needs to develop a method of monitoring the production process, particularly the offset adjustment of the CNC machine tool. This method would only apply to time series with a given behaviour.

1.3 Problem definition and objectives

The problem addressed in this dissertation is the need for a reliable and efficient approach to verifying patterns in time series data collected from CNC machines. This approach is essential for monitoring time series data related to wear offset adjustments.

By achieving the objectives, this research aims to advance event detection and QC in CNC machining processes by providing a reliable and efficient approach for pattern verification in the collected time series data. The following presents the research objectives:

- **Primary objective**: develop a methodology for verifying patterns in time series data collected from CNC machines, focusing on a specific type of time series from a large dataset;
- **Secondary objective**: create a function for monitoring time series data related to wear offset adjustments.

1.4 Research question, hypothesis and method

Research question: how effectively can ML models distinguish a target time series from others?

- **Hypothesis 1**: One-Class Classification (OCC) or Deep Learning (DL) models can distinguish a target time series from others;
- **Hypothesis 2**: LFCCs are discriminative features for pattern verification.

Method: a dataset of CNC data was collected to evaluate these hypotheses. LFCCs capture relevant information in the frequency domain. An OCC or DL model detects and classifies patterns within the CNC machining process. This model allows for identifying abnormal events related to tool wear and offset monitoring. The full methodology of the research is presented in **Chapter 4**.

1.5 Main contribution and limitations

The main contributions of this work include:

- **Development of a robust methodology** for discriminating between different types of time series;
- **Provision of a reproducible benchmarking framework** for evaluating OCC and DL methods using LFCC features;
- **Integration of a function for monitoring offset** adjustment time series into a system that supports accurate selection and utilization of cutting tools.

These contributions have the potential to significantly improve efficiency, QC, and overall performance in CNC machining.

However, a limitation of this study is the reliance on data gathered from only 3 CNC machines and 2 products, which may restrict the generalizability of the proposed methodology to a broader range of CNC machining processes and equipment. Further research may be needed to determine the generalizability of the methodology to different types of CNC machining processes or time series data.

The results obtained from this research serve as evidence of the effectiveness and reliability of the proposed methodology. Specifically, the IF model achieves an Accuracy of $\mathbf{95.11}\% \pm \mathbf{0.67}$ and an F1-score of $\mathbf{96.27}\% \pm \mathbf{0.52}$.

1.6 Document structure

This dissertation is organized into six chapters. In **Chapter 1**, the foundations of Industry 4.0 are introduced, along with its associated challenges. The motivation and objectives of this research work are also outlined, which focuses on developing a methodology for pattern verification in CNC machining using ML techniques. **Chapter 2** provides the necessary background information to understand the context and fundamental concepts of the research. In **Chapter 3**, the existing literature and related work relevant to event detection and quality control QC using ML techniques are explored. **Chapter 4** delves into the methodology employed, covering aspects such as data collection, feature extraction techniques, and the implementation of multiple ML models. Moving on to **Chapter 5**, the experimental analysis results are presented, showcasing the performance of the developed methodology. A comprehensive discussion on the results is offered, including addressing limitations and strengths. Finally, in **Chapter 6**, the main findings and overall conclusions of the study are presented. This chapter summarizes the contributions made, outlines the implications and significance of the developed work, and provides suggestions for future research. Additionally, **Chapter A** delves into supplementary information and visual representations of the results obtained.

Chapter 2

Background

The background chapter of this dissertation provides essential concepts and knowledge required to understand the research context and objectives. The chapter is structured into 7 sections.

Section 2.1 begins with an overview of Industry 4.0 and Quality 4.0, focusing on their control concepts for manufacturing processes. **Section 2.2** introduces ML concepts and their relevance in event detection for CNC machining. Moving on, **Section 2.3** delves into CNC machining and offset adjustment in turning, emphasising the importance of precision in manufacturing. **Section 2.4** explores feature extraction techniques, specifically LFCCs, and their application in analysing multivariate time series data. **Section 2.5** discusses OCC as an ML technique for identifying instances belonging to a single class, highlighting its potential use in anomaly detection. **Section 2.6** delves into DL, exploring essential techniques for robust data analysis. **Section 2.7** explains the Generative Adversarial Network (GAN) framework, a powerful tool for capturing complex data distributions and identifying anomalies in multivariate manufacturing data. Finally, **Section 2.8** summarises key concepts from the background chapter and introduces the core ideas to be explored in the upcoming chapter about related work on anomaly detection.

Following this structured approach, the background chapter provides a comprehensive foundation for this dissertation's subsequent research and analysis.

2.1 Industry 4.0 and Quality 4.0: control concepts for manufacturing processes

Industry 4.0 marked a new era of advanced technologies and concepts, revolutionising the manufacturing sector. Integrating cyber-physical systems, the IoT, cloud computing, and Artificial Intelligence (AI) creates smart factories and enables more efficient and flexible production processes [3]. The IoT manifests through various devices such as sensors, actuators, and mobile phones, enabling interaction and communication through local or internet-based connections [4]. With the growing demand for high-quality products and increased customer expectations, Quality 4.0 has emerged as a critical component of Industry 4.0 [5], focusing on ensuring and enhancing product quality by implementing innovative approaches.

Quality is defined by International Organization for Standardization (ISO) 9000 as "the ability of a set of inherent characteristics of a product, system, or process to meet the needs of customers and other interested parties" [6]. Quality, corresponding to Juran, is "fitness for purpose"; Crosby refers to "conformance to requirements", while D. Montgomery defines that “quality is inversely proportional to variability” [6].

Quality 4.0 emphasises the integration of digital technologies and data-driven methodologies to enable real-time monitoring, analysis, and control of manufacturing processes [3]. It leverages the vast amount of data generated by interconnected devices and sensors in the production environment to improve QC mechanisms [7]. By leveraging the power of data analytics, ML, and predictive modelling, Quality 4.0 aims to proactively identify and mitigate potential quality issues, reduce defects, optimise production parameters, and enhance overall process efficiency.

Central to the concept of Quality 4.0 is the notion of control. Control in manufacturing refers to the ability to regulate and maintain desired process conditions to ensure consistent and reliable production outcomes [1]. Juran’s main message was that quality must be planned; it cannot occur by good fortune [6]. Traditional control methods, such as Statistical Process Control (SPC), have been widely employed to monitor and maintain process stability [3]. However, modern manufacturing processes’ complexity and variability necessitate more sophisticated control techniques that can adapt to dynamic environments and handle multivariate data.

Industry 4.0 has introduced forth new control concepts that leverage advanced technologies and ML algorithms [8]. These concepts aim to enhance process control by enabling real-time detection of anomalies, early identification of potential failures, and adaptive adjustment of control parameters. By checking the power of data analytics and ML, these control concepts provide a proactive and data-driven approach to QC in manufacturing.

2.2 Machine learning concepts and their relevance in event detection for CNC machining

ML has emerged as a powerful tool in various domains, enabling extracting meaningful patterns and insights from large and complex datasets [9]. The definition of ML is the field of study that makes it possible for computers to learn from data without being explicitly programmed [9, 10]. It refers to algorithms and techniques that enable machines to recognise patterns, make predictions, and modify their behaviour based on observed data.

Two valuable advantages of the utilisation of ML are:

- ML **models can be continually refined and adapted** in dynamic industrial environments to accommodate evolving circumstances, ensuring optimal performance and adaptability [11];
- ML **enables greater automation**, minimising the need for manual intervention and reducing human errors [9], thereby enhancing system efficiency and cutting labour costs.

Leveraging these advantages, ML has the potential to revolutionise product QC, leading to increased efficiency and enhanced product quality [10, 12].

Modern machining systems produce a continuous stream of data that needs to be processed and analysed in real time. ML techniques provide the necessary scalability and efficiency to handle such data volumes, allowing for the extraction of valuable insights and enabling timely decision-making [13].

Various ML techniques can be employed in event detection for CNC machining. Based on identified patterns, supervised learning algorithms can be trained on labelled data to classify events as normal or abnormal. Unsupervised learning techniques can identify hidden patterns and anomalies in machining without needing labelled data. Reinforcement learning approaches can also be explored to enable adaptive control and decision-making in CNC machining systems [9].

One fundamental problem is detecting abnormal events in CNC machining, such as tool wear, tool breakage, or process deviations, which may affect product quality and performance. ML can help identify these events by learning patterns from historical data and recognising deviations from normal behaviour. Additionally, ML is used for predictive maintenance, where the goal is to anticipate equipment failures and schedule maintenance activities proactively, minimising downtime and optimising production efficiency [13].

2.3 CNC machining and offset adjustment in turning: ensuring precision in manufacturing

CNC machining, an area greatly influenced by ML [14], is a vital industrial manufacturing process that utilises computer software to control the movement of cutting tools and workpieces, enabling the creation of precise and complex shapes and components.

Two primary types of CNC machining processes are milling and turning, both crucial in manufacturing various precision components for multiple industries [14]. CNC machining offers several advantages over traditional methods, including enhanced accuracy, faster production times, and efficient intricate shapes and parts production [14].

Milling involves rotating a cutting tool along multiple axes (x, y, and z) to remove material from a workpiece. This process is ideal for creating intricate shapes and rapidly removing large material quantities. On the other hand, turning involves rotating the workpiece while using a stationary cutting tool to achieve high accuracy in creating cylindrical shapes and achieving tight tolerances.

In CNC machining, turrets are tool-holding devices that rotate to bring different cutting tools into the working position. CNC machines can feature multiple turrets with multiple toolholders, enabling efficient production of parts with diverse geometries and specifications [15, 16]. **Figure 2.1** illustrates a turning CNC machine, while **Figure 2.2** shows a diagram explaining the wear offset of tool T03.

In the manufacturing process of high-precision metallic pieces, tools in each turret play a critical role in 2 main stages: rough machining and finishing machining. The rough machining stage removes excess material, while the finishing machining stage fine-tunes the shape and dimensions of the piece to meet tight tolerance requirements.



Figure 2.1: High-precision turning centre.

Due to the precision needed in the final product, even minor variations in tool wear can significantly impact the quality. The exigent requirement of precision in manufacturing is particularly true for tools in the finishing machining stage, where tolerances are even tighter. Therefore, operators often need to perform wear adjustments to maintain the desired accuracy of the finished pieces.

2.4 Feature extraction: linear frequency cepstral coefficients

Feature extraction plays a crucial role in ML applications, where the objective is analysing multivariate time series data and detecting events of interest [17]. One effective technique for feature extraction is the use of LFCCs.

LFCC is a feature extraction technique that leverages the spectral representation of time series data. LFCCs are derived from the Mel-Frequency Cepstral Coefficients (MFCCs), a widely used feature representation in speech recognition and audio processing [18]. LFCC combines Discrete Fourier Transform (DFT) and Discrete Cosine Transform (DCT) to reveal underlying patterns that may not be readily apparent in the time domain [18]. This spectral representation offers descriptive capabilities that enable adequate characterisation of the data.

The benefits of LFCCs are numerous [18]:

- **Enable extracting relevant features from multivariate time series data**. By representing the data in the frequency domain, LFCCs capture valuable information related to intricate patterns;

- **Provide a compact and informative representation of the time series data**. By selecting a subset of LFCC coefficients that capture the most discriminative features, it becomes feasible to reduce the dimensionality of the data, leading to more efficient processing and improved computational performance;

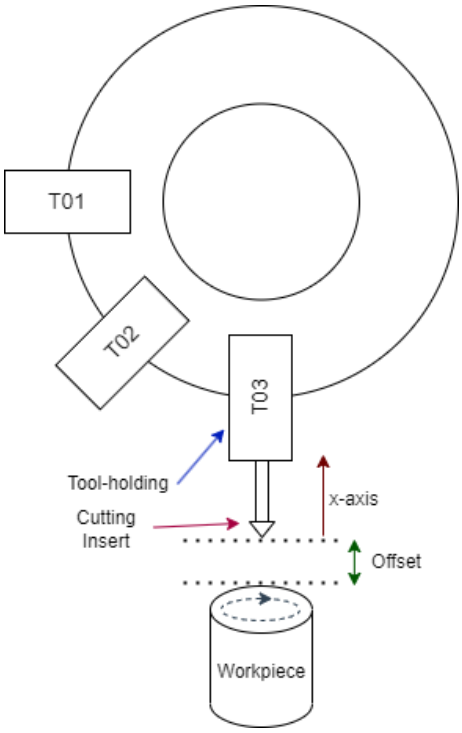


Figure 2.2: Illustration of the wear offset on tool T03.

- **Exhibit robustness against noise and other sources of interference**. The magnitude spectrum's logarithmic compression and the frequency scale's linearity contribute to a resilient feature representation that can effectively handle variations and disturbances.

Using LFCCs in combination with ML algorithms will facilitate the detection of abnormal events and potential anomalies, contributing to the enhancement of QC mechanisms.

2.5 One-class classification

OCC is an ML technique employed to identify instances belonging to a single class based on their similarity to previously observed instances of that class. OCC is widely applied in anomaly detection and class verification tasks [19]. This approach is beneficial when the number of examples of the target class is limited or when instances of other classes need better definitions.

In OCC, the algorithm is trained on a set of examples from a single class to identify instances that deviate from that class. This methodology is commonly applied in anomaly detection, which focuses on identifying rare or unexpected events within a dataset.

Several popular OCC methods exist for identifying instances belonging to a single class based on their similarity to known examples:

- **One-Class Support Vector Machine (OCSVM)** is a popular type of OCC used for unsupervised anomaly and novelty detection. OCSVM constructs a hyperplane that maximises

the margin and captures only a tiny fraction of data points. It performs well with high-dimensional and nonlinear data with sparse and unbalanced features [20];

- **Local Outlier Factor (LOF)** detects outliers by measuring local deviations and assigning anomaly scores, making it suitable for global and local outliers, non-uniform datasets, and noise [21];

- **Isolation Forest (IF)** detects outliers by isolating them in fewer dataset partitions, demonstrating efficiency with high-dimensional datasets [22].

Each method has strengths and considerations, such as its performance with high-dimensional data, sensitivity to hyperparameters, or efficiency in detecting outliers. The OCC method's choice depends on the dataset's requirements, characteristics, and problems.

In the industrial paradigm, OCC can be applied to detect deviations in the wear offset of a tool, indicating the need for replacement or adjustment [23]. This approach is valuable as it enables tool wear detection without explicit labels indicating when it becomes worn out. It also identifies early signs of wear, allowing for preventive measures to be taken before tool failure occurs.

2.6 Concepts and models in deep learning

DL is a subset of Artificial Neural Networks (ANNs) that uses multiple layers to learn from data, delivering superior performance to traditional methods, especially with large datasets [9]. The well-known DL algorithms include Multilayer Perceptron (MLP), Convolution Neural Network (CNN), Long Short-Term Memory Recurrent Neural Network (LSTM-RNN), and Autoencoder (AE).

MLP, a feed-forward ANN, employs the backpropagation technique for model structure. CNN enhances the design of traditional ANNs with convolutional, pooling, and fully connected layers, making it applicable for tasks such as image recognition and Natural Language Processing (NLP) [9]. AE is widely used for unsupervised dimensionality reduction and feature extraction. LSTM-RNN is well-suited for sequential data analysis, making it proper for time series analysis and speech recognition [9].

Apart from these classic methods, other DL approaches serve various purposes. Deep Belief Networks (DBNs) combine simple unsupervised networks like AE with a Backpropagation Neural Network (BPNN). GAN can generate data similar to the input data. Transfer Learning (TL) usually is used to train deep neural networks with limited data by reusing pre-trained models [9].

In summary, DL contains a variety of methods that play essential roles in ML, including classification, regression, clustering, feature selection and extraction, dimensionality reduction, reinforcement learning, and more.

2.7 Generative adversarial network

A GAN consists of a generator and a discriminator. The generator inputs random noise and produces synthetic samples that resemble the desired data distribution. The discriminator acts as a binary classifier, differentiating between real and fake samples [24, 25].

During training, the generator aims to deceive the discriminator by generating realistic samples, while the discriminator tries to accurately classify real and fake samples [24, 25]. This competitive process leads to improvement in both networks.

The optimisation process involves updating the generator and discriminator weights using gradient-based methods. The generator's weights are adjusted to minimise the discriminator's ability to distinguish between real and fake samples, while the discriminator's weights are adjusted to better differentiate between the two.

GANs use a combination of loss functions to guide the training. The generator loss measures how well it fools the discriminator, promoting the generation of realistic samples. The discriminator loss measures its ability to classify samples correctly. These loss functions are optimised iteratively, updating the network weights.

Training a GAN involves an iterative process where the generator and discriminator continuously improve. The goal is to reach an equilibrium where the generator produces highly realistic samples, and the discriminator struggles to differentiate between real and fake.

In summary, GANs utilise adversarial learning to generate synthetic samples resembling the training data. The interaction between the generator and discriminator during training drives the network to produce increasingly realistic and high-quality samples.

2.8 Summary

This chapter overviews Industry 4.0 and Quality 4.0, emphasising their control concepts in manufacturing processes. It discusses the role of ML in event detection for CNC machining and the significance of precision and offset adjustment in turning machines. Additionally, it introduces LFCCs as a feature extraction technique for analysing multivariate time series data. The chapter also explores OCC, DL methods and the GAN framework as ML techniques for anomaly detection.

In the upcoming **Chapter 3**, we will delve into various approaches developed by other researchers for anomaly detection in manufacturing processes. These include using GANs to capture complex data distributions and model temporal dependencies, combining DL with prototype selection techniques, leveraging bispectral features and one-class classifiers for fault detection, and introducing a comprehensive framework based on time series prediction, residual calculation, and OCC.

Chapter 3

Anomaly detection: related work

This chapter explores related work to this research, with various approaches for anomaly detection in manufacturing processes. Anomaly detection plays a crucial role in ensuring the quality and reliability of production systems.

Section 3.1 describes step-by-step the literature research that was made. **Section 3.2** explore the utilisation of GANs for capturing complex data distributions and identifying anomalies in multivariate manufacturing data. **Section 3.3** references an adaptation of GANs to model temporal dependencies and effectively detect anomalies in time series data from manufacturing processes. **Section 3.4** presents a combination of DL with prototype selection techniques to enhance the detection of anomalies in manufacturing processes. **Section 3.5** analyses the leverage of bispectral features and one-class classifiers to identify faults and anomalies in manufacturing processes. **Section 3.6** presents a method specifically designed for manufacturing, combining time series prediction, residual calculation, and OCC to detect faults in imbalanced datasets accurately. Finally, **Section 3.7** exhibits a summary of the contents of the present chapter and provides a short preview of crucial points of the following chapter.

3.1 Research methodology

In recent years, anomaly detection has stimulated the curiosity of many researchers, causing a growth in the number of scientific publications on the topic. The following steps were applied to ensure a systematic and thorough approach to the literature search:

1. **Initial keyword query**

   For the material collection process, some detailed search queries were made on the well-known online databases "Scopus", "ScienceDirect", and "Springer Link", as well as in the famous social network "ResearchGate". The following keywords were defined for this dissertation:

   - Industry 4.0;
   - Quality Control;

- Anomaly Detection;
- CNC Turning Machining;
- Time Serie;
- Machine Learning;
- Deep Learning;
- Generative Adversarial Network;
- One-Class Classification;
- Linear Frequency Cepstral Coefficient;
- Statistical Process Control;
- Cutting Insert.

The search was not restricted to specific publication years, but the technology topics were focused on recent studies to ensure up-to-date information;

2. **Review of retrieved literature**

   The initially collected literature was filtered based on titles, abstracts, and keywords to determine the most relevant. Publications that emerged directly connected to the research topic were further examined. The reference lists of these articles were used to determine extra sources that were not found in the initial search;

3. **Recursive reference exploration**

   The recursive exploration technique was used to extend the literature search further. Essential articles and their references were analysed to identify more publications not initially retrieved;

4. **Inclusion and exclusion criteria**

   Based on the relevance and alignment with the research goals, articles that concentrated on the essential topics were retained in the review. Publications that were not peer-reviewed, not written in English, or not accessible were excluded.

The search process unveiled valuable insights and conducted the selection of essential references for the dissertation. The final evaluation resulted in 27 most relevant papers, 22 with publications from 2019 to 2023.

Figure 3.1 showcases a graph illustrating the interconnections among the top 20 keywords extracted from the references used in this dissertation, using the "Pagerank" algorithm to generate the graph. This algorithm assigns importance scores to each keyword based on their connections and influence within the reference network. The node sizes reflect the relative importance of the keywords, while the edges illustrate the co-occurrence relationships.

The discoveries from the literature review will function as a foundation for developing the presented methodology and contribute to the overall quality of the dissertation.

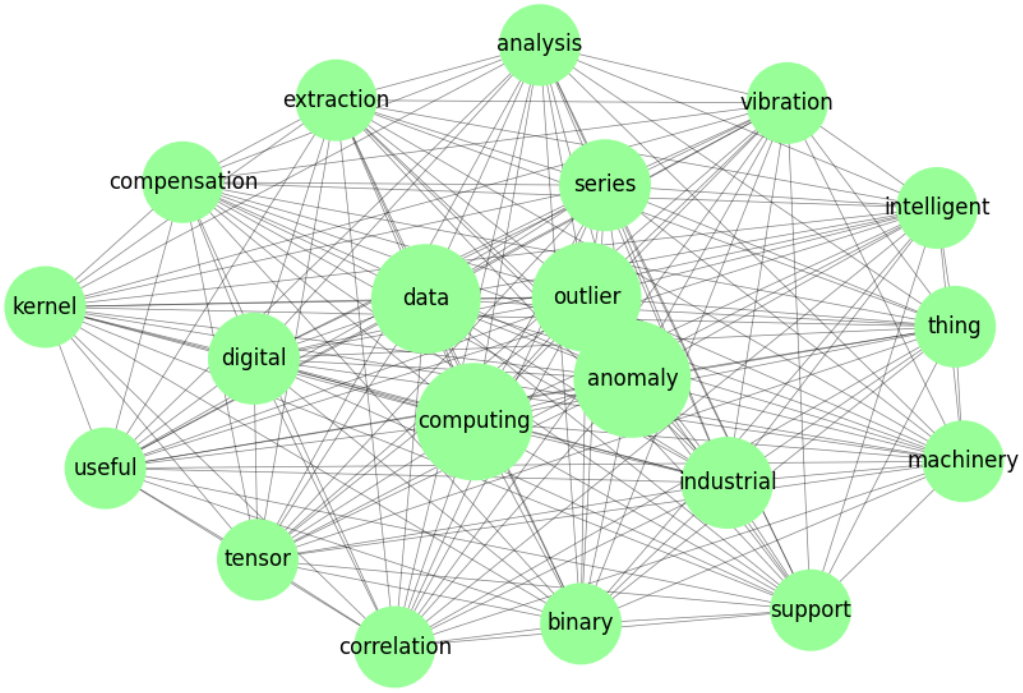


Figure 3.1: Network visualisation of top 20 keywords extracted from references.

3.2 Multivariate anomaly detection with generative adversarial networks

In [24], the challenge of unsupervised anomaly detection is addressed due to the need for labelled anomaly data for training supervised algorithms. The proposed approach utilises a DL-based strategy that leverages GANs to identify anomalies in multivariate time series data. The architecture, known as Multivariate Anomaly Detection Generative Adversarial Network (MAD-GAN), adapts the GAN framework previously used for image-related applications to analyse time series data by incorporating LSTM-RNN. Both the GAN's discriminator and generator are employed for anomaly detection.

The main objective of time series anomaly detection is to classify whether the testing data corresponds to normal data distributions or exhibits anomalous behaviour. The GAN's generator and discriminator are structured as 2 LSTM-RNNs in [24]. Following the typical GAN framework, the generator generates synthetic time series sequences based on random latent space inputs, which the discriminator then evaluates. The discriminator aims to distinguish between the generated (synthetic) data sequences and the natural (regular) training data sequences. Through iterative training, the discriminator learns to assign accurate labels to real and fake sequences, while the generator strives to deceive the discriminator by generating realistic sequences.

The results show that the proposed MAD-GAN approach outperformed popular unsupervised detection methods [24]. However, there are some limitations to consider. LSTM-RNNs can be

computationally intensive, especially when dealing with longer sub-sequences. Additionally, determining the optimal subsequence length and addressing potential model instability of GAN approaches are areas of concern highlighted in the study. This article demonstrates the effectiveness of utilising GANs for multivariate anomaly detection in time series data while acknowledging the associated challenges and potential improvements needed.

3.3 Time series anomaly detection using generative adversarial networks

There are many unsupervised approaches to time series anomaly detection: proximity-based, reconstruction-based and prediction-based methods.

The authors of [25] propose an adversarial learning approach for time series reconstruction. The underlying concept in reconstruction-based anomaly detection is to learn a model capable of encoding a segment of a time series and decoding it. A practical model should not be able to reconstruct anomalies at the same level as in regular instances due to the loss of information during encoding.

The GAN is trained to improve the reconstruction process with adversarial losses (specifically, Wasserstein losses) [25]. However, relying solely on adversarial losses does not guarantee the accurate mapping of individual inputs to the desired outputs. So, a modified cycle consistency loss is incorporated to constrain the search space of possible mapping functions, enhancing time series reconstruction.

The experimental results demonstrate the effectiveness of the proposed Time Series Anomaly Detection Generative Adversarial Network (TadGAN) method. TadGAN outperforms baseline methods regarding the averaged F1-score across multiple datasets [25]. Additionally, window-based reconstruction errors outperform pointwise methods. Combining both reconstruction errors and critic outputs yields more robust anomaly scores [25].

In [25] is delivered an innovative approach using GANs for time series anomaly detection. By leveraging adversarial learning and incorporating modified cycle consistency loss, TadGAN performs better than baseline methods, offering more accurate and robust anomaly detection capabilities.

3.4 Hybrid prototype selection-based deep learning approach

A novel methodology for training a hybrid system for anomaly detection is present in [26]. The approach combines feature extraction and classification using separate models.

The authors of [26] use a CNN as the feature extractor, which learns to reconstruct input information based on a maximum likelihood criterion. Prototype selection using Nearest Neighbor (NN) enhances the feature extraction and separates regular instances from anomalous ones.

The anomaly detector consists of 2 stages. The first stage involves a CNN-based feature extractor that condenses input data, specifically spectrograms obtained from vibration signals of

industrial equipment. The second stage uses an NN-based one-class classifier to classify feature vectors based on the average distance to the closest set of training prototypes. Labels of normal or anomalous are assigned depending on whether the average distance exceeds a threshold [26].

The article provides a detailed description of the fault detection system, including the layers of the feature extractor and their functions. Linear activation functions are chosen for convolutional and dense layers to align with the maximum likelihood criterion and a modified Root Mean Square Error (RMSE)-based loss function, which aims to generate features with a Gaussian distribution.

The proposed approach optimises the feature extraction process, incorporates prototype selection, and offers an effective solution for unsupervised OCC tasks [26].

3.5 Fault detection using bispectral features and one-class classifiers

A discussion of fault detection methods based on OCC techniques is in [23]. Principal Component Analysis (PCA)-based fault detection methods convert the data into linearly uncorrelated Principal Components (PCs), which form an orthogonal basis set. The k-NN classifier labels a sample by calculating its similarity with samples in the training data. It identifies the k-NN and computes the squared distance for each sample. Support Vector Data Description (SVDD) fits a hypersphere around the target class using optimisation techniques. The centre and radius of the hypersphere are determined by minimising the structural error.

The proposed fault detection method involves training the one-class classifiers and testing fault detection [23]. The method involves extracting 2 Higher Order Statistics (HOS) features using specific equations. The standard HOS training data is used to extract eigenvectors through PCA. Both training and testing data are projected into the common eigenvector space. The standard "HOS + PCA" data is used to train the one-class classifiers, while the testing "HOS + PCA" data is used for fault detection.

A publicly available manufacturing process dataset obtained from a semiconductor metal etch process is the object of the study. The proposed fault detection method incorporates HOS features and PCA for training the one-class classifiers [23].

3.6 Fault detection based on one-class deep learning for manufacturing applications

A fault detection module for manufacturing applications using one-class DL is demonstrated in [27]. The object of study is a dataset gathered from a pilot production from the die-casting process. The proposed framework comprises 3 submodules: time series prediction, residual calculation, and OCC.

The time series prediction submodule utilises 4 independent networks: MLP, Residual Network (ResNet), stacked Long Short-Term Memory (LSTM), and combined ResNet-LSTM [27].

The MLP consists of input, hidden, and output layers with batch normalisation, Rectified Linear Unit (ReLU) activation, and dropout layers. The ResNet is constructed with residual blocks

and convolutional and fully connected layers. The stacked LSTM comprises multiple LSTM layers with fully connected layers. The ResNet-LSTM combines ResNet with one-dimensional convolutional layers and stacked LSTM. These networks are trained with production success cases, using specific optimisers and hyperparameters [27].

The residual calculation involves averaging the raw measurement values and reconstructing them using the time series prediction submodule. The difference between the raw measurement and prediction is calculated as residuals for input into the OCC submodule [27].

The OCSVM and IF algorithms are trained with standardised residuals to determine production failure [27]. The hyperparameters for these algorithms are optimised using grid search.

3.7 Summary

This chapter analyses anomaly detection in manufacturing processes from the literature review. The approaches in this chapter utilise advanced techniques such as GANs, prototype selection, and OCCs to identify anomalies and faults, ensuring the reliability and quality of production systems.

Chapter 4 delivers systematic research approaches, including hardware and software descriptions. It outlines the data collection process from CNC turning machines, explains the LFCC feature extraction technique and data preparation, selects and designs ML models for pattern verification, and discusses evaluation metrics.

Chapter 4

Methodology

The methodology chapter outlines the systematic approach adopted for conducting the research. It encompasses the framework, tools, and techniques for collecting and analysing data to answer the research questions. This chapter serves as a roadmap, providing a clear and concise overview of the steps undertaken to ensure the reliability and validity of the study's findings. By following this methodology, the research aims to provide valuable insights and contribute to the existing knowledge in the field. This chapter is structured as follows:

Section 4.1 details the hardware platform and software components utilised for the research project. **Section 4.2** exposes the overall idea of the methodology of this work. **Section 4.3** explains how a dataset was collected from CNC turning machines, including details about the machines, products manufactured, data collection period, and variables measured. **Section 4.4** discusses the application of the LFCC feature extraction technique to transform the raw data into suitable features for ML models. **Section 4.5** focuses on preparing the data and establishing consistent testing conditions for evaluating different outlier detection models. **Section 4.6** presents the selected ML models for outlier detection, describing each model, its parameters, and its architecture. **Section 4.7** describes the process of fine-tuning the selected outlier detection models. **Section 4.8** explains the various evaluation metrics used to assess the outlier detection model's performance. **Section 4.9** explains the algorithm responsible for detecting the cutting insert change of the tool from the CNC machine. Lastly, **Section 4.10** summarises the essential concepts applied in the methodology roadmap and introduces the following chapter about the results obtained.

4.1 Hardware and software description

In the upcoming, we will discuss the hardware and software employed in the dissertation. The hardware section will provide an overview of the laptop used for the research, highlighting its specifications and capabilities. The software section will outline the programming language, Integrated Development Environment (IDE), packages, and custom modules to implement the research methodology.

4.1.1 Hardware

The hardware utilised for this research was an Asus Rog Strix G513IE computer. This computer is a high-performance laptop that was utilised as the hardware platform for this research project. It features advanced specifications and capabilities, providing the necessary computational power and resources to carry out the experiments and analysis required for the study.

- **Processor**: the laptop has a powerful processor, an AMD Ryzen 7. These processors are known for their high clock speeds, multiple cores, and efficient multitasking capabilities, which facilitate the execution of computationally intensive tasks;
- **Random Access Memory (RAM)**: the laptop is configured with a substantial amount of system memory, 16GB. Ample memory capacity is crucial for handling large datasets, running memory-intensive algorithms, and ensuring smooth performance during simultaneous computation and data processing tasks;
- **Graphics Processing Unit (GPU)**: the laptop incorporates a dedicated graphics card, an "NVIDIA GeForce RTX 3050 Ti" from the "GeForce RTX 30" series. This Graphics Processing Unit (GPU) offers exceptional graphics processing power, making it well-suited for ML tasks that involve training deep neural networks and performing complex computations on large matrices;
- **Storage**: The laptop has a spacious and fast storage system. It typically includes a Solid-State Drive (SSD) with a capacity ranging from 512GB. SSDs provide fast read and write speeds, enabling quick access to data and efficient handling of large datasets.

The research was conducted with a powerful and capable computing platform that enabled efficient data processing, model training, and analysis while providing a smooth user experience throughout the research workflow.

4.1.2 Software

The software used in this research project consisted of the following components:

- **Programming language**: Python was the language for implementing the research methodology. Python offers many libraries and frameworks well-suited for ML and data analysis tasks;
- **Integrated Development Environment (IDE)**: the IDE used for coding and development was Visual Studio Code, which provides a user-friendly and feature-rich environment for writing, debugging, and managing code;
- **Python packages**: several Python packages were utilised to implement various functionalities and algorithms required for data processing, feature extraction, and ML. The following packages were utilised:

1. **Keras**: the Keras library was used to build and train neural network models, incorporate regularisers, and define input, dense, and dropout layers;
2. **SciPy**: the SciPy library, along with its sub-packages, was utilised for performing statistical computations, signal processing (including the spectrogram function), and hypothesis testing;
3. **scikit-learn**: the scikit-learn library implemented ML algorithms such as IF, Local Outlier Factor (LOF), and OCSVM. It was also used for evaluating the performance of these models using various metrics, including Accuracy, Precision, Recall, and F1-score;
4. **spafe**: the spafe library extracted LFCC features from the data;
5. **Other libraries**: additional Python libraries such as NumPy, Pandas, Matplotlib, seaborn and random were used for various data manipulation, analysis, and visualisation tasks.

- **Custom modules**: a set of custom functions and utilities were explicitly developed for this research project. These functions were implemented in a separate Python module named "jasil_functions" and imported for use in the research workflow.

4.2 Big picture

The overall picture of the methodology can be divided into 6 distinct blocks, as shown in **Figure 4.1**: data collection and feature engineering, followed by the application of ML models, which are needed in the verification step, and finally, offset monitoring to obtain statistics regarding the manufacturing process.

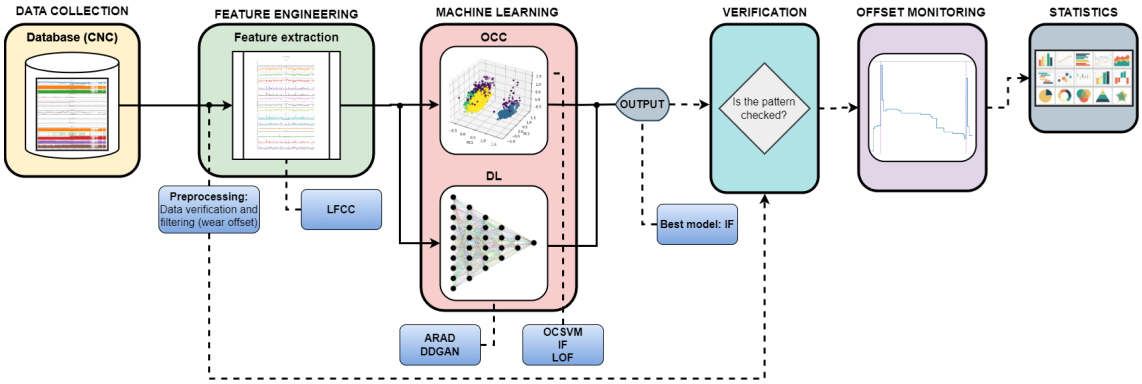


Figure 4.1: Big picture of the methodology.

In the first step, data from the CNC machines is collected, forming a database with dozens of variables. Then, the data is checked and filtered to select only the variables that represent the offset that compensates for the wear of the cutting insert of the CNC turning. Through a spectral

representation of the LFCC technique, it is possible to extract the characteristics of the time series of interest. These features feed the various ML models implemented, which are of 2 distinct types: 3 OCC models and 2 DL models. The model that shows the best performance is selected for the pattern verification, i.e., it classifies whether the time series under analysis corresponds to the pattern to be studied. If it does, a study of the time series is carried out, which will result in meaningful statistics indicating the state of the manufacturing process.

4.3 Data collection

Jasil, a Portuguese company specialising in high-precision metallic workpieces utilising CNC technology, collected a dataset. The data were obtained from 3 distinct CNC turning machines that operated continuously, 6 days a week, and 24 hours a day, with exceptions for maintenance and customer demand. During the collection period, 2 products, P1 and P2, were manufactured.

The dataset was gathered over 4 months, from October 2022 to February 2023, with a sampling rate of 1.25 Hz, resulting in approximately 150,000 data points per series. Extensive dataset analysis revealed the absence of missing values, outliers, and measurement errors.

The collection of variables from 2 turrets of the CNC machine includes variables such as temperature, spindle torque, spindle speed, wear offset, working tool, and corresponding turret. For the subsequent statistical analysis, the focus was placed on the wear offset of the x-axis as the studied variable and on 1 specific turret. In **Figure 4.2**, it is possible to observe several variables that constitute the database, identifying differences between the time series at the magnitude and frequency level.

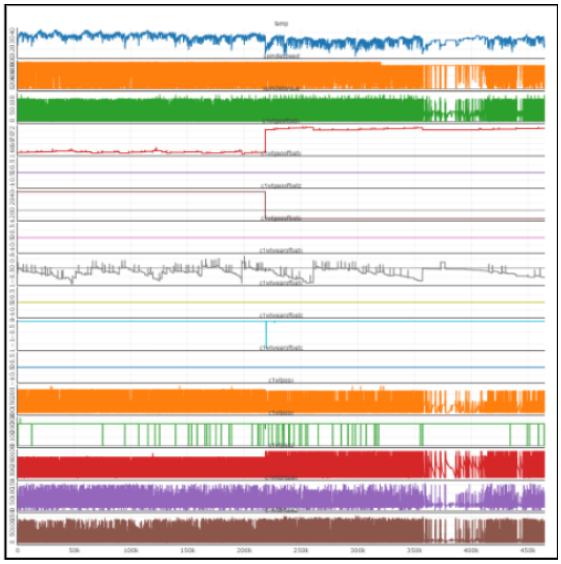


Figure 4.2: Database composed of several variables of the CNC turning production process.

Jasil's QC and maintenance department communicated that, for this study, tool T03 from turret 1 was employed to compensate for any offset variations that might occur during the machining process, particularly concerning the relevant dimensions. This choice was based on the fact that the tolerances associated with the measurements involving tool T03 are the narrowest. It is essential to mention that the design of this tool may vary depending on the product being manufactured.

4.4 Feature engineering

In line with the background knowledge presented in **Chapter 2**, we employed the LFCC feature extraction technique to transform the raw data from the studied time series into a set of features suitable for feeding ML models. LFCCs enabled us to extract relevant features from the time series data by leveraging spectral representation.

The time series data was divided into same-sized overlapping sliding windows, multiplied by a Hamming window, to capture potential desired patterns. Each window consisted of 2048 samples, and a hop size of 512 samples was used to ensure overlap. In the implementation, the number of cepstral coefficients was defined as 22.

The data suffered a normalisation process using the standard deviation of each feature to ensure the resulting features' comparability across different datasets. This step, consistent with standard practices in ML, normalised each feature to have a mean of zero and a standard deviation of one. **Figure 4.3** shows the result from this process, which consists of multiple time series that characterised the original one.

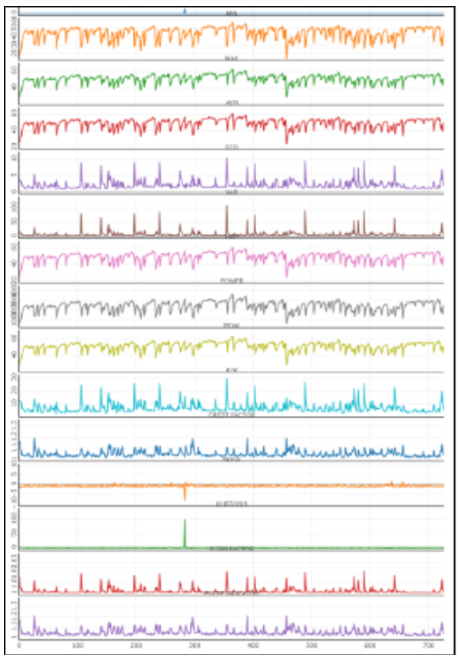


Figure 4.3: Time series of characteristics that resulted from the feature extraction process with LFCCs.

4.5 Data preparation

For the data preparation of the data previously collected, the following steps are required:

- **Data processing**: the data used in the simulation consisted of time series from CNC machining equipment related to the performance of cutting inserts. The first step was the data collection (**Section 4.3**), followed by a feature engineering process (**Section 4.4**). Both positive and negative samples are relative to the x-axis wear offset, but the positive ones are from tool T03, unlike the negative samples from tool T01. A "StandardScaler" was applied to normalise the data samples to prepare the data for analysis, ensuring consistent scaling across features;

- **Data split**: the data split structure follows a standard model development and evaluation approach. The dataset is divided into train, validation, and test sets. The split ratio for each subset is 50% for training, 25% for validation and the remaining 25% for testing. The training data only contains positive cases. The validation and test sets are similar, with a composition of 70% positive and 30% negative samples. The validation set will be used to fine-tune the hyperparameters of each model. Since the evaluation of the models consists of 10 iterations, the random state argument was diversified to ensure that these multiple runs had different data splits. This process helps to evaluate the model's performance, identify potential issues such as overfitting, and estimate its effectiveness in real-world scenarios.

4.6 Selection and design of ML models

This section will discuss the selection and design of ML models for outlier detection. We will provide a detailed explanation of each model, including its parameters and architecture.

- **OneClassSVM (OCSVM)**:

  1. **Parameters**: kernel, nu;
  2. **Description**: OCSVM is a Support Vector Machine (SVM)-based algorithm for outlier detection. It fits a hyperplane to separate regular data instances from potential outliers in a high-dimensional feature space. The 'kernel' parameter defines the type of kernel function. The 'nu' parameter determines the proportion of outliers expected in the data.

- **Isolation Forest (IF)**:

  1. **Parameters**: contamination, n_estimators, random_state;
  2. **Description**: IF is an ensemble-based algorithm for anomaly detection. It constructs isolation trees to separate outliers by randomly selecting features and splitting points at each node. The parameter 'n_estimators' is the number of base estimators (isolation

trees) in the forest, ‘contamination‘ determines the expected ratio of outliers in the data, and ‘random_state‘ appoints the random seed used for initialising the model.

- **Local Outlier Factor (LOF)**:
  1. **Parameters**: algorithm, n_neighbors, contamination, leaf_size, p, novelty;
  2. **Description**: LOF is a density-based algorithm that measures the local deviation of data points to identify outliers. It calculates the local density of a point relative to its neighbours and identifies points with significantly lower densities as outliers. The parameter ’algorithm’ select the algorithm for the NN search, ’n_neighbors’ is the number of neighbours to consider, ’contamination’ is the proportion of outliers in the data, ’p’ is the parameter for the Minkowski metric, and ’novelty’ is whether to use LOF for novelty detection (in this case it is).
- **Autoencoder Reconstruction for Anomaly Detection (ARAD)**:
  1. **Parameters**: outlier_ratio, n_features;
  2. **Description**: ARAD is an algorithm that utilises an AE to detect anomalies in datasets. With training, the AE learns to reconstruct input data accurately. During testing, the algorithm compares the reconstructed outputs with the original inputs, identifying instances with higher reconstruction errors as anomalies. The ’outlier_ratio’ represents the expected ratio of outliers, and ’n_features’ is the number of data features that will be considered to compute the reconstruction error;
  3. **Autoencoder architecture**: the AE is composed of several layers: the input layer with shape based on the data, the encoded layer that applies a dense transformation and ReLU activation function, along with L2 regularisation, a dropout layer is added after the encoding layer, an additional hidden layer with ReLU activation is included, and finally the decoding layer uses a dense layer with sigmoid activation and the same shape as the input layer;
  4. **Model training**: the AE model is compiled using the Adam optimiser and the binary cross-entropy loss function. The train data is used as both the input and target output, aiming to reconstruct the input data as accurately as possible;
  5. **Reconstruction and detection**: the reconstructed outputs are generated using the trained AE and the test data as input. The reconstruction errors represent the discrepancies between the input and reconstructed outputs. The standard deviation of the reconstruction errors for each feature is computed. From the features with the highest standard deviation, ’n_features’ are selected. The reconstruction error for each sample is the sum of the discrepancies between the selected features. These errors are normalised to the range [0, 1] using min-max normalisation. Samples are labelled as inliers or outliers based on whether the errors are above or below the ’outlier_ratio’.
- **Deviation Distance with GAN (DDGAN)**:

1. **Parameter**: std_adj;

2. **Description**: Deviation Distance with Generative Adversarial Network (DDGAN) is based on a generator network that learns to generate synthetic LFCC samples and a discriminator network that learns to distinguish between real and generated LFCC samples. The GAN model combines these networks and trains them adversarially. The generator aims to generate realistic LFCC samples to fool the discriminator, while the discriminator tries to classify real and generated samples correctly. After training, the similarity/discrepancy between generated and test samples are used to compute deviation distances to detect anomalies. The parameter 'std_adj' can vary from 0 to 3 and represents the multiplication factor of standard deviation in the deviation distance calculation;

3. **GAN architecture**: it is composed of a generator network and a discriminator network. The generator takes random noise as input and produces synthetic LFCC samples as output. It consists of dense layers with ReLU activation, followed by dropout layers to prevent overfitting. The final dense layer uses hyperbolic tangent activation to generate LFCC samples. Contrarily, the discriminator takes LFCC samples as input and classifies them as real or generated. It also consists of dense layers with ReLU activation and dropout layers. The final dense layer uses sigmoid activation to output a probability indicating the authenticity of the input sample;

4. **Model training**: the training process of the GAN network implicates several steps. First, the generator and discriminator are compiled with the Adam optimiser and the binary cross-entropy loss function. Then, within each training epoch, random noise is generated as input for the generator, which produces synthetic LFCC samples. LFCC samples are randomly selected from the training data to be combined with generated samples for the discriminator train;

5. **Deviation distance**: a set of synthetic LFCC samples is generated using the trained generator. The mean Euclidean distance between each real training LFCC sample and the set of generated LFCC samples is computed. The deviation distance is the sum of the mean value and the product of the standard deviation by coefficient 'adj_std'. The mean value represents the average distance of the distances computed before. The standard deviation measures the dispersion of these distances;

6. **Detection**: first, a set of synthetic samples are generated using the trained generator for comparison with test samples. Test samples are classified as regular or anomalies by comparing their mean Euclidean distance to the generated samples with the deviation distance computed for the model.

4.7 Model fine-tuning

An ML approach was employed to optimise the performance of the models. Grid search is a technique that systematically searches for the best combination of hyperparameters or model configurations for a given algorithm. This process was conducted using the validation subset.

In this context, grid search was applied to tune various parameters, such as the hyperparameters of the models and other relevant parameters. The goal was to find the optimal values that maximise the model's performance on the given dataset.

A parameter grid was defined for each ML algorithm, specifying each parameter's possible values or options. The grid search algorithm then exhaustively searched through all possible combinations of these parameters and evaluated the model's performance using cross-validation. The performance metric used for evaluation varied depending on the specific algorithm and problem, such as Mean Squared Error (MSE), Precision, or F1-score.

This approach allowed for systematically exploring the hyperparameter space, improving model performance and generalisation capabilities.

4.8 Evaluation of models

During the evaluation of the models, various metrics are used to assess their performance and effectiveness in identifying patterns of interest. These metrics provide insights into the model's predictions, such as Accuracy, Precision, Recall, the balance between Precision and Recall (F1-score), and the model's ability to discriminate between positive and negative cases (AUC). First, let us understand some notions about the tests used for the evaluation of predicting models, which have 4 terms to classify each prediction of the model:

- **True Positive (TP)**: it refers to the cases where the model correctly identifies the positive class as positive;
- **True Negative (TN)**: it refers to the cases where the model correctly identifies the negative class as negative;
- **False Positive (FP)**: it occurs when the model incorrectly identifies the negative class as positive;
- **False Negative (FN)**: it occurs when the model incorrectly identifies the positive class as negative.

Understanding these terms is crucial because they form the basis of various evaluation metrics. Now let us explore these evaluation metrics in more detail:

- **Accuracy** measures the overall correctness of the model's predictions. It is the quotient between the sum of True Positive (TP) and True Negative (TN) by the number of predictions;

- **Precision** focuses on the accuracy of the positive predictions made by the model. It is calculated as the quotient of TP to the total of TP and False Positive (FP). Precision tells us how well the model identifies the desired pattern among all the predicted positive samples;
- **Recall**, also known as Sensitivity or True Positive Rate (TPR), measures the ability of the model to identify the positive class. It is calculated as the quotient of TP to the total of TPs and False Negatives (FNs). Recall indicates the proportion of the actual positive cases the model successfully identifies;
- **F1-Score** is the harmonic mean of Precision and Recall. It provides a balanced measure that considers both Precision and Recall. The F1-score helps, especially when the data is imbalanced;
- **Area Under the Curve (AUC)** is the definite integral of the area under the Receiver Operating Characteristic (ROC) curve. AUC represents the separability measure between the model's true and false positive rates. AUC summarises the model's ability to discriminate between positive and negative cases.

4.9 Offset monitoring

To monitor the wear offset, a function that consists of a cutting insert change algorithm was developed. This algorithm employed in this study is based on a second-order Butterworth digital high-pass filter with a cutoff frequency of 0.125 Hz. This filter enables the capture of sudden variations in the wear offset in the x-axis (T03). Subsequently, an adaptive threshold is applied to the resulting signal to determine the peaks accurately. The methodology assumes that the analysed time series exhibits specific characteristics.

Figure 4.4 illustrates an example of the wear offset observed in a CNC machine used during the analysis of product P1. The blue line represents the wear offset, and the magenta vertical lines correspond to the cutting insert changes across the production process.

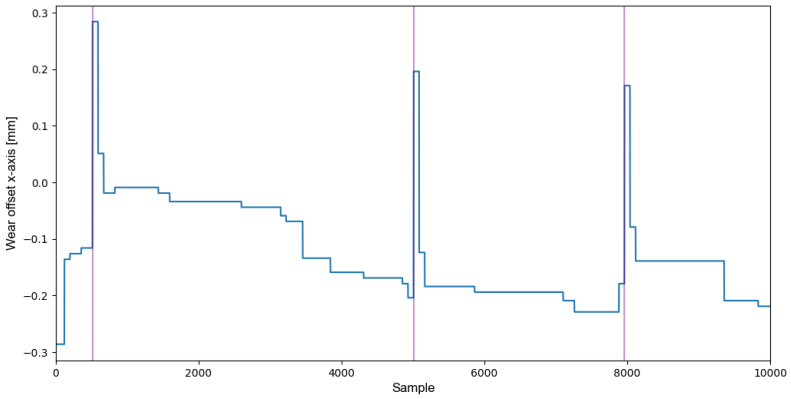


Figure 4.4: Wear offset from the x-axis of tool T03 during product P1 manufacturing.

Table 4.1 summarises the statistical measures obtained from the cutting insert change algorithm. These measures include the number of units manufactured, the number of cutting insert changes, the average units produced per cutting insert, the average offset adjustments, the average cutting insert wear, and the respective standard deviations. These statistics are directly influenced by the profile of each operator, resulting in a significant standard deviation in some parameters.

Table 4.1: Manufacturing comparison of 2 products across 3 CNC machines.

Machine	Product	Total units	Cutting changes	μ units manufactured	μ offset adjustments	μ cutting insert wear [mm]
56	P1	4869	96	50 ± 36	9 ± 6	0.423 ± 0.153
	P2	1460	42	35 ± 27	9 ± 5	0.421 ± 0.187
57	P1	4538	100	45 ± 29	9 ± 6	0.404 ± 0.194
	P2	1863	59	31 ± 22	8 ± 5	0.407 ± 0.159
217	P1	5796	129	44 ± 28	4 ± 3	0.399 ± 0.125
	P2	2953	63	45 ± 32	4 ± 3	0.399 ± 0.081

These statistical results will provide valuable insights to the quality and maintenance department regarding the production process of each machine and product.

4.10 Summary

The chapter on methodology covers several critical steps of this work. It begins with a description of the hardware and software used. Next, it presents the big picture of the methodology, with subsequent sections that detail each step of this work.

Next up, **Chapter 5** presents an in-depth analysis of the outcomes obtained from applying the ML models to the collected data. It explores the effectiveness of each model in detecting outliers and provides a comprehensive comparison of their performance.

Chapter 5

Results and discussion

This chapter presents the experimental evaluation results conducted to assess the performance of 5 ML models for pattern verification on time series data from CNC machining equipment. The tested models include DDGAN, LOF, ARAD, OCSVM, and IF. The evaluation aimed to determine the effectiveness of these models in identifying a specific pattern of the tool wear of the tool T03 across 3 machines producing 2 different products.

Section 5.1 display the results of the performance metrics obtained for each model. **Section 5.2** exhibits a graphic to visualise the inherent differences and similarities between the models and establish comparisons. **Section 5.3** presents a study of the critical difference of the metrics scores of the models with statistical analysis. **Section 5.4** presents the advantages and disadvantages of the evaluated models for pattern verification. Lastly, **Section 5.5** finalise the chapter with a summary of the key talking points and previews the next chapter.

5.1 Results

The results obtained from the evaluation, in percentage format, are summarised in **Table 5.1**, which presents the mean and standard deviation values for critical performance metrics for each model. It is possible to understand that the top 4 models (IF, OCSVM, ARAD, and DDGAN) can effectively distinguish the desired pattern in the data.

Table 5.1: Performance metrics of tested models for pattern verification.

Model	Accuracy	Precision	Recall	F1-score	AUC
IF	**95.11 ± 0.67**	**98.03 ± 0.38**	94.57 ± 0.82	**96.27 ± 0.52**	**95.38 ± 0.64**
OCSVM	93.89 ± 1.06	97.97 ± 0.40	92.76 ± 1.56	95.29 ± 0.85	94.46 ± 0.85
ARAD	92.21 ± 8.29	93.01 ± 9.25	96.86 ± 1.12	94.62 ± 4.98	89.88 ± 12.89
DDGAN	85.51 ± 0.65	90.81 ± 0.58	87.10 ± 1.07	88.91 ± 0.54	84.71 ± 0.64
LOF	72.11 ± 1.43	70.89 ± 0.97	**98.76 ± 0.77**	82.53 ± 0.80	58.75 ± 1.97

Considering that LOF results are significantly lower than the other 4 models, it is out from the following analysis.

In the following sections, we will delve into the specifics of the results with the help of a visual plot. Therefore, it is possible to identify the characteristics of each model and compare them.

5.2 Performance comparison and visualisation

The mean values of 5 performance metrics for the OCSVM, IF, ARAD, and DDGAN models are present in the bar chart in **Figure 5.1**.

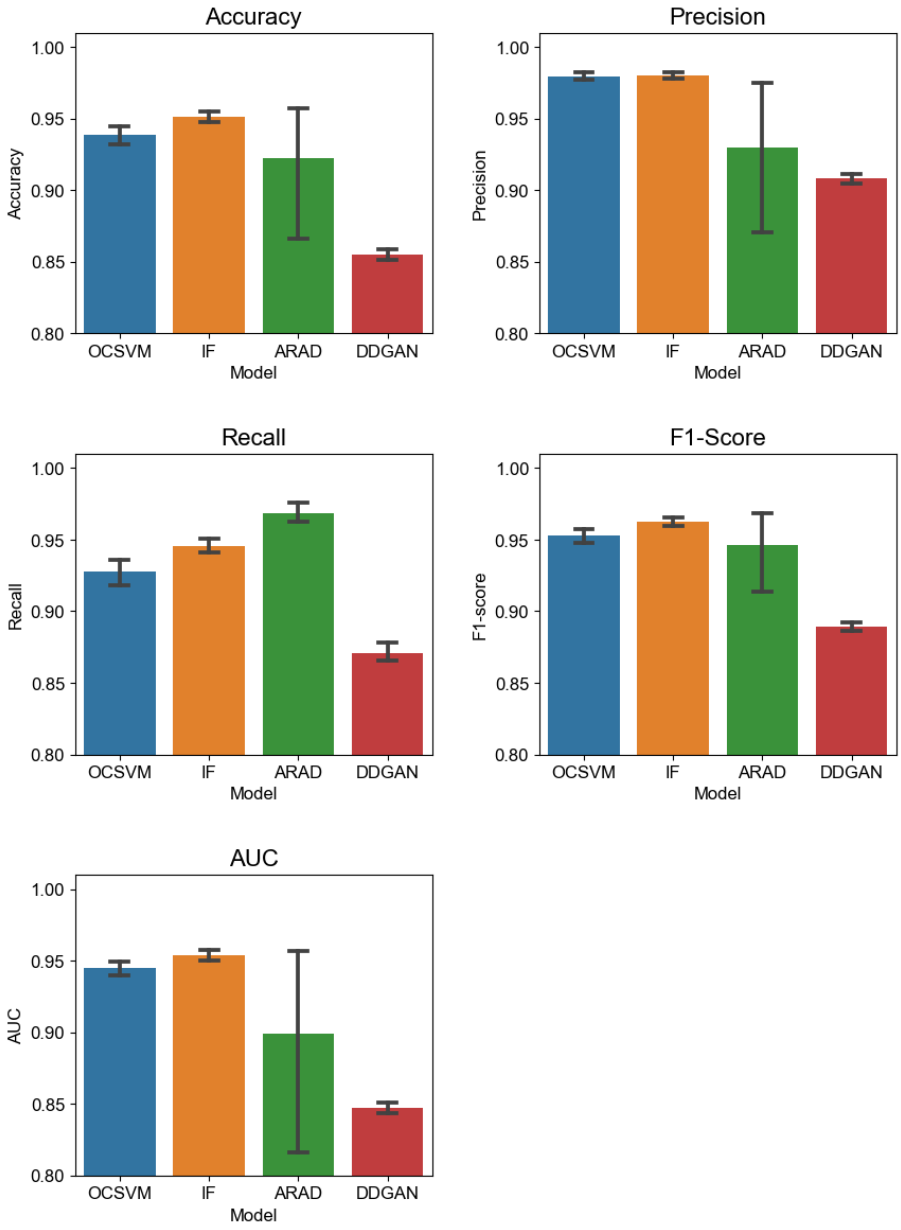


Figure 5.1: Comparative analysis of performance metrics using bar plot.

The y-axis of the bars starts at a value of 0.80 to ensure the visibility of slight differences, and the whisker in each bar is an error bar representing the confidence interval. The confidence interval estimates the uncertainty around the mean values. The confidence level in this graphic is set at 95%.

IF consistently outperforms the other models across all metrics. It shows the highest mean values for Accuracy, Precision, F1-score, and AUC. The confidence intervals for IF are narrow, indicating consistent and reliable performance. IF demonstrates to be a powerful and effective model for this problem.

OCSVM follows closely behind IF in terms of performance. While its mean values for the metrics are slightly lower, OCSVM still exhibits strong outlier detection capacities. The confidence intervals for OCSVM are also relatively narrow.

ARAD exhibits moderate performance compared to IF and OCSVM. Its mean values for the metrics are lower than the top-performing models, indicating room for improvement. Nevertheless, the confidence intervals for ARAD are wider, suggesting more variability in its performance. This variability implies that ARAD's effectiveness may vary depending on the specific dataset or application.

DDGAN consistently shows the lowest mean values for all metrics among the models considered. It indicates that DDGAN may not be as effective as the other models. However, the confidence intervals for DDGAN are narrow, indicating stability in its performance. DDGAN may require further refinement in the architecture and training.

In the appendix (**Chapter A**), for a visual and more straightforward comparison of the 5 models selected are present 2 additional graphics: a spider chart and a box plot graphic.

5.3 Statistical analysis

The Friedman-Nemenyi test is a statistical procedure used to compare multiple models or algorithms across multiple datasets or metrics. It helps determine if there is a significant difference in the performance of the models.

The critical difference is a measure used in the Nemenyi test to identify which models have statistically significant differences in their performance. It represents the minimum average rank difference required between the 2 models to conclude that they differ significantly. If the difference in the average ranks of the 2 models exceeds the critical difference value, it indicates a significant distinction between their performances.

The significance level, typically denoted as alpha (α), is a predetermined threshold that determines the level of significance required to reject the null hypothesis. In this case, the significance level is set to 5%, meaning there is a 5% chance of committing a Type I error (incorrect rejection of the null hypothesis) in this statistical test.

The average ranks provide an overall summary of the performance of each model, calculated as the mean of the ranks obtained in each iteration of the models. A lower average rank indicates a better performance, as a higher average rank indicates the lowest performance on average.

From **Figure 5.2**, it is possible to observe the pair of models with a statistically significant difference in their performances. The pairs that show non-significant differences between the 5 models across all metrics are:

- **Accuracy**:
  1. ARAD and DDGAN;
  2. ARAD and IF;
  3. ARAD and OCSVM.
- **Precision**:
  1. ARAD and IF;
  2. ARAD and OCSVM;
  3. ARAD and DDGAN.
- **Recall**:

  There are no significant differences between all models in this metric.
- **F1-score**:
  1. ARAD and IF;
  2. ARAD and OCSVM.
- **AUC**:
  1. ARAD and DDGAN;
  2. ARAD and IF;
  3. ARAD and OCSVM.

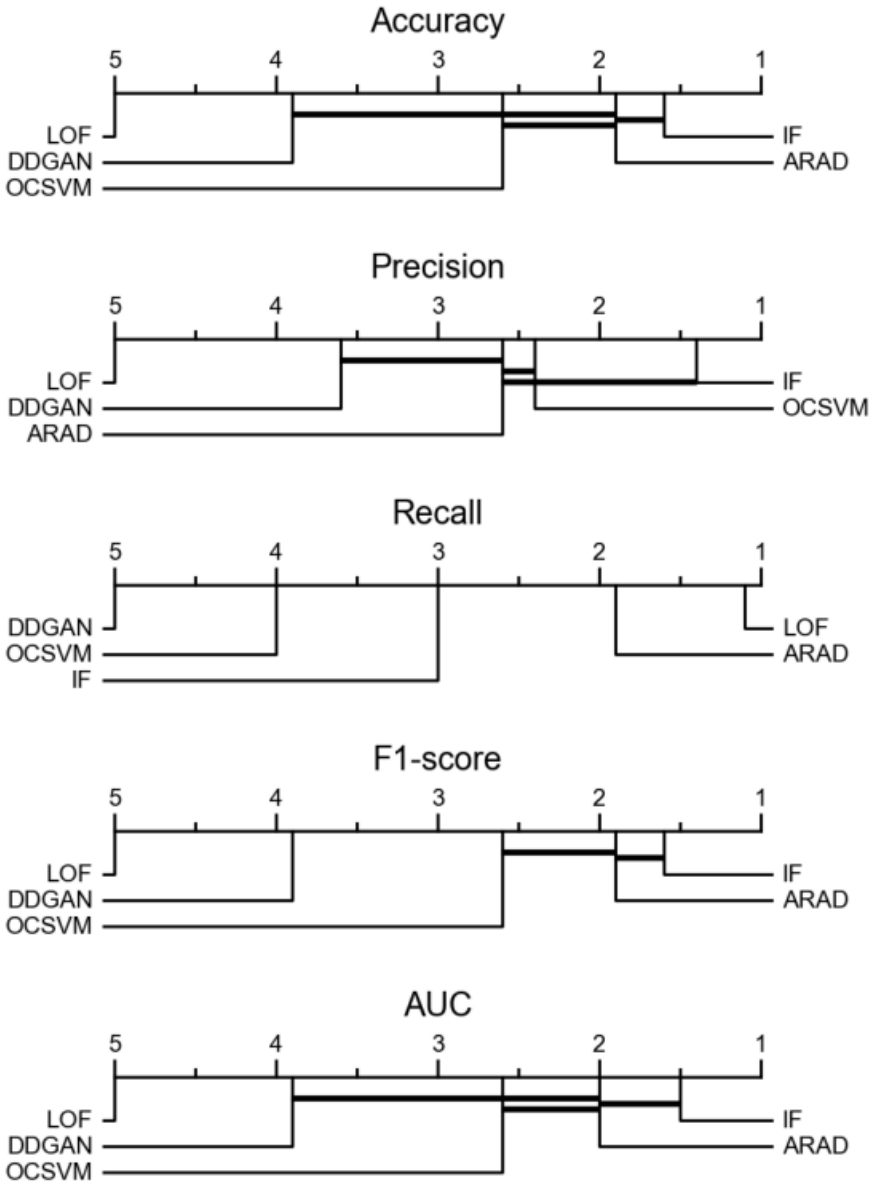


Figure 5.2: Critical difference diagram for Accuracy, Precision, Recall, F1-score and AUC.

The order of the average ranks across all metrics is very similar to the rankings observed in the bar plots. However, when the critical difference is the matter, ARAD showed that 4 of 5 metrics are not critically different from IF, and in the Recall, it has a better rank. ARAD also presents no critical difference between OCSVM and DDGAN in several metrics. However, in 4 out of 5 metrics, ARAD got second place in the average ranks, leaving OCSVM and DDGAN behind.

The results from ARAD raise some interest in analysing the scores for each simulation of every metric in detail. In **Figure 5.3**, it is possible to observe a scatterplot with the scores of the 10 simulations for all metrics. Excluding the Recall metric, there are 2 groups of points. The first with high scores comprises 6 simulations, and the second with good results represents 3 simulations.

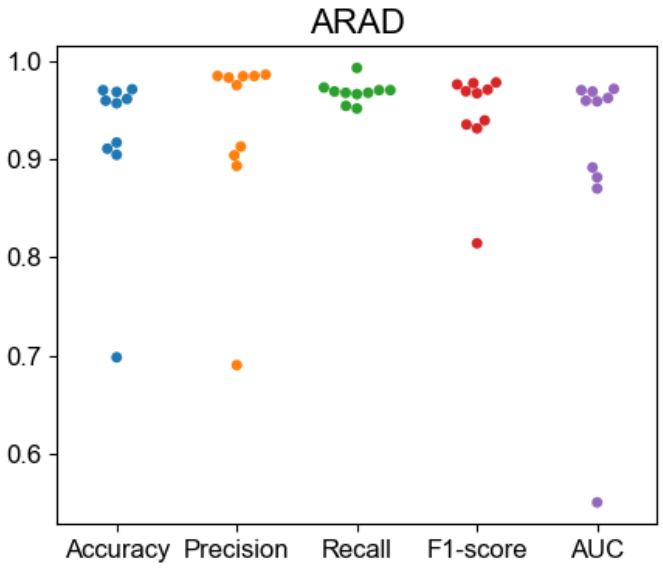


Figure 5.3: Categorical scatterplot of ARAD across all metrics.

The mystery is solved in this graphic: there is an outlier relatively distant from the rest, negatively impacting the mean and standard deviation overall values. However, the fact that there is a single outlier does not imply a problem with the model since the specific characteristics of the data in the respective simulation are likely responsible for this event.

In summary, IF got the top performance for 4 metrics, demonstrating its effectiveness in another round of analysis. Nevertheless, ARAD and OCSVM show promising results for the assigned task, demonstrating the potential of applying these models.

5.4 Advantages and disadvantages

This section discusses the advantages and disadvantages of the 5 models used for pattern verification. Understanding the strengths and limitations of these models is crucial for making informed decisions when selecting the most suitable approach. This comprehensive analysis will provide valuable insights into the capabilities and challenges associated with each model.

- **DDGAN**:
  - **Advantages**:
    1. **Generates synthetic data**: DDGAN can generate synthetic data that resembles the original dataset;
    2. **Captures complex patterns**: DDGAN's generative nature allows it to capture intricate and complex patterns present in the data. It can learn and generate realistic patterns, enhancing its ability to identify and verify patterns accurately.
  - **Disadvantages**:

1. **Training complexity**: GAN training can be challenging and require careful optimisation. This complexity may increase the computational time and resources required for training;
2. **Sensitivity to training data**: GANs heavily depend on the quality and quantity of the training data. If the training data is scarce or of insufficient quality, it can lead to suboptimal or biased generation of samples.

- **LOF**:
  - **Advantages**:
    1. **Localized outlier detection**: LOF focuses on local neighbourhoods, measures the density deviation of data points, and excels at identifying local anomalies;
    2. **Robust to varying data densities**: LOF can handle datasets with varying data densities. It can effectively detect anomalies in regions of different densities, making it adaptable to different pattern verification scenarios.
  - **Disadvantages**:
    1. **Sensitivity to parameter tuning**: the performance of LOF is highly dependent on the appropriate selection of parameters, such as the number of neighbours and the algorithm used to compute the nearest neighbour;
    2. **Scalability issues**: LOF can face scalability challenges when dealing with large datasets, which require calculating distances and densities for each data point.
- **ARAD**:
  - **Advantages**:
    1. **Semi-supervised approach**: ARAD utilizes a semi-supervised learning approach that only requires training data consisting of regular instances. It can be advantageous in situations where obtaining labelled anomalous data is challenging;
    2. **End-to-end learning**: ARAD learns to reconstruct the input data through AE training. This approach allows the model to capture complex patterns and relationships in the data.
  - **Disadvantages**:
    1. **Limited interpretability**: the interpretability of the learned features and decision boundaries might be limited. Understanding the reasoning behind the model's decisions could be challenging.
    2. **Sensitive to variability**: the natural variability in the positive data distribution can impact performance. The model might need help differentiating anomalies from regular instances if the positive data exhibits significant variations or outliers.
- **OCSVM**:

  - **Advantages**:
    1. **Simplicity and efficiency**: this is a relatively straightforward and efficient method for pattern verification. It constructs a decision boundary around the target class based on a selected kernel function;
    2. **Outlier detection without negative samples**: OCSVM can detect anomalies in the data without relying on labelled negative samples, being useful for scenarios where obtaining labelled anomalies may be difficult or costly.
  - **Disadvantages**:
    1. **Difficulty handling complex patterns**: OCSVM is effective when the patterns are to be verified with well-defined boundaries. However, it may need help with complex and overlapping patterns, as it assumes a simple decision boundary;
    2. **Sensitivity to kernel selection**: OCSVM's performance is sensitive to the choice of the kernel function. Different kernels impact the decision boundary differently and might perform better or worse depending on the nature of the patterns.

- **IF**:
  - **Advantages**:
    1. **Effective for anomaly detection**: IF is known for detecting anomalies in datasets. It performs well even when the anomalies are few and differ from most data points. It can address global and local outliers, making it suitable for various anomaly detection tasks;
    2. **Ability to handle high-dimensional data**: IF is well-suited for tasks involving high-dimensional data. It can effectively identify anomalies in datasets with many features.
  - **Disadvantages**:
    1. **Difficulty capturing complex dependencies**: IF may need help to capture complex dependencies or correlations between features in the data. It treats each feature independently, which can limit its ability to detect anomalies that rely on intricate interactions between multiple features;
    2. **Sensitivity to contamination level**: IF's performance can be affected by the contamination level in the dataset. When the proportion of anomalies is exceptionally high or low, the algorithm may need help identifying the anomalies.

5.5 Summary

This chapter presented the results and discussion of evaluating 5 ML models for pattern verification on time series data. The models were tested on CNC machining data to identify a specific pattern of tool wear. The IF model performed best, followed by OCSVM, ARAD, DDGAN, and

LOF. The performance metrics, visualization charts, and statistical analysis were used to compare the models. The advantages and disadvantages of each model were discussed, such as IF's ability to handle high-dimensional data and ARAD's sensitivity to variability.

Chapter 6 will provide a comprehensive overview of the main findings and conclusions drawn from the entire study. It will summarize the performance of the different ML models for pattern verification on time series data and highlight their strengths and limitations. Critical insights from the evaluation process and potential avenues for future research are points of reflection.

Chapter 6

Conclusions and future work

This final chapter delivers the conclusions and future directions of the dissertation. Here is a revision of the problem statement and objectives of the research, examining the key findings, highlighting the study's limitations, and suggesting potential areas for future work. The chapter concludes with an overview of the main findings and their contributions and implications.

6.1 Problem and objectives review

The problem identified in this dissertation is the degree of effectiveness of ML models in distinguishing a target time series from others. So, the objectives were to develop a methodology that effectively verifies patterns in a specific type of time series data obtained from a large dataset and to create a function for monitoring time series data related to wear offset adjustments. Relatively to patterns from time series data, the focus is on wear offset adjustments.

6.2 Research findings

The research findings demonstrate the successful application of ML techniques for pattern verification in time series data from CNC machining equipment. Out of 5 approaches based on OCC or DL, 4 can distinguish a target time series from others at a proper level, validating the effectiveness of these models. Additionally, the research confirmed the discriminative nature of LFCCs and the importance of feeding these features into the models implemented.

For experimental evaluation, 5 performance metrics were assessed for each ML model. The evaluation task focused on identifying a time series of the wear offset of tool T03.

All the models achieved a great score in the Recall metric, meaning there is a reduced number of FN cases, which means that the models learned the pattern well in training and classified the positive samples at a high level. IF consistently outperformed the other models across all metrics, except Recall, and exhibited minimal variability, highlighting its reliability. ARAD also exhibited outstanding performance, showcasing its potential to become the best model, with the need for refinement and adjustments.

The strengths of the IF model include its ability to handle high-dimensional data and its effectiveness for anomaly detection. However, a limitation of IF lies in capturing complex dependencies, which was addressed with the feature engineering done with the LFCC help.

The applicability of IF in an industrial environment holds significant promise. IF can be utilized for various applications in the industrial sector because its ability to capture complex patterns can enhance QC by accurately identifying anomalies or deviations in the machining process.

6.3 Limitations

There are several limitations to consider in this study:

- **Limited scope of data**: the study concentrated on data gathered from 3 distinct CNC turning machines and 2 specific products. This narrow scope may restrict the generalizability of the findings to other machining processes or product types;
- **Limited interpretability**: while the ML models showed promising performance, their interpretability may be limited. Comprehending the underlying reasoning behind the models' predictions and decision limitations can be difficult;
- **Computational requirements**: some models, such as DDGAN, may have high computational necessities, including training time and computational resources.

Addressing these limitations via future research will contribute to a better understanding of the proposed methods and their applicability in real-world industrial scenarios.

6.4 Future work

Based on the identified limitations, future work can focus on addressing the following aspects:

- **Data expansion and diversity**: collect data from a broader range of CNC machining equipment and product types, providing a more comprehensive understanding of the models' performance across different scenarios and increasing their applicability in diverse industrial environments;
- **Model interpretability**: investigating strategies such as model visualization, feature importance analysis, or rule extraction methods would enable a better understanding of the models' decisions, easing their practical implementation;
- **Scalability and efficiency**: regarding computational requirements, researching procedures for model optimization, parallelization, or hardware acceleration can improve the models' efficiency and enable their deployment in real-time industrial environments;

- **Transfer learning**: explore how pre-trained models on related tasks or datasets can be adapted to enhance the performance of pattern verification models for CNC machining equipment.

By handling these areas in future research, the presented models can be further refined, validated, and optimized, eventually enhancing their practical applicability in industrial settings and advancing the field of pattern verification in CNC machining equipment.

6.5 Conclusion

Our main findings indicate that the hypothesis of using OCC or DL (such as IF or ARAD) and LFCC as discriminative features for pattern verification is valid.

The significance of this work lies in its practical implications for the industry, particularly in the domain of QC. By effectively verifying patterns related to tool wear, manufacturers can proactively address potential issues, optimize machining processes, and ensure consistent product quality. This process control improves productivity, reduces downtime, and enhances customer satisfaction.

The research also presents insights into the applicability of different ML models, such as IF and ARAD, in an industrial environment. These models demonstrate the capability to handle complex and noisy datasets, capturing intricate patterns and reducing the impact of variations. Their potential for real-time monitoring and detecting anomalies makes them valuable tools for maintaining operational efficiency and preventing costly failures.

However, it is essential to acknowledge the limitations of this study. The research was conducted on a specific dataset from CNC turning machines, and generalizability to other manufacturing processes or equipment may vary.

In conclusion, this research contributes valuable insights for addressing the problem of effectively utilizing ML models to distinguish target time series from CNC machining equipment. The findings provide a foundation for improving QC practices, optimizing machining processes, and fostering the adoption of Industry 4.0 technologies. By continuing to explore and refine these approaches, we can drive innovation, enhance productivity, and ensure the competitiveness of manufacturing industries in the era of advanced automation and data-driven decision-making.

Appendix A

Additional graphical representations

This dissertation features two graphics in the appendix: a spider chart and a box plot. The spider chart presents the evaluation metrics scores for the 5 models in a radial pattern, aiding the comparison of variables across categories. On the other hand, the box plot provides a concise summary of the results from the models IF and OCSVM, highlighting central tendencies and outliers.

A.1 Spider chart

The performance metrics for the tested models are exhibited graphically by the spider chart in **Figure A.1**.

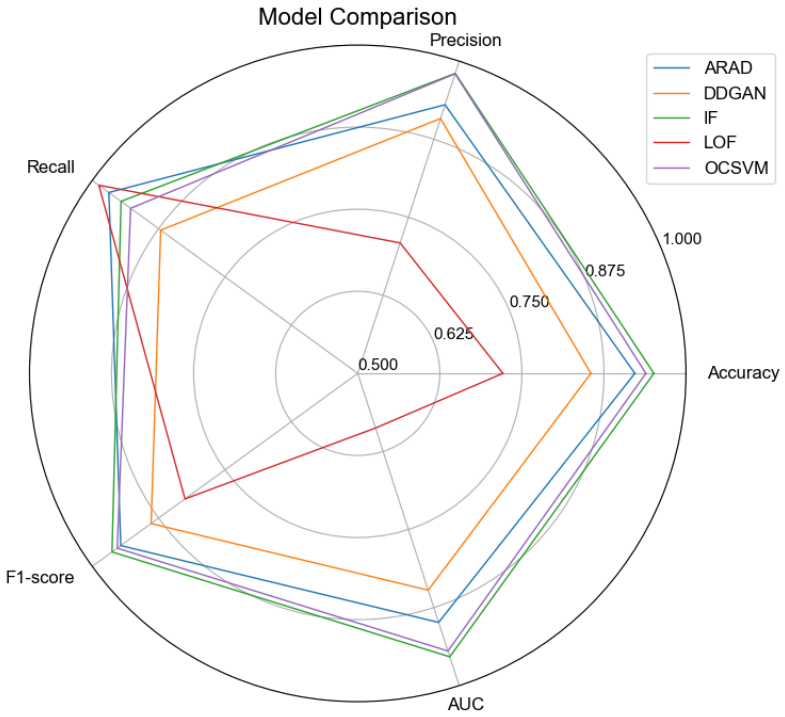


Figure A.1: Spider chart comparison for Accuracy, Precision, Recall, F1-score and AUC.

The chart allows for a comparative analysis of the models' performance across these metrics.

When comparing the results of the models, some overall similarities and dissimilarities emerge. Notably, IF, OCSVM, and ARAD show relatively high performance across numerous metrics, consistently indicating their effectiveness. On the other hand, DDGAN exhibits moderate performance compared to the top 3 models. LOF consistently performs lower than the other models, except for the Recall metric.

Overall, the spider chart concisely visualises the performance trends across the models enabling a quick assessment of their relative performance.

A.2 Box plot

The box plot graphic in **Figure A.2** visualises the performance comparison between OCSVM and IF. A box plot concisely summarises a dataset's distribution and critical statistical measures. It displays the five-number summary, which includes the minimum, first quartile, median, third quartile, and maximum.

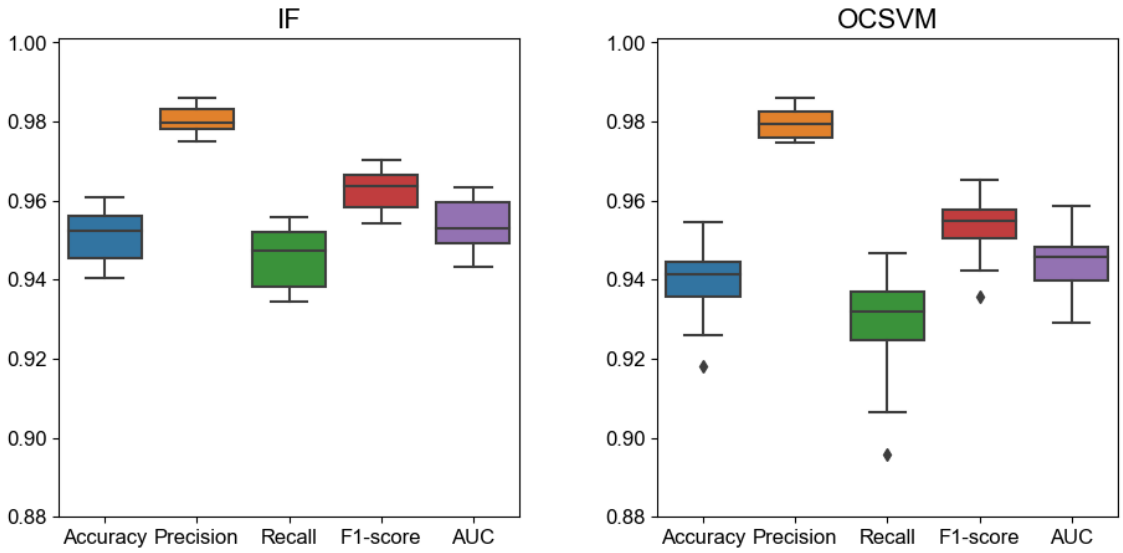


Figure A.2: Box plot for comparison of IF and OCSVM models.

Regarding overall performance, the boxes representing IF are positioned higher in the plot. IF exhibits consistently higher median and quartile values across all metrics than OCSVM, demonstrating their superior performance.

Notably, most of the performance values for IF have a shorter height between the minimum and maximum values. Also, it is worth mentioning that OCSVM presents an outlier below the average performance. This outlier, represented by a data point below the lower whisker, indicates a particular instance where OCSVM did not perform as well as expected in 1 of the 10 simulations. This outlier could be due to various factors, such as data characteristics or model configuration.

In summary, IF is the top-performing model in outlier detection. OCSVM follows closely behind, exhibiting strong performance as well, but it is not at the same level and even has an outlier in the performance results.

References

[1] Bianca Caiazzo, Mario Di Nardo, Teresa Murino, Alberto Petrillo, Gianluca Piccirillo, and Stefania Santini. Towards Zero Defect Manufacturing paradigm: A review of the state-of-the-art methods and open challenges. *Computers in Industry*, 134:103548, January 2022. URL: https://www.sciencedirect.com/science/article/pii/S016636152100155X, doi:10.1016/j.compind.2021.103548.

[2] On Making Factories Smarter through Actionable Predictions based on Time-Series Data. *Procedia Manufacturing*, 51:1207–1214, January 2020. Publisher: Elsevier. URL: https://www.sciencedirect.com/science/article/pii/S2351978920320266, doi:10.1016/j.promfg.2020.10.169.

[3] Mohd Ammar, Abid Haleem, Mohd Javaid, Shashi Bahl, and Ajay Singh Verma. Implementing Industry 4.0 technologies in self-healing materials and digitally managing the quality of manufacturing. *Materials Today: Proceedings*, 52:2285–2294, January 2022. URL: https://www.sciencedirect.com/science/article/pii/S2214785321060958, doi:10.1016/j.matpr.2021.09.248.

[4] Dinesh Soni and Neetesh Kumar. Machine learning techniques in emerging cloud computing integrated paradigms: A survey and taxonomy. *Journal of Network and Computer Applications*, 205:103419, September 2022. URL: https://www.sciencedirect.com/science/article/pii/S1084804522000765, doi:10.1016/j.jnca.2022.103419.

[5] Adriana Ventura Carvalho, Daisy Valle Enrique, Amal Chouchene, and Fernando Charrua-Santos. Quality 4.0: An Overview. *Procedia Computer Science*, 181:341–346, January 2021. URL: https://www.sciencedirect.com/science/article/pii/S1877050921002167, doi:10.1016/j.procs.2021.01.176.

[6] Kate Mccormick and Janet H. Sanders. Chapter 7 - Quality improvement programs. In Kate Mccormick and Janet H. Sanders, editors, *Quality (Second Edition)*, pages 149–185. Butterworth-Heinemann, January 2022. URL: https://www.sciencedirect.com/science/article/pii/B9780323908153000177, doi:10.1016/B978-0-323-90815-3.00017-7.

[7] Ji-Hyeong Han and Su-Young Chi. Consideration of manufacturing data to apply machine learning methods for predictive manufacturing. In *2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN)*, pages 109–113, July 2016. ISSN: 2165-8536. doi:10.1109/ICUFN.2016.7536995.

[8] Maremys Galindo-Salcedo, Altagracia Pertúz-Moreno, Stefania Guzmán-Castillo, Yulineth Gómez-Charris, and Alfonso R. Romero-Conrado. Smart manufacturing applications for

inspection and quality assurance processes. *Procedia Computer Science*, 198:536–541, January 2022. URL: https://www.sciencedirect.com/science/article/pii/S1877050921025217, doi:10.1016/j.procs.2021.12.282.

[9] Iqbal H. Sarker. Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Computer Science*, 2(3):160, March 2021. URL: https://doi.org/10.1007/s42979-021-00592-x, doi:10.1007/s42979-021-00592-x.

[10] D. Cardoso and L. Ferreira. Application of predictive maintenance concepts using artificial intelligence tools. *Applied Sciences (Switzerland)*, 11(1):1–18, 2021. doi:10.3390/app11010018.

[11] Juan Pablo Usuga Cadavid, Samir Lamouri, Bernard Grabot, Robert Pellerin, and Arnaud Fortin. Machine learning applied in production planning and control: a state-of-the-art in the era of industry 4.0. *Journal of Intelligent Manufacturing*, 31, August 2020. doi:10.1007/s10845-019-01531-7.

[12] Tiago Zonta, Cristiano André da Costa, Rodrigo da Rosa Righi, Miromar José de Lima, Eduardo Silveira da Trindade, and Guann Pyng Li. Predictive maintenance in the Industry 4.0: A systematic literature review. *Computers & Industrial Engineering*, 150:106889, December 2020. URL: https://www.sciencedirect.com/science/article/pii/S0360835220305787, doi:10.1016/j.cie.2020.106889.

[13] Serkan Ayvaz and Koray Alpay. Predictive maintenance system for production lines in manufacturing: A machine learning approach using IoT data in real-time. *Expert Systems with Applications*, 173:114598, July 2021. URL: https://www.sciencedirect.com/science/article/pii/S0957417421000397, doi:10.1016/j.eswa.2021.114598.

[14] Mohsen Soori, Behrooz Arezoo, and Roza Dastres. Machine Learning and Artificial Intelligence in CNC Machine Tools, A Review. January 2023. doi:10.1016/j.smse.2023.100009.

[15] Imhade Okokpujie, Christian Bolu, Olayinka Ohunakin, Esther Akinlabi, and Damola Adelekan. A Review of Recent Application of Machining Techniques, based on the Phenomena of CNC Machining Operations. *Procedia Manufacturing*, 35:1054–1060, January 2019. doi:10.1016/j.promfg.2019.06.056.

[16] E. García Plaza, P. J. Núñez López, and E. M. Beamud González. Efficiency of vibration signal feature extraction for surface finish monitoring in CNC machining. *Journal of Manufacturing Processes*, 44:145–157, August 2019. URL: https://www.sciencedirect.com/science/article/pii/S1526612518311149, doi:10.1016/j.jmapro.2019.05.046.

[17] Samina Khalid, Tehmina Khalil, and Shamila Nasreen. A survey of feature selection and feature extraction techniques in machine learning. In *2014 Science and Information Conference*, pages 372–378, August 2014. doi:10.1109/SAI.2014.6918213.

[18] Ricardo Sousa, Joel Antunes, Filipe Coutinho, Emanuel Silva, Joaquim Santos, and Hugo Ferreira. Robust cepstral-based features for anomaly detection in ball bearings. *The International Journal of Advanced Manufacturing Technology*, 103(5):2377–2390, August 2019. URL: https://doi.org/10.1007/s00170-019-03597-2, doi:10.1007/s00170-019-03597-2.

[19] Lorne Swersky, Henrique Marques, Joerg Sander, Ricardo Campello, and Arthur Zimek. On the Evaluation of Outlier Detection and One-Class Classification Methods. pages 1–10, October 2016. doi:10.1109/DSAA.2016.8.

[20] Bernhard Schölkopf, John Platt, John Shawe-Taylor, Alexander Smola, and Robert Williamson. Estimating Support of a High-Dimensional Distribution. *Neural Computation*, 13:1443–1471, July 2001. doi:10.1162/089976601750264965.

[21] Lan You, Qingxi Peng, Zenggang Xiong, Du He, Meikang Qiu, and Xuemin Zhang. Integrating aspect analysis and local outlier factor for intelligent review spam detection. *Future Generation Computer Systems*, 102:163–172, January 2020. URL: https://www.sciencedirect.com/science/article/pii/S0167739X19307551, doi:10.1016/j.future.2019.07.044.

[22] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation Forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422, December 2008. ISSN: 2374-8486. doi:10.1109/ICDM.2008.17.

[23] Xian du. Fault detection using bispectral features and one-class classifiers. *Journal of Process Control*, 83, September 2019. doi:10.1016/j.jprocont.2019.08.007.

[24] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, and S.-K. Ng. MAD-GAN: Multivariate Anomaly Detection for Time Series Data with Generative Adversarial Networks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11730 LNCS:703–716, 2019. ISBN: 9783030304898. doi:10.1007/978-3-030-30490-4_56.

[25] A. Geiger, D. Liu, S. Alnegheimish, A. Cuesta-Infante, and K. Veeramachaneni. TadGAN: Time Series Anomaly Detection Using Generative Adversarial Networks. pages 33–43, 2020. doi:10.1109/BigData50022.2020.9378139.

[26] Rodrigo de Paula Monteiro, Mariela Cerrada Lozada, Diego Roman Cabrera Mendieta, René Vinicio Sánchez Loja, and Carmelo José Albanez Bastos Filho. A hybrid prototype selection-based deep learning approach for anomaly detection in industrial machines. *Expert Systems with Applications*, 204:117528, October 2022. URL: https://www.sciencedirect.com/science/article/pii/S095741742200851X, doi:10.1016/j.eswa.2022.117528.

[27] Jeongsu Lee, Young Chul Lee, and Jeong Tae Kim. Fault detection based on one-class deep learning for manufacturing applications limited to an imbalanced database. *Journal of Manufacturing Systems*, 57:357–366, October 2020. URL: https://www.sciencedirect.com/science/article/pii/S0278612520301825, doi:10.1016/j.jmsy.2020.10.013.