BERGISCHE
UNIVERSITÄT
WUPPERTAL

Faculty of Mechanical Engineering and Safety Engineering

Traffic Safety and Reliability Department

Master thesis

Reliability Engineering in the Industry 4.0 through Predictive Maintenance: A Comparative Study of Failure Analysis and Evaluation Using Machine Learning Techniques

presented by

Hasan Bahtiyar Soydan

Matriculation number: **1943066**

Supervisor: **Jun. Prof. Dr. Antoine TORDEUX**

2. Supervisor: **Tim JULITZ, M. Sc.**

Author’s Declaration

I hereby certify that I have written this thesis independently and only using the sources and aids indicated by me. Both content and verbatim content have been identified as such. The thesis has not been submitted in this or a comparable form to any other examination board. I agree that the work may be viewed by third parties and cited in compliance with copyright principles. For the sake of readability, only the masculine form is used in the chapters of this report. Of course, their orientation is to be considered gender-independent in any case.

Place and date: ______________________________

Signature: ______________________________

List of Abbreviations

IoT	Internet of Things
2IR	The Second Industrial Revolution
CPS	Cyber-Physical System
DPP	Distributed Process Planning
PdM	Predictive Maintenance
LDA	Linear Discriminant Analysis
KNN	K-Nearest Neighbor
ML	Machine Learning
SVM	Support Vector Machine
ANN	Artificial Neural Networks
DNN	Deep Neural Networks
DT	Decision Tree
RF	Random Forest
AT	Air Temperature
PT	Process Temperature
RS	Rotational Speed
T	Torque
TW	Tool Wear
MF	Machine Failure
TWF	Tool Wear Failure
HDF	Heat Dissipation Failure
PWF	Power Failure
OSF	Overstrain Failure
RNF	Random Failure
PCA	Principal component analysis
MSE	Mean Squared Error

List of Figures

List of Tables

Table of Contents

Summary

Predictive maintenance has become increasingly crucial in the context of Industry 4.0. This maintenance approach aims to identify machine malfunctions before they occur, employing diverse methods to anticipate issues, or to determine optimal conditions for swift operational restoration as soon as possible.

In the theoretical background section of the master's thesis, various fundamental concepts are explored, including Industry 4.0, Cyber-physical Systems, Cloud Manufacturing, Big Data, IoT (Internet of Things), Reliability Engineering in Industry 4.0, Predictive Maintenance, Condition-Based Monitoring, and methodologies for developing data-driven predictive maintenance models. These methodologies encompass Decision Trees, Linear Discriminant Analysis, Logistic Regression, K-Nearest Neighbors, Support Vector Machines, Artificial Neural Networks, and Deep Neural Networks.

In the analysis section of the thesis, a dataset containing sensor data with various parameters is scrutinized utilizing the R programming language to evaluate and compare different predictive maintenance methods. The primary focus of the implementation within the programming environment revolves around classifying the operational state of the machine. To achieve this classification, several statistical methods and machine learning algorithms are employed, including Logistic Regression, Linear Discriminant Analysis, Decision Trees, Random Forests, Support Vector Machines, and Consensus functions such as Minimum, Maximum, and Mean values.

After interpreting all the results, it is evident that the most appropriate algorithm for this dataset is Random Forest, followed by the Maximum of algorithm and Decision Tree algorithms. Conversely, the three algorithms exhibiting the poorest results are Minimum of Algorithm, LR, and LDA.

These outcomes suggest that the dataset comprises nonlinear data, as evidenced by the superior performance of Random Forest and other nonlinear algorithms. Additionally, it appears that the function values derived through the consensus ensemble method do not exert a significant impact on the overall result.

1 Introduction

Industry 4.0 is transforming manufacturing and production processes through the integration of digital technology. An important aspect of this revolution is the use of sensors in predictive maintenance, an approach that significantly improves the efficiency and sustainability of industries. By collecting data through sensors, companies can monitor the condition of equipment in real time and predict when maintenance needs to be performed. The shift from traditional, scheduled maintenance to a predictive approach minimizes downtime and reduces the costs associated with unplanned equipment failures.

The data collected by sensors is not only enormous but also highly diverse, covering parameters such as temperature, vibration, and acoustics [1]. By analyzing this data with advanced algorithms and machine learning techniques, companies can identify patterns and anomalies that precede equipment failures. This predictive capability enables timely intervention, ensuring that maintenance is carried out only, when necessary, thus optimizing resource utilization.

Moreover, predictive maintenance in Industry 4.0 facilitates a deeper understanding of machine performance and lifecycle. It enables companies to make more informed decisions about equipment management and investments. The efficiency gains from predictive maintenance contribute significantly to reducing operational costs and improving the overall competitiveness of businesses in a rapidly evolving technological landscape.

In summary, the role of Industry 4.0 in enhancing predictive maintenance through the use of sensors is a critical component in the modernization of industries. It not only increases operational efficiency but also drives innovation, paving the way for smarter, more sustainable industrial practices.

1.1 Problem Statement

The biggest challenge lies in determining the optimal conditions and timing for machine maintenance, which cannot yet be automatically ascertained. Different statistical and machine learning methods are compared using sensor data and various parameters in a heuristic approach to optimize the situation. Sometimes, more than one method is used on the same dataset but with different data ranges to optimize the result.

In addition, the volume and variety of data generated in Industry 4.0 environments pose challenges in data collection, processing, and analysis. Machine learning algorithms require high-quality data for accurate predictions, making data management critical.

These challenges require careful planning, resource allocation, and ongoing management to ensure successful implementation and operation.

1.2 Objectives

The aim of this master's thesis is to compare data within a dataset using various statistical and machine learning methods.

The focus is to analyze failures in a dataset where machines are constantly breaking down. The objective is to reduce the current failure rates by considering different predictive maintenance methods. It will also determine which methods can more effectively classify the machines’ health through maintenance strategies, Reliability Engineering, and Industry 4.0 practices.

1.3 Structure of Thesis

This master's thesis is organized into five main sections:

Firstly, the Introduction provides a brief description of the problem that prompted the writing of this thesis, outlines its objectives, and explains how the problem will be addressed.

The second chapter, titled 'Theoretical Background,' delves into concepts crucial to Industry 4.0, such as Cyber-physical Systems, Big Data, the Internet of Things (IoT), and Sensors. This chapter will also provide a detailed exploration of reliability technologies within the context of Industry 4.0.

In the third chapter, titled 'Maintenance Strategies,' topics crucial to Industry 4.0, such as predictive maintenance and condition monitoring, are discussed. Additionally, this chapter describes various models for improved failure prediction in predictive maintenance, categorizing them under two separate headings: statistical methods and machine learning methods. These models are not only described, but the chapter also provides a forecast of their potential applications, which will be further explored in the subsequent chapter.

The next to the last section is the 'Application Example' section. It begins with a brief introduction to R, a statistical analysis program that will be used for data analysis and comparison. Then, prior to the actual data analysis, the source of the data to be used will be identified, and the dataset will be reviewed and prepared for enhanced analysis. Subsequently, data analysis will be conducted using different statistical and machine learning methods. These analyses will be visualized, and all results will be evaluated, compared, and interpreted.

Finally, the study's results are discussed, highlighting the achievements and outcomes.

2 Theoretical Background

2.1 Industry 4.0

The idea of the industrial revolution has been fundamental in shaping our comprehension of the economic, political, and social transformations that have occurred over the last two hundred years [2].

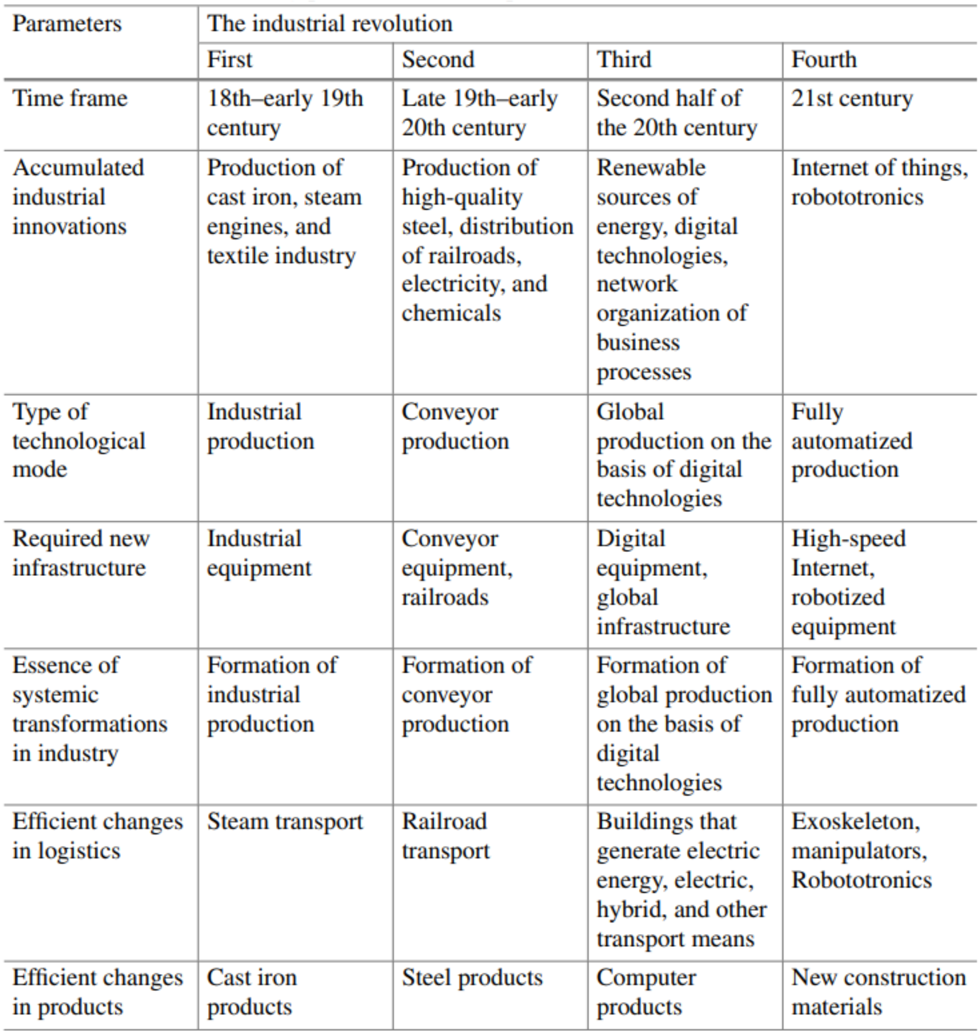
The first Industrial Revolution started in the 18th century, marked by a shift to novel production techniques in Europe, the United States, and globally. This shift encompassed a move from manual production to mechanization, advancements in chemical manufacturing and iron production, a greater reliance on steam and waterpower, the creation of machine tools, and the emergence of an automated factory system [4].

The Second Industrial Revolution, following its predecessor, started in the early 19th century. This period witnessed significant advancements in technology, particularly in steel, chemicals, and electricity, among other areas. A key breakthrough was the invention of electricity, which enabled numerous industries to grow and flourish. This progress also facilitated mineral exploration. A defining feature of the 2IR was the widespread adoption of machinery, predominantly driven by electrical power [5].

The Third Industrial Revolution, also known as the Automation Revolution, began shortly after World War II, around the 1950s. It was characterized by the introduction of partially automated processes using memory-programmable controls and computers. With these technologies, it became possible to fully automate production processes without human intervention. This era was marked by the emergence of semiconductors, mainframe computers, personal computing, and the Internet, leading to a significant technological shift [4].

Industry 4.0, also known as the Digitization Revolution, was initially introduced in 2011 in Germany as a proposal for shaping a new approach to the nation's economic policy, focusing on advanced technological strategies. This industrial revolution is distinguished by the integration of information and communication technologies, primarily in the industrial sector, but also extending to various other societal activities [4].

Figure 1 provides a more detailed description of the 4 industrial revolutions briefly described above.



Parameters	The industrial revolution			
	First	Second	Third	Fourth
Time frame	18th–early 19th century	Late 19th–early 20th century	Second half of the 20th century	21st century
Accumulated industrial innovations	Production of cast iron, steam engines, and textile industry	Production of high-quality steel, distribution of railroads, electricity, and chemicals	Renewable sources of energy, digital technologies, network organization of business processes	Internet of things, robototronics
Type of technological mode	Industrial production	Conveyor production	Global production on the basis of digital technologies	Fully automatized production
Required new infrastructure	Industrial equipment	Conveyor equipment, railroads	Digital equipment, global infrastructure	High-speed Internet, robotized equipment
Essence of systemic transformations in industry	Formation of industrial production	Formation of conveyor production	Formation of global production on the basis of digital technologies	Formation of fully automatized production
Efficient changes in logistics	Steam transport	Railroad transport	Buildings that generate electric energy, electric, hybrid, and other transport means	Exoskeleton, manipulators, Robototronics
Efficient changes in products	Cast iron products	Steel products	Computer products	New construction materials

Figure 1 – The core aspects and principal elements of the first three and the Fourth Industrial Revolution [3, p. 24]

Production systems equipped with digital computer technology are further enhanced with network connectivity and have a digital twin on the Internet. This interconnection of all systems gives rise to "cyber-physical production systems," resulting in smart factories where production systems, components, and individuals interact through a network, allowing for almost autonomous production [5].

The framework of production and manufacturing systems is undergoing a transformation, characterized by the growing use of smart devices and robotics, the Internet of Things (IoT), and advanced data analytics, all of which are enhancing manufacturing intelligence. The convergence of these elements in Industry 4.0 holds the promise of remarkable progress in factory settings. For instance, sensors capable of predicting failures can independently initiate maintenance procedures [4].

2.1.1 Cyber-physical Systems

A Cyber-Physical System (CPS) effectively combines cyber and physical aspects, utilizing advanced sensor, computing, and networking technologies. The widespread acceptance of Cyber-Physical Systems (CPS) is linked to the idea of "Industry 4.0," which revolves around the integration of technologies and knowledge to achieve autonomy, reliability, systematic operation, and control without human involvement. Fundamental technological trends that underpin CPS encompass the Internet of Things (IoT), Big Data, smart technologies, cloud computing, and more. Figure 2 shows the various areas where Cyber-Physical Systems (CPSs) serve as the foundation for development [6].

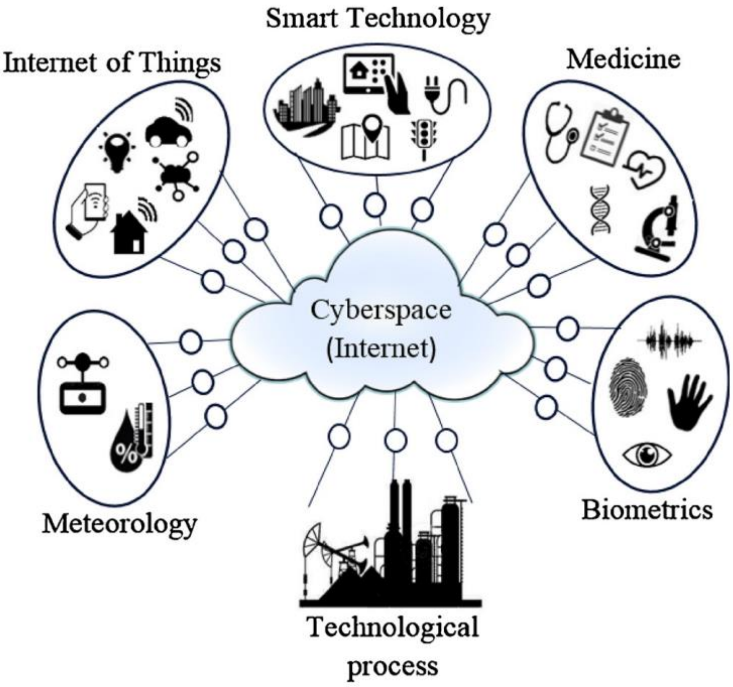


Figure 2 – Cyber-physical systems [6, p. 213]

Key Characteristics of Cyber-Physical Systems (CPSs) according to [7] and [8]:

- Integration of embedded and mobile sensors
- Utilization of sensor data from various domains
- Interaction between cyber and physical elements
- Capability for training and adaptation
- Interconnectivity via the Internet, exemplified by IoT
- Reliable functioning of centralized, automatically controlled systems, such as ATMs and POS systems
- Existence of a unified cyber realm facilitating both internal system exchanges and external interactions, along with information security measures like encryption, firewalls, antivirus software, etc.
- Requirement for dependable operation, with certification in certain cases
- System robustness achieved through automated intelligent control
- Human interaction, either directly internal or external to the system

2.1.2 Cloud Manufacturing

Cloud manufacturing is being recognized as both a new manufacturing paradigm and an integrated technology, showing great potential in evolving current manufacturing practices into future-oriented, service-centric, highly collaborative, and innovative processes. This approach integrates newly developed technologies like the Internet of Things (IoT), Cloud Computing, the Semantic Web, service-oriented technologies, virtualization, and advanced high-performance computing, along with progressive manufacturing models and information technologies [9].

Generally, cloud manufacturing is a system designed to offer both digital and physical manufacturing services, optimizing the use of manufacturing resources. Fundamentally, it needs to connect with actual manufacturing equipment to establish a Cyber-Physical System (CPS). In this context, an integrated CPS specifically for cloud manufacturing is outlined, focusing on enhancing remote access and control of factory machinery like CNC machines and robots. This is achieved by integrating 3D models, sensor data, and camera images in real-time [10].

Figure 3 illustrates the structure of the Cyber-Physical System (CPS), encompassing distributed process planning (DPP), real-time process monitoring, dynamic scheduling of resources, and remote control of devices. DPP utilizes real-time data on machinery and their operational status to facilitate adaptive decision-making in process planning. The Cloud-DPP can also dynamically create machining process plans in response to changes, enabled by informed decision-making [11]. This process involves connecting sensors embedded or attached to each machine with a manufacturing cloud in the cyber workspace. Subsequently, process plans are transmitted as function blocks to the machine controllers on the physical shop floor for implementation. Also, the Cloud-DPP service gathers information from the monitoring service, including details like machine tool ID, its current status, and available time slots, as well as the feature list of a new part that is to be machined [7].

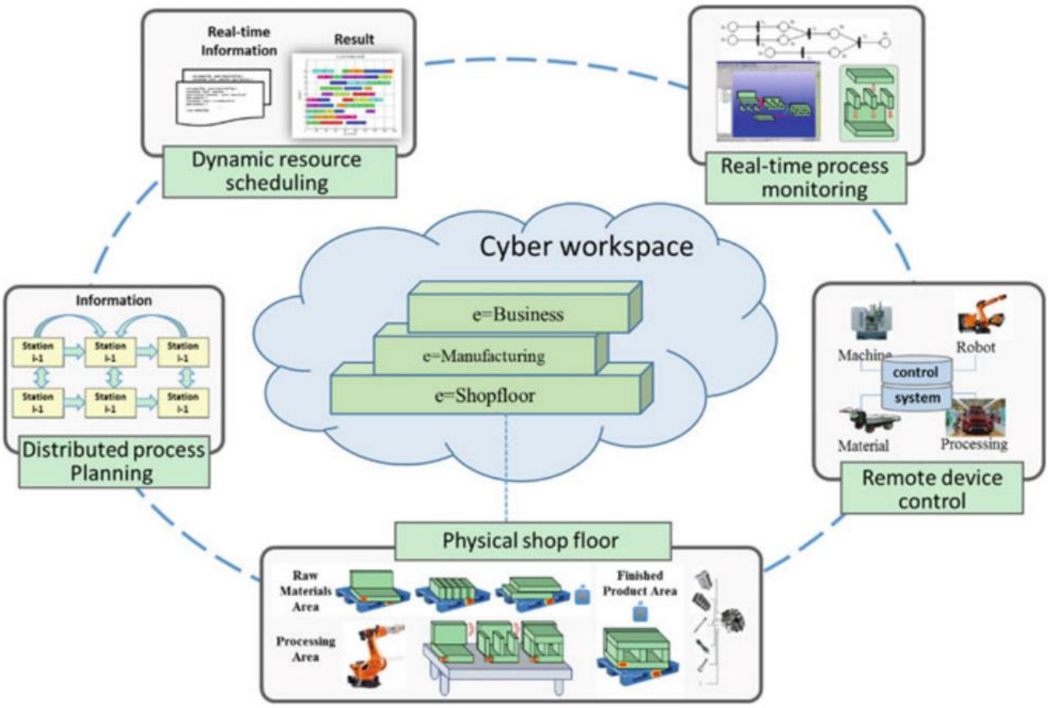


Figure 3 – Cloud-DPP in a cyber-physical system [7, p. 46]

2.1.3 Big Data

As Industry 4.0 and Big Data evolve, there will be a surge in both structured and unstructured data from various stages of the process. Historically, databases for process and product quality have been merged to create predictive models for process monitoring, control, and optimization. Tools like soft sensors and inferential models are typically used in these scenarios. In an interesting way, though, one crucial database has often been neglected by many focused-on process developments: the maintenance department database. This database gathers information on faults from all plant equipment, leading to the plausible theory that equipment failure patterns could be linked to their service conditions. By integrating process and maintenance data, vital insights can be gained on how operational conditions impact system reliability. This integration can provide valuable contributions to process refinement and add a predictive aspect to managing operational risks [13].

Data undoubtedly holds and will maintain a progressively crucial role in both current and future industries, given the consistent growth in the volume and variety of industrial data. The primary sources of industrial big data include:

- Data related to design, encompassing information from product and machinery design.
- Data regarding machine operation, including insights from control systems and equipment functioning.
- Data on staff activities, like manual operation record and videos of staff work processes.
- Information related to costs, including manufacturing and operational expenses.
- Data pertaining to logistics.
- Data about environmental conditions, including weather, indoor temperature, humidity, and noise levels.
- Data for fault detection and monitoring system status.
- Data on product quality, like the defect rates of different facilities.
- Data on product usage, including availability and repair frequencies.
- Customer-related information, such as customer details, feedback, and suggestions [12].

2.1.4 Internet of Things (IoT)

IoT is utilized across different sectors such as automotive, healthcare, manufacturing, residential, and advanced electronics, enhancing the intelligence of products, services, and processes. This technology has grown out of the rapid development of wireless technologies, sensors, and the internet. IoT links networked systems and various devices via the internet, making these systems environmentally responsive and insightful through sensors, enabling them to transmit vast quantities of data daily. These interconnected systems are user-friendly and communicate seamlessly through the internet [14].

Industry 4.0 allows for the merging of physical assets into a seamless blend of digital and physical operations, leading to the development of smart factories and intelligent manufacturing settings. The Internet of Things (IoT) is a rapidly expanding technology playing a significant role in the advent of Industry 4.0. IoT aims to infiltrate our daily surroundings and objects, bridging the gap between the physical and digital realms. It facilitates constant connectivity of people and objects, regardless of location, using various networks and services. This connectivity ideally happens anytime, with anything, and with anyone [15].

2.1.5 Sensors

A sensor functions as a unit that perceives and reacts to various forms of input from the physical environment. This input might include elements such as light, temperature, movement, humidity, or pressure, among other aspects of the environment. Typically, the sensor's output is a signal that is transformed into a format understandable by humans, either displayed directly at the sensor's site or sent digitally across a network for monitoring or additional analysis [16].

Sensors are crucial in the realm of the Internet of Things (IoT), enabling the creation of systems that gather and process information about particular environments. This facilitates more straightforward and efficient monitoring, management, and control. IoT sensors find applications in various settings, including homes, outdoor areas, automobiles, airplanes, industrial environments, and more. They serve as a link between the physical and digital realms, functioning as the perceptive organs for a computing system that interprets and responds to the data acquired from these sensors [16].

Figure 4 illustrates the diverse abilities and potentials of sensors within the Industry 4.0 framework. It delves into the standard characteristics of these sensors, including predictive maintenance, building automation, monitoring or conditioning of assets, and the comprehensive automation of processes, all tailored for the Industry 4.0 environment. Furthermore, it discloses subcategories of these capabilities, such as monitoring trends, offering efficient services, supporting cloud technologies, ensuring precision and miniaturization, optimizing processes, and reducing or eliminating downtime. These aspects further underscore and validate the proficiency of sensors in the specialized sector of Industry 4.0 [17].

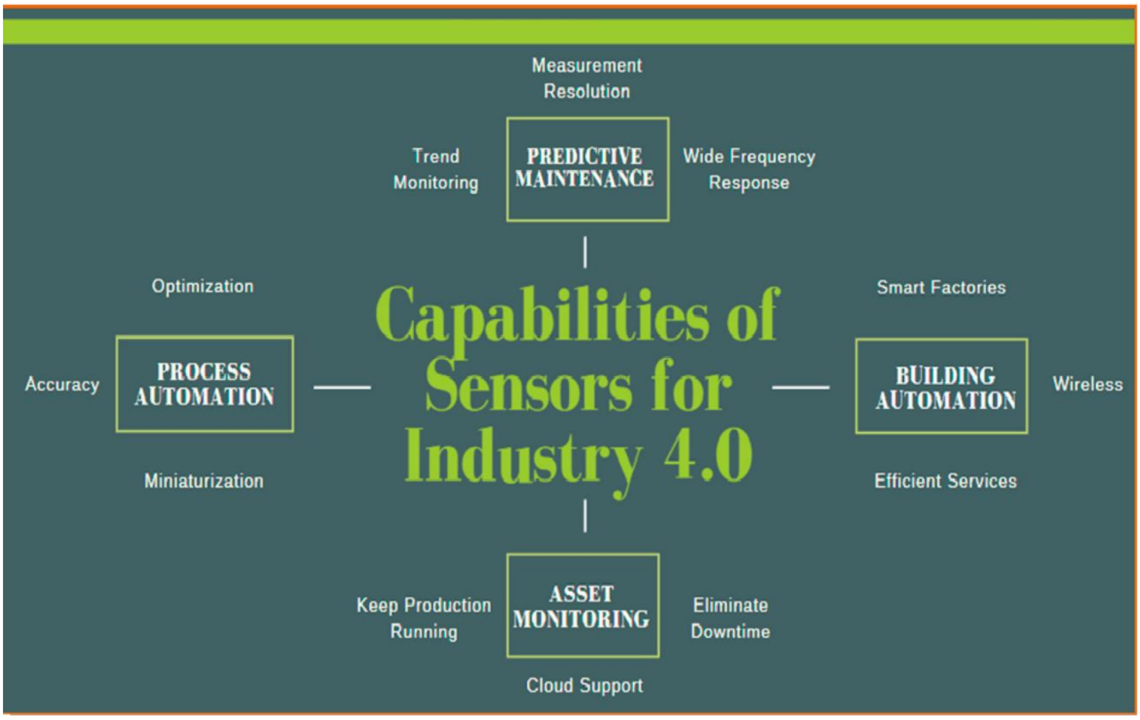


Figure 4 – Several capabilities of sensors for industry 4.0 domain [17, p. 5]

2.2 Reliability Engineering in Industry 4.0

Reliability stands as the best quantitative assessment of the design of products, components, or systems, representing the quality exhibited by parts, elements, or systems over time as they function without encountering failures in a particular environment within a defined duration [19].

Reliability engineering constitutes an engineering field that employs scientific principles to guarantee the optimal performance of a component, product, facility, or procedure, ensuring uninterrupted function within a specified environment for the necessary duration without encountering failures. It prioritizes trustworthiness throughout a product's lifecycle, defined as the capability of a system or component to operate under specified conditions for a predetermined timeframe. Put differently, reliability encompasses two crucial aspects: time and stress [20].

Reliability has not only spurred research aimed at emphasizing its significance to decision-makers but has also been a key motivator. Owing to the surge in computational capabilities and the vast data produced through technological progress, the topic of reliability has led to investigations where advanced computational models are employed to enhance the prediction of equipment failures [18].

Reliability predicts analysis, evaluates, and enhances various products and device systems. Within the context of Industry 4.0, this presents a chance to enhance efficiency and bolster productivity across connections, devices, product lines, and data transmission. This is due to the fact that reliability methodologies cater to the optimal functioning of devices such as hardware, software, and connectivity [19].

The maintenance strategies linked to reliability must continually evolve to keep pace with the technological advancements inherent in products and manufacturing equipment. Alongside these technological advancements, maintaining high quality in manufacturing is essential for consistently producing reliable products. Proactive assurance of product reliability remains a vital, routine aspect of production. In this context, Industry 4.0 offers an advantageous environment for the development and enhancement of reliability models [18].

Industry 4.0 introduces a fresh opportunity for visionary thinkers and engineers to design novel systems and pioneer smart devices and tools. These innovations not only foster infrastructures for industrial progress but also introduce unforeseen failure mechanisms, unfamiliar economic, functional, technical, and structural interdependencies among system elements, consequently leading to unfamiliar hazards and risks. However, amidst the surge in complexity and interdependence, the integration of novel concepts and the advancements in knowledge, methodologies, and technologies—such as big data, the internet of things, and swift adaptability to changes—create new prospects for refining reliability engineering techniques and augmenting the capability to predict reliability [21].

3 Maintenance Strategies

Within industry, maintenance stands as a crucial element impacting both the operational uptime and efficiency of machines. Hence, it is imperative to promptly recognize and rectify machine failures to prevent any interruptions in production [22].

Because of advancements in machine learning methods and sensor technology, the favored approach for enhancing machine and process maintenance in industrial settings nowadays revolves around a data-centric viewpoint [24].

Efficient utilization of existing resources and the avoidance of unnecessary expenses are essential within maintenance strategies. These measures are crucial to maintaining overall system efficiency and keeping maintenance costs minimized [25]. Figure 5 provides a comprehensive summary of various maintenance strategies.

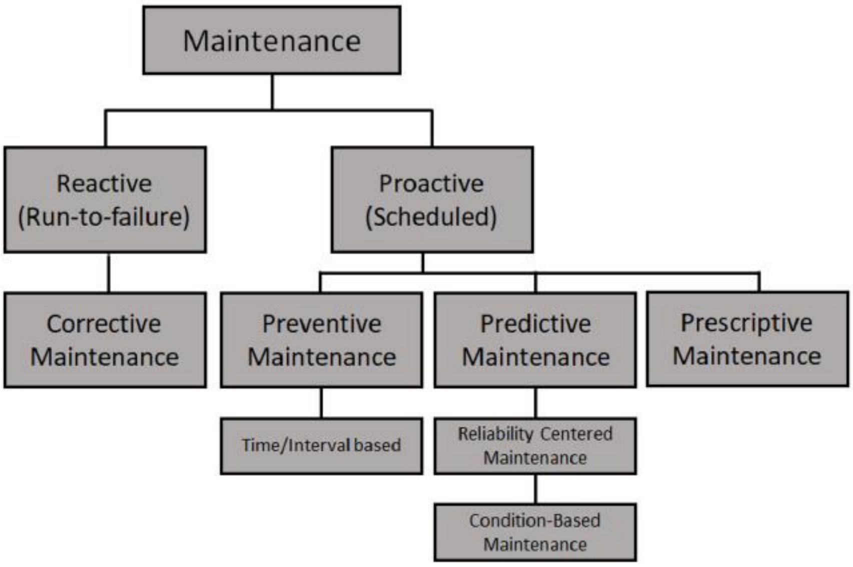


Figure 5– Different maintenance strategies [23, p. 3]

The most important Maintenance strategies are briefly explained below:

- **Corrective Maintenance**: As per the EN 13306 standard, corrective maintenance refers to maintenance conducted subsequent to identifying a fault, aiming to restore an item to a condition where it can fulfill its necessary function [26].

- **Preventive Maintenance**: According to the EN 13306 standard, preventive maintenance is described as maintenance performed at scheduled intervals or based on specified criteria, aiming to minimize the likelihood of failure or the decline in the functionality of an item. Hence, preventive maintenance encompasses a series of steps taken proactively to forestall failures or the deterioration of a machine. Time-based maintenance, within this context, is identified as the preventive maintenance method advising the execution of all maintenance tasks either after a specific number of operational hours or based on predetermined schedules, without regard to the item's current health condition [26].

- **Predictive Maintenance**: As outlined in the EN 13306 standard, predictive maintenance involves condition-based upkeep conducted after a forecast generated from repetitive analysis or recognized attributes, evaluating crucial parameters related to the item's degradation. It employs diverse methods and machine learning techniques to examine both recent and past data, developing predictive models aimed at making precise projections regarding the forthcoming condition of a machine or equipment [26].

- **Condition-based Maintenance**: The EN 13306 standard defines condition-based maintenance as a form of preventive maintenance that incorporates a blend of condition monitoring, inspections, testing, analysis, and subsequent maintenance actions. Its primary objective is to forecast a maintenance task based on evidence of degradation and deviations from an expected normal behavior of an asset. Multiple sensors are employed to monitor equipment, capturing pertinent data regarding its operational lifespan. Furthermore, contextual factors such as temperature, humidity, etc., also contribute crucial information. Typically, Key Process Indicators (KPIs) or health indicators are computed and assessed to identify patterns indicating abnormal situations and potential failure occurrences [26].

- **Prescriptive Maintenance**: In terminology, the maintenance standards EN 13306 and DIN EN 31051 do not explicitly reference it. However, its operational aspects can be inferred, representing a suggestion for specific actions derived from models related to corrective and predictive maintenance outcomes. The primary obstacle in implementing prescriptive maintenance lies in the practical complexity of constructing operational models. Current research models rely on a makeshift approach to model development, combining machine learning methods and data fusion techniques with fuzzy reasoning, simulation methods, and evolutionary algorithms [26].

The most effective maintenance strategy aims to enhance equipment health, lower failure rates, and cut maintenance expenses while extending equipment longevity. That's why predictive maintenance stands out as the foremost strategy among others, especially in the era of Industry 4.0. This approach is gaining significant traction due to its ability to predict and prevent issues, utilizing advanced technologies and data analysis to streamline maintenance processes and optimize equipment lifespan [22].

3.1 Predictive Maintenance

Predictive Maintenance (PdM) relies on predictive mechanisms to ascertain the timing of necessary maintenance actions. This method involves ongoing monitoring of machine or process integrity, enabling maintenance to be executed solely when deemed necessary. Additionally, it facilitates early detection of failures through predictive tools leveraging historical data (such as machine learning techniques), integrity indicators (such as visual cues, wear, color variations differing from the original, among others), statistical inference techniques, and engineering methodologies [27]. Within the realm of the Internet of Things (IoT) and Industry 4.0, there's a growing potential to merge predictive maintenance with other systems within the production process. This integration offers expanded possibilities to interconnect predictive maintenance with various facets of production systems [22].

The comprehensive framework for predictive maintenance comprises 10 categories. Figure 6 displays these 10 categories, representing the topmost level of this framework [27].

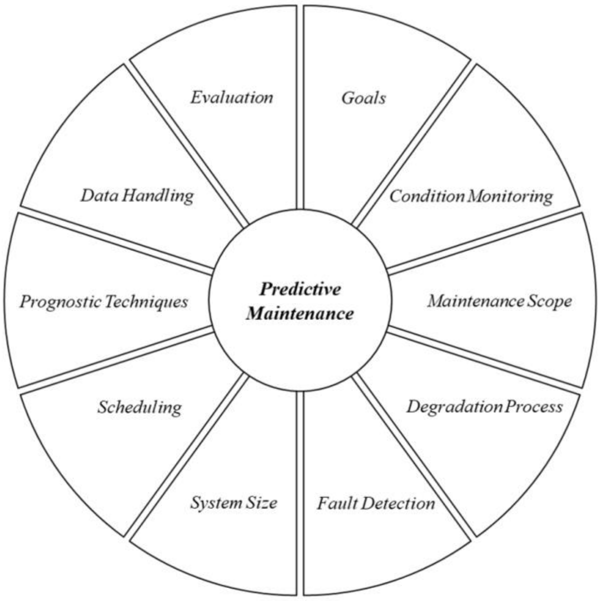


Figure 6– Categories of the Framework for Predictive Maintenance [27, p. 5]

Predictive maintenance offers several benefits. Firstly, it diminishes the need for unnecessary maintenance tasks that adhere to fixed periodic intervals, potentially decreasing the total number of maintenance activities throughout a machine's lifespan. Secondly, it not only helps avoid premature maintenance but also averts late maintenance actions, as equipment failures might occur before the scheduled periodic maintenance interval. This is because these intervals are typically based on average lifespans that might encompass significant deviations from the norm, such as specific structural features within machinery components. Both the reduction in unnecessary maintenance and the prevention of critical breakdowns lead to enhanced productivity and reduced production downtime. Consequently, depending on the accuracy of the predictive method employed, predictive maintenance can be viewed as an overall enhancement in efficiency compared to traditional maintenance approaches [22].

3.2 Condition-Based Monitoring

The aim of condition-based maintenance is to conduct repairs or replacements proactively, preventing part failures to minimize breakdowns and ensuring that maintenance occurs only when required. This approach seeks to strike a balance between inspection expenses and the costs resulting from failures. In opposition to conventional maintenance practices reliant on system age or breakdown occurrences, traditional preventive maintenance relies on routine time-based inspections and maintenance activities determined through the analysis of failure time data obtained from a series of degradation experiments. Conversely, condition-based maintenance utilizes measurements of a degradation variable gathered during product use to schedule maintenance tasks [28].

Condition monitoring technology is categorized into two main types: offline monitoring and online monitoring. Offline monitoring involves periodic assessments conducted at specific intervals using machine sensor data. These analyses are performed in a separate laboratory setting, away from the machine itself. On the other hand, online condition monitoring involves sensors within the machine continually gathering data, which is then compared to predefined acceptable values in real-time. In recent times, online condition monitoring has gained significant traction, primarily attributed to the advancements in Internet of Things (IoT) solutions [30].

A crucial aspect of successful condition-based maintenance (CM) lies in its ability to pinpoint potential failure indicators, serving as the foundation for conducting reliability analysis, estimating lifespan, and predicting the remaining useful life (RUL) [29].

3.3 Methods for data-driven predictive maintenance models

In the last few years, industrial wireless sensor networks and industrial cyber-physical systems have emerged as crucial technologies for gathering data in intricate industrial settings. These systems enable the collection of mechanical data through diverse, highly reliable sensors in real-time. Clearly, due to the ongoing enhancements in data collection capabilities and the exponential surge in data volume, data-driven approaches for monitoring health have made significant strides. This progress has garnered considerable attention, particularly concerning predictive maintenance for industrial equipment [31].

Due to the significant resurgence of artificial intelligence, data-driven methods for predictive maintenance (PdM) have emerged as the most efficient way for tackling smart manufacturing and vast industrial data. This is particularly true when it comes to tasks like health assessment, such as fault diagnosis and estimating remaining life [31].

The predictive maintenance (PdM) techniques are primarily categorized into three groups: Prognosis based on models; knowledge; and driven by data. In particular, a thorough investigation into PdM methods for mechanical equipment was conducted, focusing on aspects such as data collection, data analysis, and decision-making support. An intelligent PdM system driven by data was suggested with the aim of achieving flawless manufacturing [31].

Data-driven models use various methods. These methods are divided into statistical methods of predictive Maintenance and machine learning methods of predictive Maintenance. These methods are explained in detail below.

3.3.1 Statistical methods of predictive Maintenance

Statistical modeling involves the complex process of creating sample data and making predictions about the real world by employing various statistical models and specific assumptions. Within this procedure, there is a mathematical connection established between variables that exhibit randomness and those that do not [32].

Typical data sources used for statistical analysis encompass a variety of sources such as Internet of Things (IoT) sensors, census records, public health statistics, social media information, imagery datasets, and other data from the public sector. These diverse datasets are particularly advantageous for generating real-world predictions [33].

The following significant classification methods such as Decision Tree, Linear Discriminant Analysis, Logistic Regression and K-Nearest Neighbors are being reviewed.

3.3.1.1 Decision Tree

A decision tree is a non-parametric supervised learning technique employed for tasks involving both classification and regression. Its structure is hierarchical, resembling a tree, comprising a root node, branches, internal nodes, and leaf nodes [34].

Decision tree classifiers are recognized as one of the most popular methods for representing classifiers in data classification. Researchers from diverse fields and backgrounds have explored the challenge of expanding decision trees using existing data, including machine learning, pattern recognition, and statistics. Decision tree classifiers have been suggested for application in numerous domains, including medical disease analysis, text classification, user smartphone classification, image analysis, and various other fields, showcasing their versatility and potential usefulness in different contexts [35].

The process of the decision tree is illustrated in Figure 7.

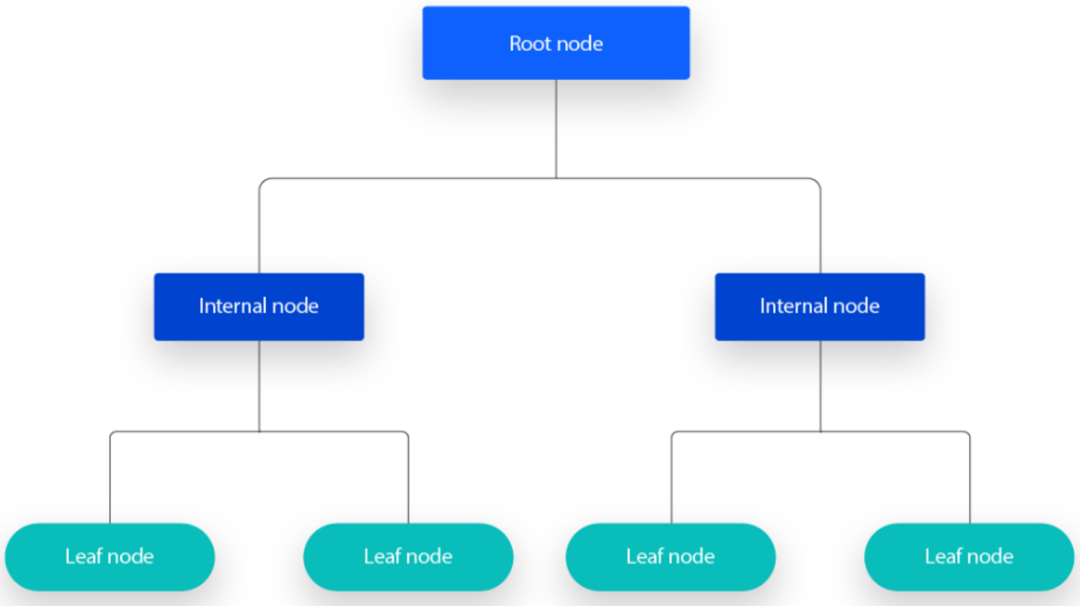


Figure 7– The process of the decision tree [34]

3.3.1.2 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a supervised machine learning method employed for solving multi-class classification problems and is widely utilized in data classification and reducing dimensions. It adeptly manages scenarios where the frequencies within classes differ, and its efficacy has been evaluated using randomly generated test data. This technique aims to optimize the ratio between the variance among classes and within classes within a dataset, ensuring maximum distinguishability. Its application in speech recognition involves employing Linear Discriminant Analysis for solving classification issues [39].

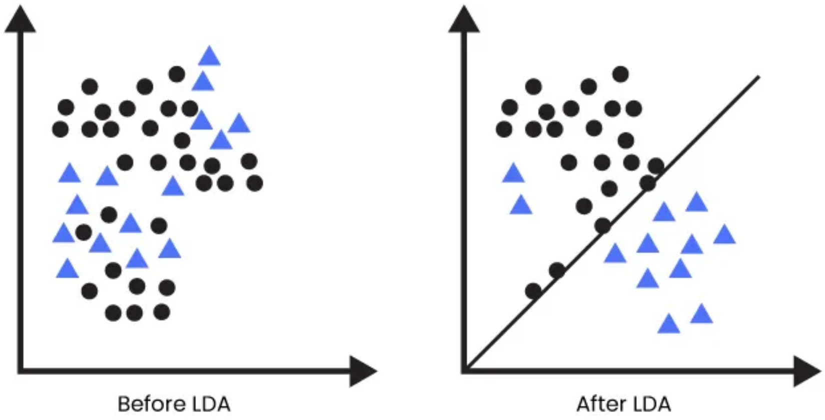


Figure 8– An example for classification before and after implementing to LDA [40]

Figure 8 displays the classification of data for both pre- and post-LDA implementation. Throughout this process, the following steps were sequentially applied [40]:

- Data Preparation
- Compute Class Statistics
- Compute Between-Class and Within-Class Scatter Matrices
- Compute Eigenvectors and Eigenvalues
- Select Discriminant Directions
- Transform Data
- Classification

3.3.1.3 Logistic Regression

Logistic regression models are statistical models that explain the connection between a qualitative dependent variable (one that assumes certain discrete values) and an independent variable. These models are utilized to analyze the impact of predictor variables on categorical outcomes, typically with a binary outcome, making it a binary logistic model. If there's only one predictor variable, it's termed a simple logistic regression. When there are involved multiple predictors, including categorical and continuous variables as predictors, it's termed a multiple or multivariable logistic regression [36].

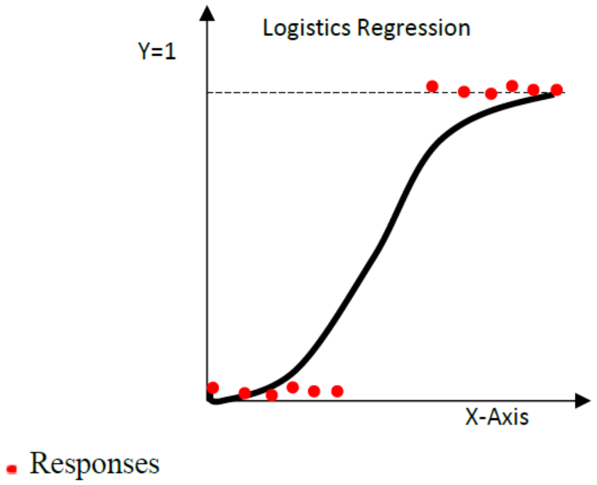


Figure 9– Example of logistic regression [38, pp.291]

As seen in Figure 9 that the outcome in logistic regression represents a probability, the dependent variable is constrained within the range of 0 to 1. To model this, logistic regression employs a logit transformation on the odds, which is the ratio of the probability of success to the probability of failure [37].

3.3.1.4 K-Nearest-Neighbour

The K-Nearest Neighbor (KNN) technique is utilized for object classification by referencing the closest learning data points concerning the object under consideration, comparing them based on past and present data. During the learning phase, KNN computes the proximity of the nearest neighbor using the Euclidean distance formula.

Alternative methods involve optimizing the distance formula by comparing it with similar approaches to achieve superior results [42].

Figure 10 shows how two different classes are separated on a simple data set with the K-Nearest-Neighbor Method.

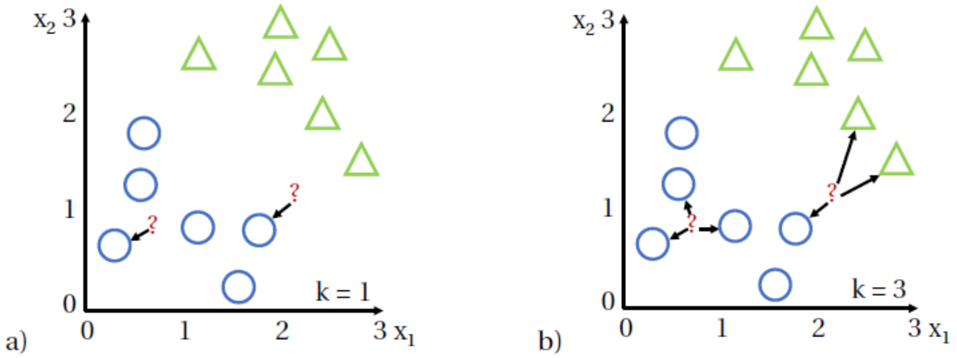


Figure 10– K-Nearest-Neighbor Method for a) k= 1 and b) k = 3 on a simple data set with two classes shown as circles and triangles. The “?” symbols represent the data points to be classified. [41, pp. 101]

In addition to the choice of k, there are a variety of definitions for the distance between two data points that are required to determine the closest center point. In the example before, the Euclidean (geometric) distance was used, which follows from the sentence of the Pythagoras. For d properties (= dimensions) the distance is d (p, q) in Equation 1 of two points P and Q [41].

$$d_{(p,q)} = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_d - q_d)^2} \tag{1}$$

3.3.2 Machine Learning methods of predictive Maintenance

Predictive maintenance involves forecasting potential malfunctions by leveraging data obtained from monitoring equipment and performance measurements of processes. Machine learning algorithms are frequently employed to scrutinize this equipment monitoring data. Machine learning entails the computer's ability to operate more accurately through data collection and analysis. Typically, machine learning algorithms utilize supervised learning, where labeled data is employed to train the algorithm. Nevertheless, numerous supervised machine learning algorithms exist, making the selection of the optimal one for addressing predictive maintenance challenges a non-trivial task [43].

Within industries, maintaining equipment holds significant importance, directly impacting equipment uptime and efficiency. Detecting and resolving equipment faults is crucial to prevent disruptions in production processes. Machine Learning (ML) methods have emerged as promising tools in Predictive Maintenance (PdM) applications, aiming to avert failures in the machinery constituting production lines on the factory floor [22].

The following significant classification methods such as Support Vector Machine, Deep Learning, Random Forest and Artificial Neural Network and Deep Neural Network are being reviewed.

3.3.2.1 Support Vector Machine

A Support Vector Machine (SVM) represents a supervised learning technique utilized in machine learning for addressing classification and regression tasks. SVMs are good in resolving binary classification challenges, where the objective involves categorizing elements within a dataset into two distinct groups. The fundamental goal of an SVM algorithm revolves around identifying the optimal line, referred to as a decision boundary, that effectively segregates data points belonging to different classes. In higher-dimensional feature spaces, this boundary is termed a hyperplane. The primary aim is to maximize the margin, denoting the space between the hyperplane and the nearest data points from each category, thereby facilitating clear differentiation between data classes. SVMs prove valuable for analyzing intricate data sets that cannot be delineated by a simple linear boundary. Nonlinear SVMs achieve this by leveraging a mathematical technique, transforming the data into higher dimensions, where discerning a boundary becomes more feasible [44].

In Figure 11, there exists a depiction of two distinct categories, delineated and separated by a decision boundary or hyperplane.

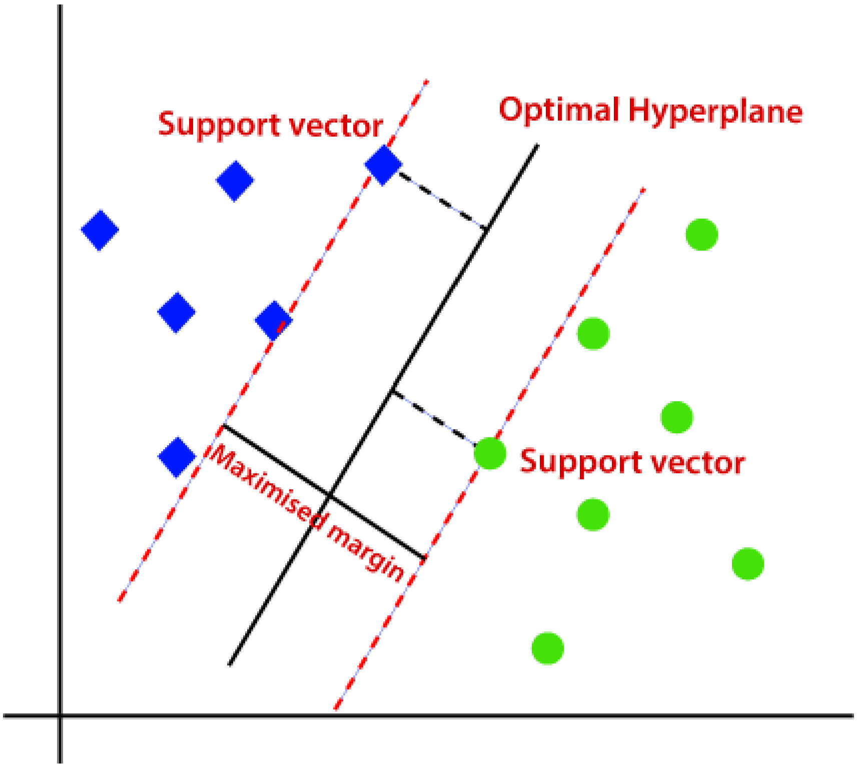


Figure 11– An example SVM model with two categories for classification [45]

3.3.2.2 Random Forest

Random Forest stands as a well-known machine learning algorithm within the realm of supervised learning. This versatile method finds application in addressing both Classification and Regression tasks in machine learning. Its foundation lies in ensemble learning, a methodology that involves amalgamating multiple classifiers to address intricate problems and enhance the model's performance [46].

The Random Forest classifier, as its name implies, comprises multiple decision trees generated on diverse subsets of the provided dataset. It leverages averaging across these trees to enhance the predictive accuracy of the dataset. Unlike relying on a single decision tree, this method incorporates predictions from each tree and determines the final output based on the majority consensus among these predictions. Increasing the number of trees in the forest contributes to heightened accuracy and serves as a preventive measure against overfitting issues [46].

In Figure 12 provided the working of the Random Forest algorithm:

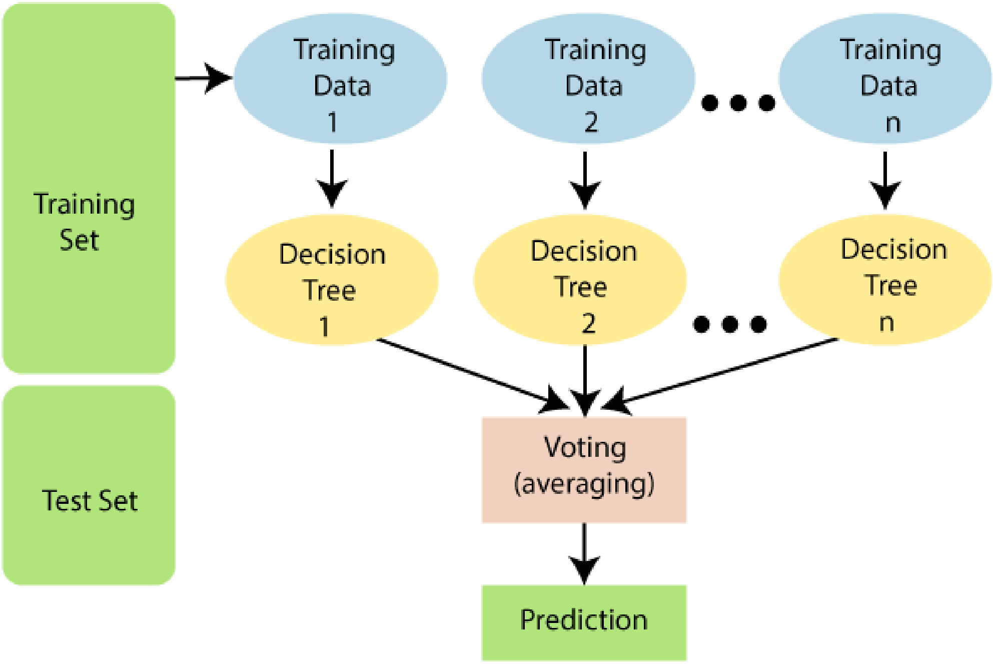


Figure 12– Random Forest algorithm [46]

3.3.2.3 Artificial Neural Network and Deep Neural Network

Artificial Neural Networks (ANNs), a subset of machine learning, lie at the core of deep learning algorithms. Inspired by the human brain, they replicate the signaling process among biological neurons. As can be seen in Figure 13, ANNs consist of node layers: an input layer, one or more hidden layers, and an output layer. Each artificial neuron within these layers connects to others, possessing associated weights and thresholds. When the output of a node surpasses the specified threshold, it activates, forwarding data to the subsequent network layer; otherwise, no data transmission occurs. ANNs have shown competitive utility compared to traditional regression and statistical models [47].

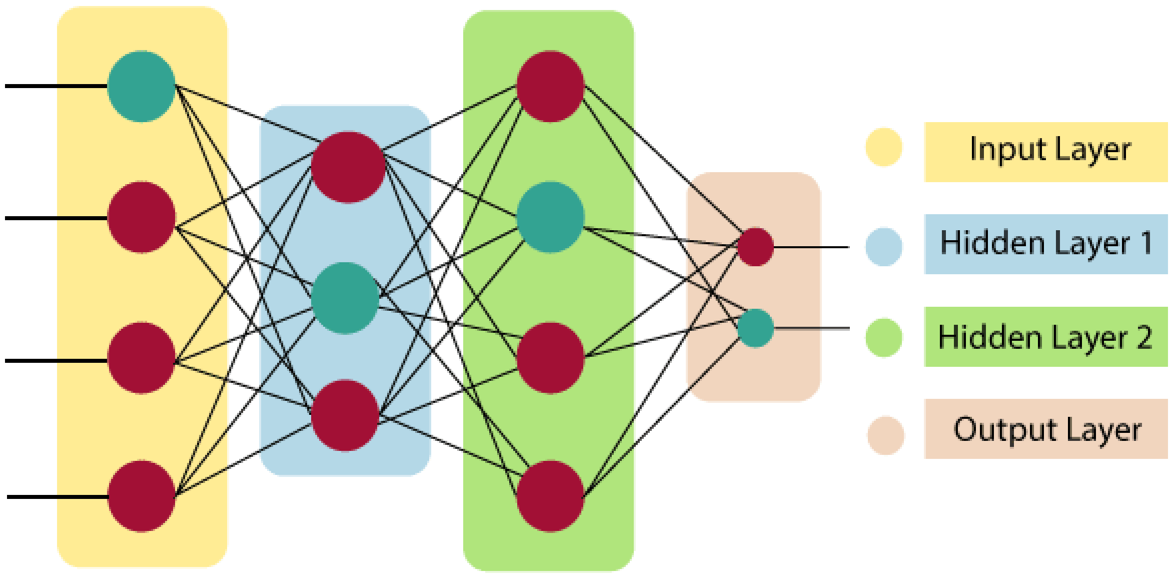


Figure 13– Artificial Neural Network [48]

A Deep Neural Network (DNN) is an advanced form of a neural network characterized by numerous hidden layers. In a DNN, these layers are densely interconnected. DNNs excel in extracting adaptable fault features, effectively representing crucial information, and conducting intelligent diagnoses. Their capability to enhance detection accuracy is notable, and they are highly efficient in reducing errors that stem from manually designed features [31].

3.4 Current applications of ML in predictive maintenance and reliability engineering

Table 1 provides an overview of the current applications of machine learning (ML) in predictive maintenance and reliability engineering. Additionally, the advantages and disadvantages of some ML methods to be used in this paper are also stated.

Table 1 – The advantages, disadvantages, and applications of ML models in predictive maintenance and reliability engineering [49]

ML model	Reported advantage	Model disadvantage	Application
DT	1. Excellent accuracy with great training efficiency	1.Not robust to data noise	1. Assessing stakeholders corporate governance 2. Dirt and mud detection on a wind turbine blade 3. Forest fire risk assessment
LDA	1.Excellent prediction accuracy	1. Not suitable for nonlinear applications	1. Dirt and mud detection on a wind turbine blade
KNN	1. Excellent accuracy and efficiency	1. Not suitable for high dimensional data 2. Not robust to data noise	1. Risk-based inspection screening assessment 2. Dirt and mud detection on a wind turbine blade
SVM	1. Excellent accuracy and efficiency for feature extraction 2. Accurate in detecting the early signs of system anomalies	1. Not suitable for sparse and high dimensional data 2. Requires prior knowledge for kernel selection	1. Real-time Motor machine failure identification and early fault diagnosis 2. Spacecraft health monitoring
RF	1. Suitable for discrete classification 2. Excellent estimation accuracy	1. More complex than DT model 2. Hard to interpret the 'black box' model	1. Rank the importance of each component of an engineering system
DNN	1. Excellent prediction accuracy and training efficiency 2. Excellent long- and mid-term prediction accuracy	1. Computationally expensive 2. Hard to interpret the 'black box' model	1. RUL of aircraft engine prediction 2. Human errors prediction 3. Component reliability and degradation level prediction

4 Application Example

4.1 Source of data

This paper utilizes a dataset obtained from the open website Kaggle [50], specifically designed for the classification of synthetic milling processes. The synthetic predictive maintenance dataset was generated due to the challenges in acquiring and publishing real predictive maintenance datasets, which are often hard to obtain. The dataset is crafted to closely mirror the real predictive maintenance scenarios encountered in the industry, aligning with the scope of knowledge and experience [50].

The dataset encompasses 10,000 data points with 14 columns, each column representing the features of a data point. The features align with the specifications outlined in references [50]:

- UID: Unique identifier spanning from 1 to 10,000.
- Product ID: Comprising a letter designation (L for Low, M for Medium, and H for High) to represent product quality variants, with low accounting for 50% of all products, medium for 30%, and high for 20%. Additionally, each variant is associated with a variant-specific serial number.
- Type: The product type, denoted as L, M, or H, retrieved from column 2.
- Air temperature [K]: Generated through a random walk process and subsequently normalized to a standard deviation of 2 K around 300 K.
- Process temperature [K]: Generated using a random walk process normalized to a standard deviation of 1 K, added to the air temperature plus 10 K.
- Rotational speed [rpm]: Calculated through a power of 2860 W, with the addition of normally distributed noise.
- Torque [Nm]: Torque values follow a normal distribution centered around 40 Nm, with a standard deviation of 10 Nm and no negative values.
- Tool wear [min]: The quality variants High/Medium/Low contribute 5/3/2 minutes of tool wear to the utilized tool during the process.
- Machine failure: Specifies whether the machine has encountered failure in this specific data point, indicating the presence of any of the following true failure modes.

The machine failure consists of five independent failure modes:

- Tool wear failure (TWF): The tool replacement or failure occurs at a randomly chosen tool wear time within the range of 200 to 240 minutes (120 instances in our dataset). At this specific time interval, the tool has been replaced 69 times, and failure has occurred 51 times, randomly assigned.
- Heat dissipation failure (HDF): Process failure due to heat dissipation occurs when the difference between air and process temperature is below 8.6 K, and the tool's rotational speed is under 1380 rpm. This condition is met in 115 data points.
- Power failure (PWF): The product of torque and rotational speed (in rad/s) corresponds to the required power for the process. If this power falls below 3500 W or exceeds 9000 W, the process fails. This situation is observed in 95 instances within our dataset.
- Overstrain failure (OSF): Process failure occurs due to overstrain if the product of tool wear and torque surpasses 11,000 minNm for the Low (12,000 for Medium, 13,000 for High) product variants. This condition is met in 98 data points.
- Random failures (RNF): Each process carries a 0.1% probability of failure, irrespective of its process parameters. However, this scenario is observed in only 5 data points, which is less than expected for a dataset containing 10,000 data points.

If any of the failure modes is true, the process is considered to fail, and the "Machine failure" label is assigned a value of 1. Consequently, the specific failure mode that led to the process failure is not discernible to the machine learning method.

4.2 Programming Language: R and Preparation of the data set

Jun.-Prof. Dr. Antoine Tordeux, the Chair of Traffic Safety and Reliability at the University of Wuppertal, authored the analysis script specifically for this paper and provided it for use. The script comprises correlation analysis, component analysis, and classification using diverse machine learning approaches.

RStudio program was executed to analyze this data set. The "R" programming language is a scripting language designed for statistical data manipulation and analysis. Developed by Ross Ihaka and Robert Gentleman at the University of Auckland's Department of Statistics, R is open-source software provided free of charge. It is widely employed for statistical calculations, including linear and nonlinear modeling, classical statistical tests, time series analysis, classification, and clustering. Presently, R is compatible with nearly every standard computing platform and operating system, and reports indicate successful execution on modern tablets, phones, PDAs, and game consoles. Notably, R boasts sophisticated graphics capabilities, offering precise control over various aspects of plots or graphs through its base graphics system. Beyond its role as a statistics package, R also accommodates machine learning and artificial intelligence methods. Its functionality can be expanded through packages, facilitating the incorporation of additional features and capabilities [51].

To conduct the required analyses, it is essential to install a few packages initially (see Appendix A). These packages are neuralnet, MASS, corrplot, rpart, rpart.plot, e1071, randomForest, LPCM and GGally respectively. The relevant packages and meanings are detailed in Table 2.

Table 2 – R Packages and Meanings [52]

R Package	**Meanings**
neuralnet	Training of Neural Networks
MASS	Support Functions and Datasets for Venables and Ripley's MASS
corrplot	Correlation Matrix
rpart	Recursive Partitioning and Regression Trees
e1071	Support Vector Machine
randomForest	Random Forest
LPCM	Local Principal Curve Methods
GGally	Reducing the complexity of combining geoms with transformed data
Rpart.plot	Plot 'rpart' Models: An Enhanced Version of 'plot.rpart'

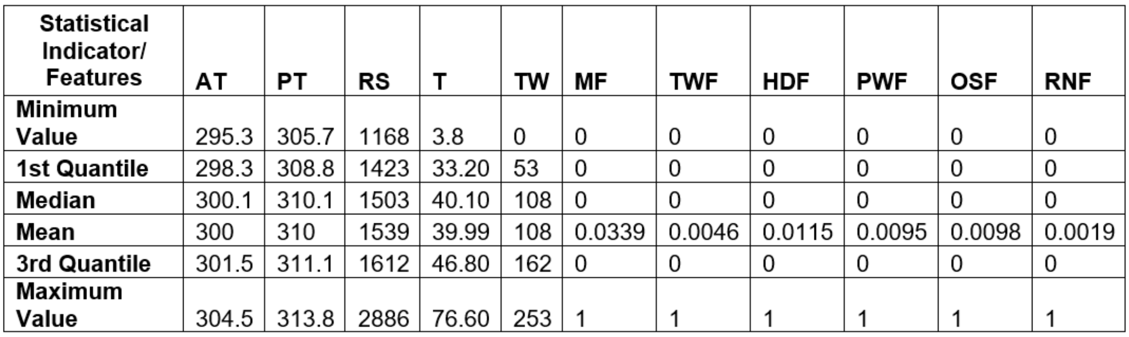
After the installation of the packages, data recording becomes possible for subsequent reading (see Appendix A). It's crucial to note that the file intended for reading should be located in the specified directory. Once the dataset is read, pertinent columns are stored as both a data frame and a variable. Certain sensor data is omitted from consideration, as factors like ID number or product type are irrelevant for classification. The relevant measurement variables are then allocated to a separate data frame, encompassing parameters such as "air temperature," "process temperature," "rotational speed," "torque," and "tool wear." Machine operational states are linked to variables: machine failure, TWF, HDF, PWF, OSF, and RNF. These variables assume a value of 0 for the operating state and a value of 1 for the failed state, facilitating the classification process.

4.3 Data Analysis

4.3.1 Correlation analysis and principal component analysis

Initially, the dataset underwent analysis, presenting the Minimum and Maximum values, 1st and 3rd Quantiles, Median, and Mean values for all variables in Table 3.

Table 3 – Statistical Indicators



Statistical Indicator/ Features	AT	PT	RS	T	TW	MF	TWF	HDF	PWF	OSF	RNF
Minimum Value	295.3	305.7	1168	3.8	0	0	0	0	0	0	0
1st Quantile	298.3	308.8	1423	33.20	53	0	0	0	0	0	0
Median	300.1	310.1	1503	40.10	108	0	0	0	0	0	0
Mean	300	310	1539	39.99	108	0.0339	0.0046	0.0115	0.0095	0.0098	0.0019
3rd Quantile	301.5	311.1	1612	46.80	162	0	0	0	0	0	0
Maximum Value	304.5	313.8	2886	76.60	253	1	1	1	1	1	1

Correlation analysis is essential for examining the connection between variables. In order to explore the relationship between variables, a correlation analysis is conducted (see Appendix B). Correlation, also referred to as correlation analysis, involves examining the association or connection between two or more quantitative variables, primarily based on the assumption of a linear relationship. Analogous to measures of association for binary variables, correlation evaluates the "strength" or "extent" of the association between variables and its direction. The outcome of this analysis is expressed through a correlation coefficient, ranging from -1 to +1. A correlation coefficient of +1 indicates a perfect positive (linear) relationship, -1 suggests a perfect negative (linear) relationship, and zero signifies no linear relationship between the two variables being studied [53]. In this case, the calculation of the correlation is suitable for examining the dependence of the six variables. The two-dimensional correlation graph visually illustrates the direction and strength of the linear relationship among these six variables, and the correlation matrix is presented in Figure 14.

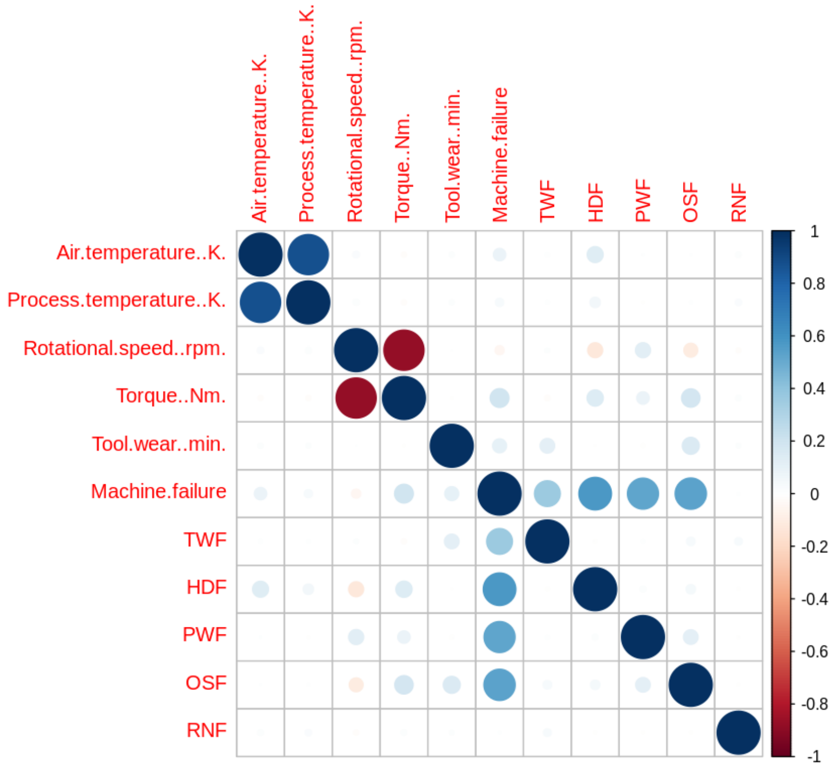


Figure 14– Correlation matrix of the variables

The positive correlations are represented by blue-colored dots, while the negative correlations are depicted by red-colored dots. The intensity of the dot's color corresponds to the strength of the correlation; darker dots indicate stronger correlations. Conversely, lighter-colored dots suggest correlation coefficients close to zero, indicating weak correlations between variables. For instance, in the illustration, it can be inferred that "Air Temperature" and "Process Temperature" exhibit a linear relationship, whereas "Rotational Speed" and "Torque" display a negative correlation. The machine failure states HDF and OSF demonstrate mild linear relationships with variables, excluding "Rotational Speed," and exhibit very slight negative relationships overall.

Following the correlation analysis, a principal component analysis was conducted (see Appendix B). Principal Component Analysis (PCA) is a mathematical technique designed to decrease the dimensionality of data while preserving the majority of the variation within the dataset. This reduction is achieved by recognizing principal components—directions where the data's variation is most significant. Through a selection of these components, each sample can be represented with a relatively small set of numbers instead of values for numerous variables. This facilitates the visualization of samples, allowing for the assessment of similarities and differences, and the determination of possible sample groupings [54].

Table 4 – Principal component analysis data

	PC1	PC2	PC3	PC4	PC5
Standard deviation	1,3823	1,3568	0,9998	0,35543	0,3500
Proportion of Variance	0,3821	0,3682	0,1999	0,02527	0,0245
Cumulative Proportion	0,3821	0,7503	0,9502	0,9755	1,0000

In order to enhance the comprehensibility of the analysis, the standard deviation and the proportion of variance are computed for each primary component. The relevant data for this analysis is presented in Table 4 and Figure 15. Notably, PC1, PC2, and PC3 exhibit a higher proportion of variance compared to other components.

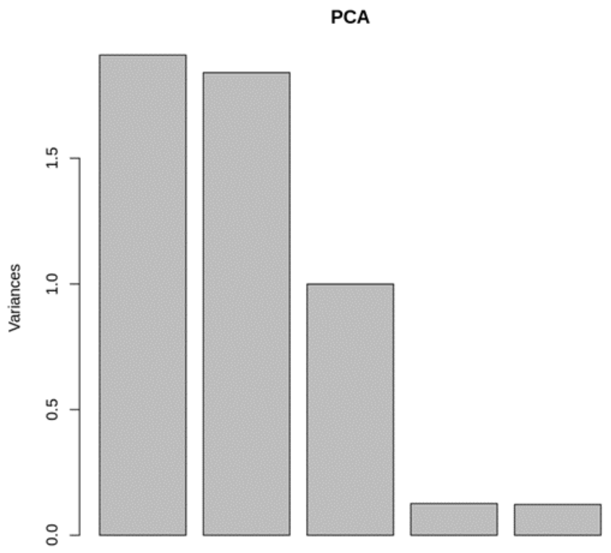


Figure 15– Proportion of variance of principal components

The initial three principal components display elevated standard deviation and variance. Consequently, an examination of the connection between the machine's condition (based on both the main failure and the five sub failure types) and these three principal components was conducted, as illustrated in Figures 16,17 and 18. In these plots, operational states are denoted by circles and blue color, while failed states are represented by triangles and red color.

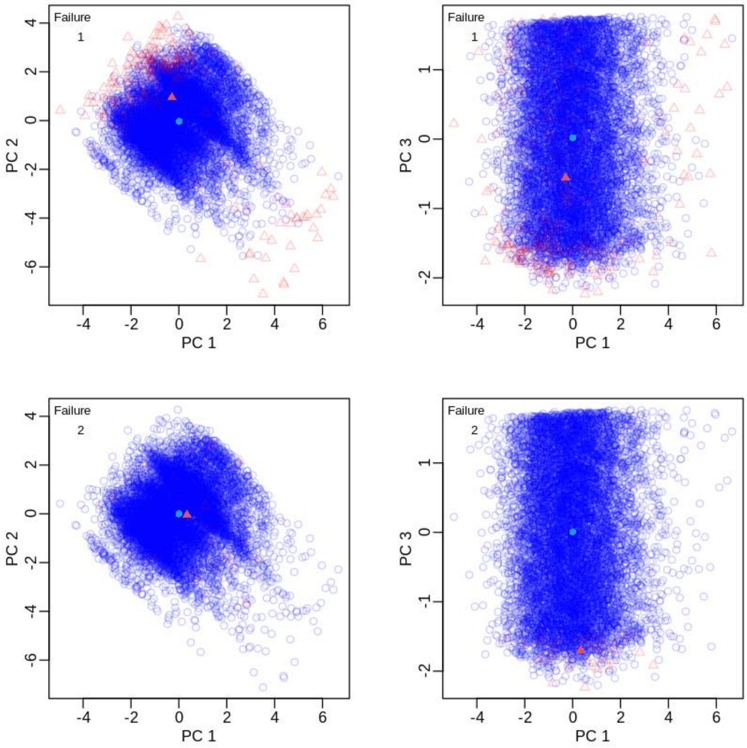


Figure 16– Plot of components 1-2 and 1-3 (Failure 1 and 2)

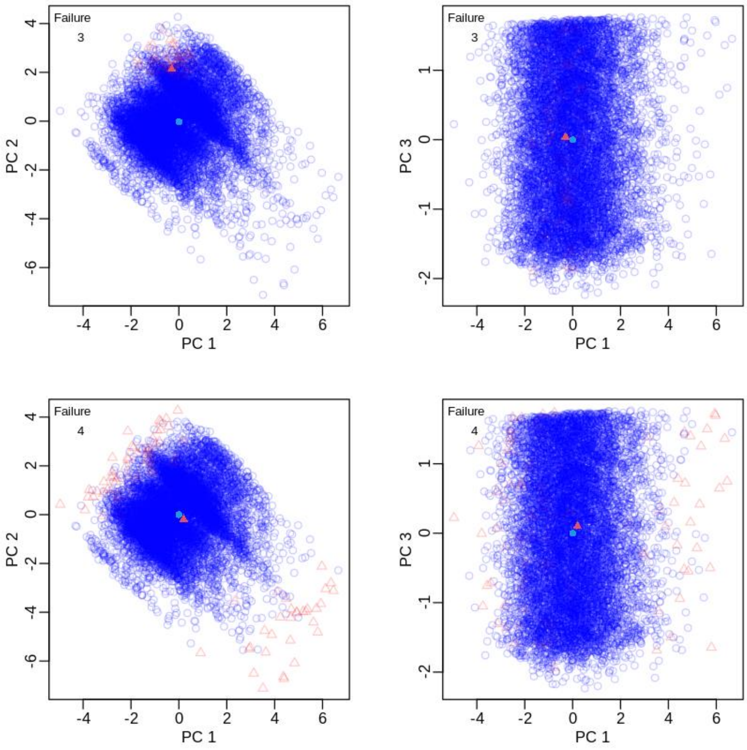


Figure 17– Plot of components 1-2 and 1-3 (Failure 3 and 4)

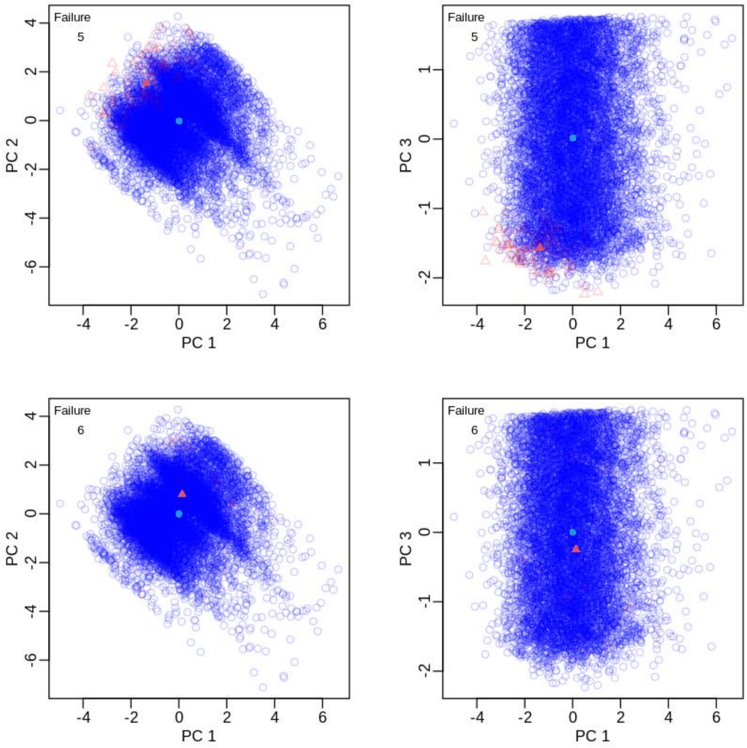


Figure 18– Plot of components 1-2 and 1-3 (between Failure 5 and 6)

A correlation analysis was conducted to examine the relationship between the principal components and the variables. The findings are presented in Table 5 and Figure 19. It is evident that there exists a linear correlation between "Rotational speed" and PC1, while "Torque" and PC3 exhibit a non-linear association. No significant relationship is observed between "Tool wear" and PC2, however, a notably strong negative correlation exists between "Tool wear" and PC3. These correlation patterns are clearly depicted in Figure 19, where the relationships are visually represented in a circular manner.

Table 5 – Correlation analysis between principal components and variables

	PC1	**PC2**	**PC3**	**PC4**	**PC5**
Air temperature	0.69907255	0.67016533	0.0153211682	-0.1506063195	-0.1981388428
Process temperature	0.69787205	0.67139795	0.0157262125	0.1522286529	0.1969225937
Rotational speed	0.68627876	-0.68303434	0.0030391755	-0.2004692296	0.1492930175
Torque	-0.68079682	0.68851655	0.0004900335	-0.2007074098	0.1489192723
Tool wear	0.02344833	0.01909654	-0.9995422782	-0.0006213836	0.0005881003

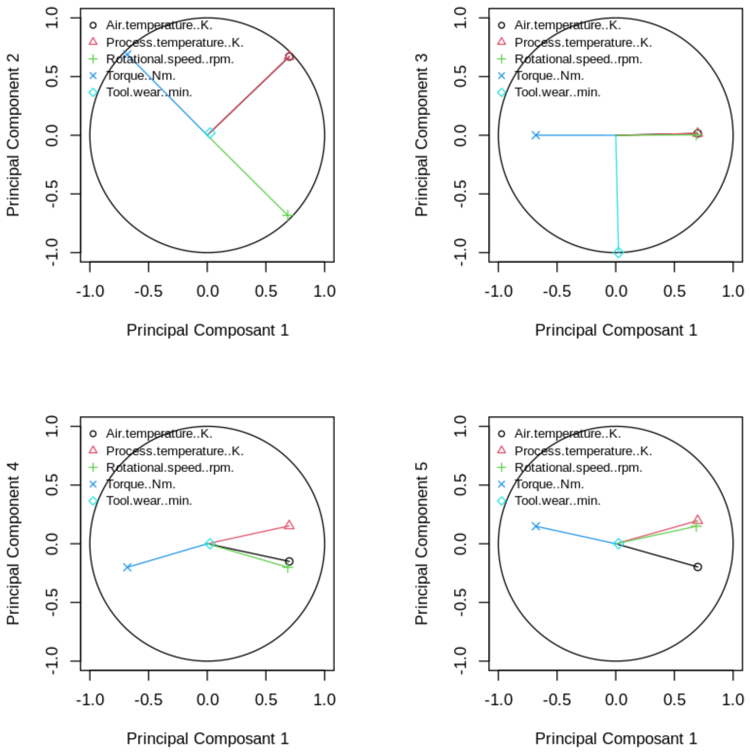


Figure 19– Correlation circles

When categorizing operational states, key components are utilized for prediction. Initially, principal component analysis is employed to condense the dataset's dimensions. The initial three principal components capture a significant portion of the variance. Figure 20 displays a new correlation matrix to assess whether these initial three principal components correlate with both main and sub-failures. The analysis reveals that the first principal component exhibits a slight negative correlation with OSF. PC2 demonstrates weak positive associations with main machine failures, HDF, and OSF. Notably, PC4 displays the strongest negative correlation with main machine failure and PWF. Conversely, despite the limited positive correlation between PC5 and main machine failure, PWF, and OSF, a weak negative correlation is observed with HDF. Consequently, all five key components are employed in the condition classification process, enhancing the accuracy of the forecasting in condition classification.

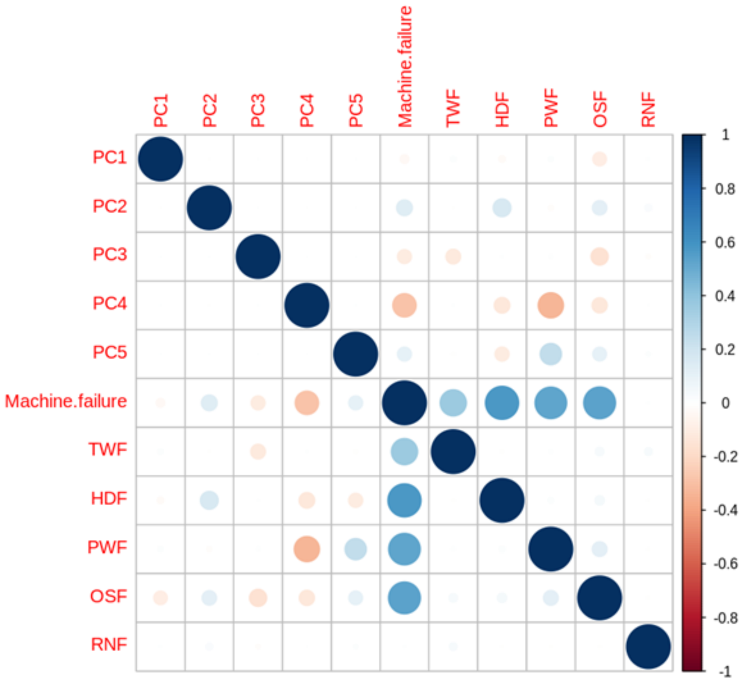


Figure 20– Correlation matrix of principal components and main and sub-failures

4.3.2 Classification of operating states according to Data set

Following the correlation and principal components analyses, along with the preparation of the dataset, it can be subjected to additional analysis utilizing different algorithms (see Appendix C). The objective is to determine which algorithm yields the most accurate prediction for both the operating and failed states with minimal error for this dataset. The calculation of these errors employs the mean squared error (MSE) as outlined in Equation 2.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \qquad (2)$$

where y_i represents the i^{th} observed value, $\hat{y}_i$ denotes the corresponding predicted value for y_i, and n signifies the total number of observations. The summation symbol $\sum$ denotes that the operation is performed over all values of i. To assess the effectiveness of a statistical learning method with a given dataset, its performance is evaluated by comparing the predicted data with the observed data. The Mean Squared Error (MSE) quantifies the level of error in statistical models by measuring the average squared difference between the observed and predicted values. An MSE of zero indicates a model with no error, while increasing model error leads to higher MSE values [55].

The dataset's observed data is initially split into two distinct data frames: a training set and a testing set. The training data predominantly comprises observations from data frame X, while the remaining data constitutes the test data. During the training phase, algorithms utilize the training data to learn patterns and relationships within the dataset. Conversely, the test data remains independent of the training process and is not employed in model training. This division into training and testing subsets is crucial for enabling robust comparison of results, ensuring the evaluation of model performance on unseen data.

The forecast's evaluation involves calculating three types of errors. Error 1 assesses the accuracy of algorithms in predicting operational states, while Error 2 evaluates their ability to predict failed states. The Total Error reflects the combined performance of predicting both operational and failed states. Consequently, the Total Error comprises Error 1 and Error 2. The objective is to minimize these errors, enabling the comparison of algorithms to determine which one achieves superior state prediction capabilities.

Five distinct algorithms are employed for decision-making purposes: Logistic Regression (LR), Linear Discriminant Analysis (LDA), Decision Tree (DT), Random Forest (RF), and Support Vector Machine (SVM). Furthermore, these models are aggregated using the Consensus ensemble method to form a unified consensus function. Ensemble classifiers combine the outputs of multiple autonomous base classifiers or machine learners with the aim of enhancing classification accuracy beyond what individual classifiers can achieve [56]. In this study, the analysis of data obtained from the consensus function involved the use of Minimum, Maximum, and Mean values, alongside the utilization of five different algorithms.

This study will explore three distinct scenarios. In the initial scenario, the analysis will involve 9661 operational machines with a value of b = 10, based on the dataset. In the first alternative scenario, the evaluation will focus on 500 operational machines, with the results analyzed under b = 10, maintaining the same number of failure machines. For the second alternative scenario, the operational machine count will remain consistent with the first alternative scenario, while the analysis will be conducted with b = 50.

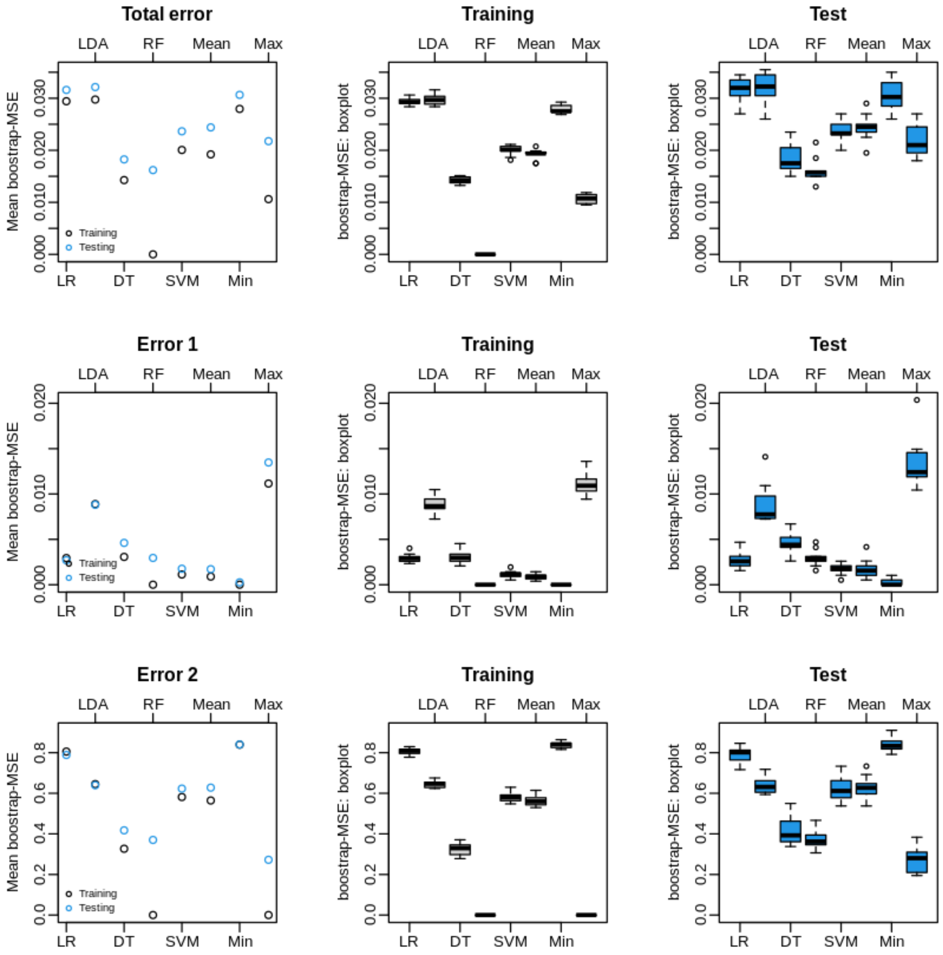


Figure 21– First results according to Machine Failure

Figure 21 displays the outcomes of algorithm implementation according to Machine Failure variable, showcasing error rate distributions through boxplots. These boxplots effectively illustrate the scatter of error rates. The spread within the boxplot indicates the extent to which error rates deviate from the mean. When error rates exhibit similar values, the boxplot's distribution is narrow. Conversely, if error rates vary significantly, the spread within the boxplot widens. Narrow spreads in the boxplot are indicative of more accurate error rate interpretations, suggesting minimal deviation from the mean.

The initial three plots depict the outcomes concerning total errors. In the leftmost plot, the points represent the mean values of Mean Squared Error (MSE) for both training and test data. The black dots represent the average error rate during training, while the blue dots denote the average MSE for testing. Subsequent boxplots illustrate the distribution of MSE across all algorithms. Each boxplot displays MSE values, providing insight into their spread. Additionally, the black lines within the boxplots represent the average values of the error rates.

In terms of total error, Random Forest exhibits the lowest error rate during training. Furthermore, it maintains a nearly identical rate compared to Random Forest in the Maximum value of the algorithms. During testing, Logistic Regression (LR), Linear Discriminant Analysis (LDA), Support Vector Machine (SVM), Mean, and Minimum values demonstrate almost identical and the lowest error rates. Conversely, the Maximum value of the algorithms shows the highest error rate during testing. The error rate during testing holds significance for forecasting as the test data differs from the training data, and the algorithm solely predicts operational states based on the test data.

In the remaining three plots displayed in the middle, the focus shifts to forecasting operational states. In this context, the algorithms exhibit minimal errors due to the abundance of data pertaining to operational states within the dataset, enabling more accurate forecasts. Random Forest (RF) and the Minimum value of the algorithms nearly achieve a 0-error rate for prediction in this category. Similar to the total error analysis, the Maximum value of the algorithms yields the highest error rate in predicting operational states.

In the final three graphics, the focus is on the error rate of forecasting failed status. The error rate is notably high in this case due to the limited data available for failed states within the dataset. Consequently, algorithms encounter challenges in accurately predicting these states. When comparing the forecasts of operational states and failed states, Logistic Regression (LR) and the Minimum value of the algorithms exhibit an 80% error rate. Conversely, Random Forest (RF) and the Maximum value of the algorithms demonstrate very few errors in predicting failed states.

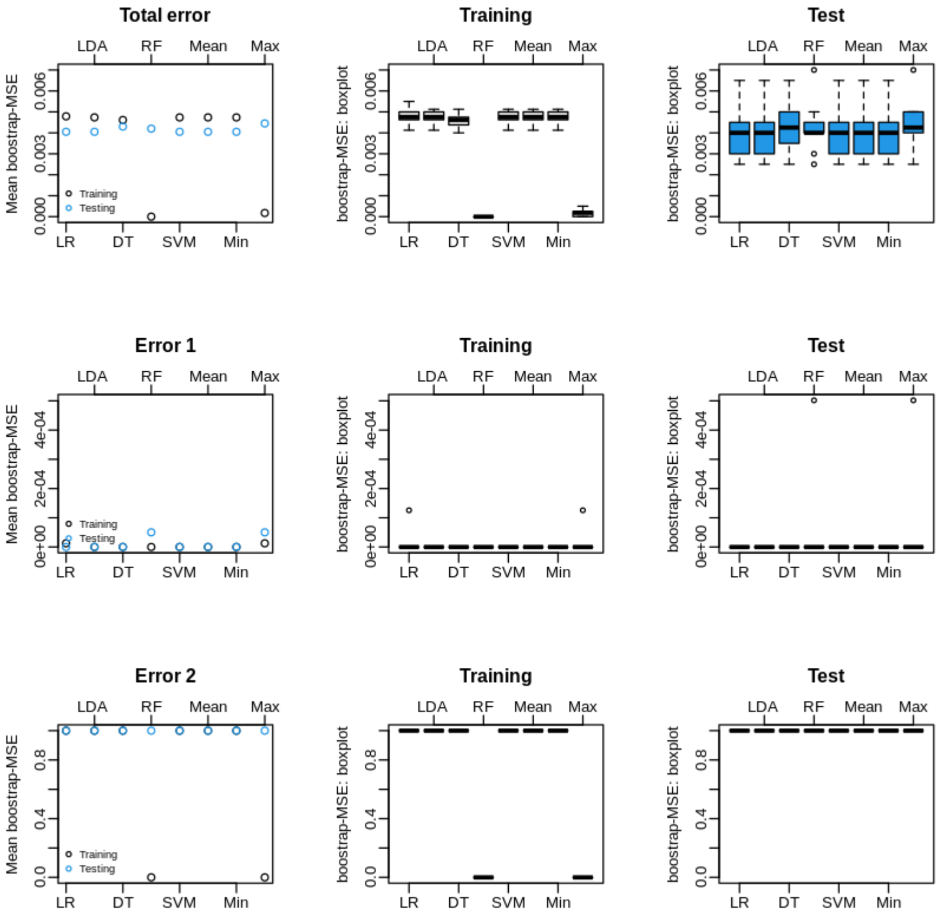


Figure 22– First results according to TWF

Figure 22 illustrates the results of implementing the algorithm based on variable TWF. In the initial three plots, both Random Forest and the Maximum value of the algorithms demonstrate the lowest error rates during the training phase. Additionally, the other algorithms exhibit nearly similar values but with higher error rates. However, during testing, Logistic Regression (LR), Linear Discriminant Analysis (LDA), Support Vector Machine (SVM), Mean, and Minimum values show almost identical and the lowest error rates. In contrast, the Maximum value of the algorithms indicates the highest error rate during testing.

Commenting on the middle graphs and the last three graphs is challenging due to the limited amount of available data. From the graphs, it's evident that making a precise analysis is difficult due to the scarcity of data points.

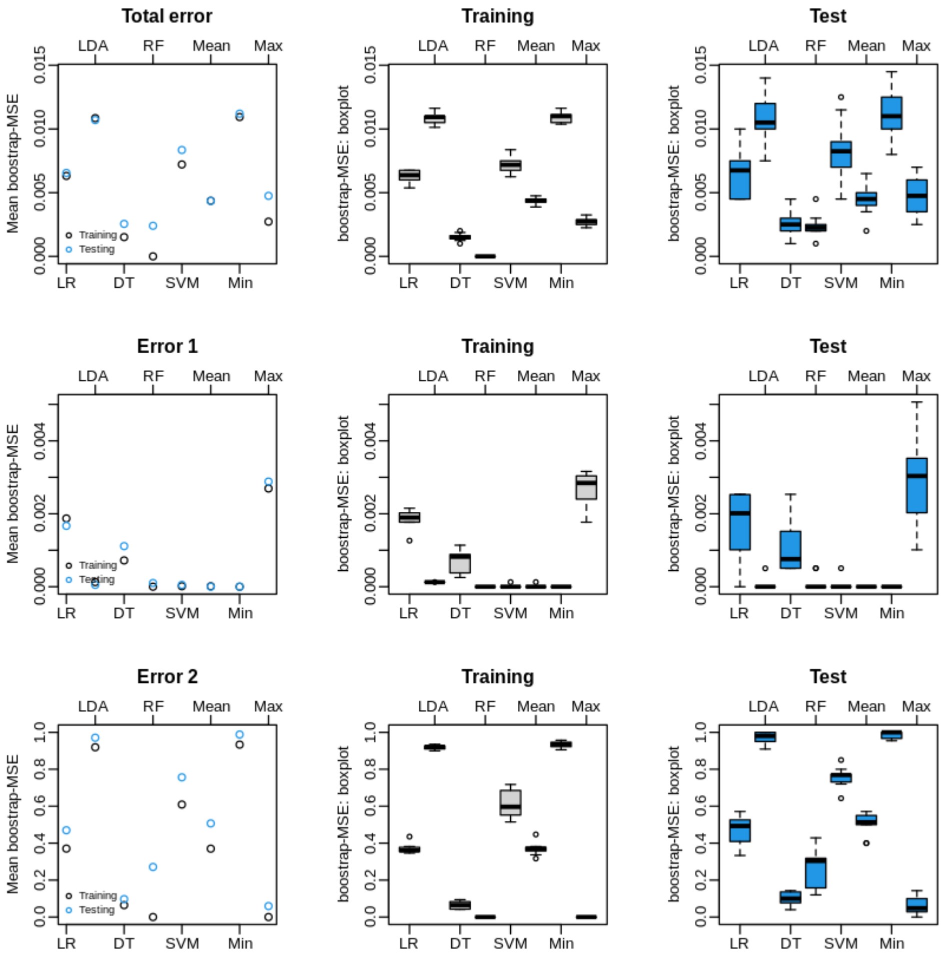


Figure 23– First results according to HDF

Figure 23 shows the outcomes of employing the algorithm with variable HDF. In the first three plots, Random Forest exhibits the lowest error rates during the training phase. Conversely, both LDA and Min algorithms demonstrate the highest error rates, which are approximately ten times greater than that of the RF error rate. However, during testing, the Min algorithm maintains the highest error rate, similar to during training.

Interestingly, in the three remaining graphs presented in the center, LDA, Random Forest (RF), SVM, mean, and minimum values of the algorithms almost achieve a close to perfect prediction scores for this category. This applies to both the training and testing stages. However, the "maximum value" method consistently suffers from the highest error rate.

In the last three visuals, the minimum value of the algorithms exhibits an error rate of nearly 50%. Conversely, Random Forest (RF) and the maximum value of the algorithms show very few errors during training. Additionally, the maximum value of the algorithms boasts the lowest error rate in both training and testing phases.

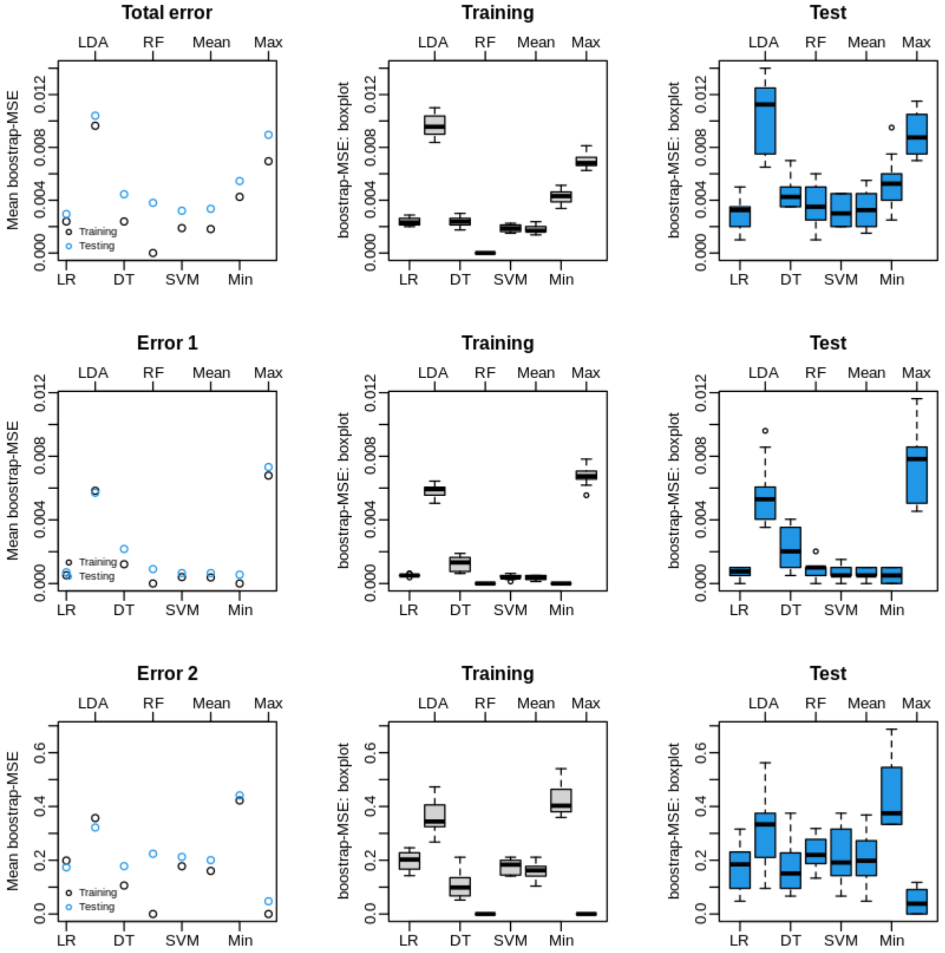


Figure 24– First results according to PWF

Figure 24 depicts the results obtained by utilizing the algorithm with different PWF settings. In the first three graphs, Random Forest displays the least number of errors during the training period. Nevertheless, in the testing phase, LDA shows the highest error rates, mirroring its performance during training.

In the subsequent three diagrams depicted in the center, LR, Random Forest (RF), SVM, the mean, and the minimum value of the algorithms almost reach a zero-error rate for forecasting during both training and testing, similar to the previous phase.

However, the "maximum value" and LDA methods consistently experience higher error rates compared to others in predicting operational states, which remains consistent across both the training and testing stages.

In the last trio of visual representations, Logistic Regression (LR) and the Minimum value of the algorithms exhibit the highest error rate in both Training and Test phases. Conversely, Random Forest (RF) and the Maximum value of the algorithms showcase minimal errors in forecasting failed states during the Training phase.

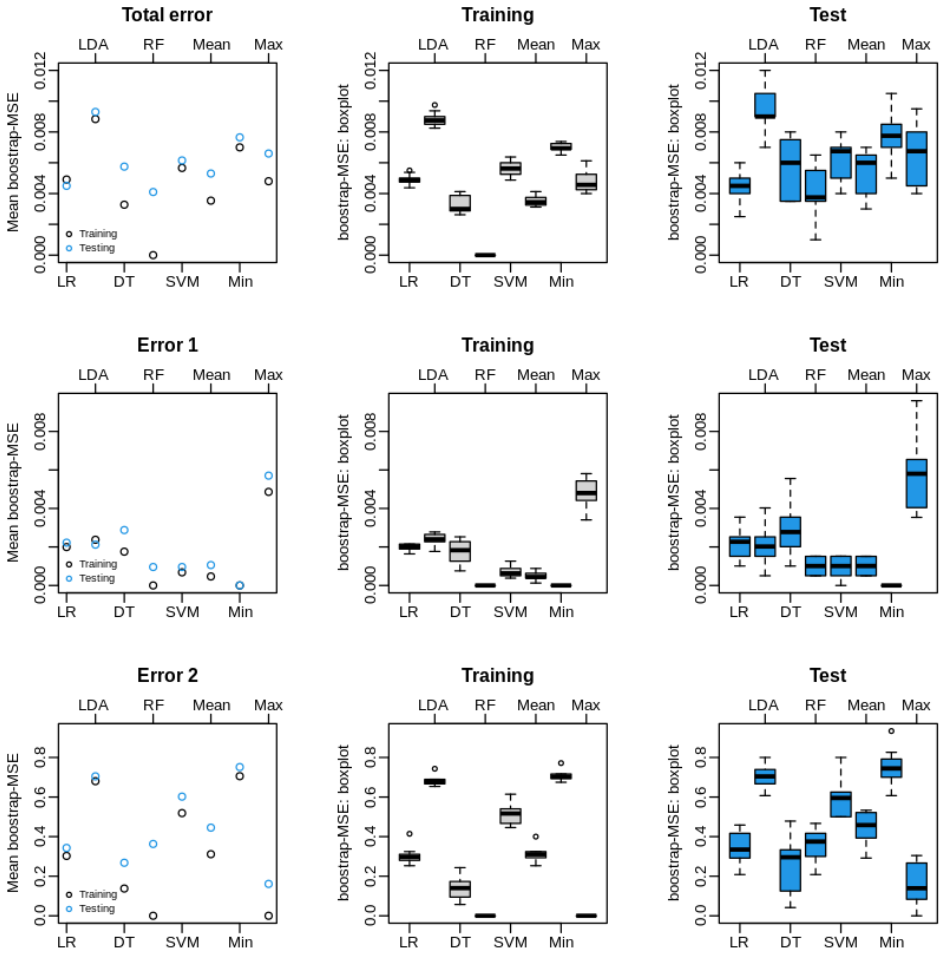


Figure 25– First results according to OSF

Figure 25 illustrates the outcomes of applying the algorithm with varying OSF. Initially, Random Forest exhibits the lowest error rates during the training phase in the first three plots. However, during testing, LDA demonstrates the highest error rates, reflecting its performance observed during the training phase.

In the following set of three graphs presented in the center, Random Forest (RF) and the minimum values of the algorithms nearly attain close-to-perfect prediction scores during the Training phase. This pattern is consistent for the minimum values of algorithms across both the training and testing stages. However, the maximum value of algorithms consistently exhibits higher error rates compared to others in predicting operational states.

In the final three visuals, LDA and the minimum value of the algorithms demonstrate an error rate of approximately 70%. In contrast, Random Forest (RF) and the maximum value of the algorithms exhibit minimal errors during the training phase. Moreover, the maximum value of the algorithms achieves the lowest error rate across both the training and testing phases.

Understanding the graphs in Figure 26 poses a challenge due to the limited amount of available data. It is evident from the graphs that conducting a precise analysis is difficult due to the scarcity of data points.

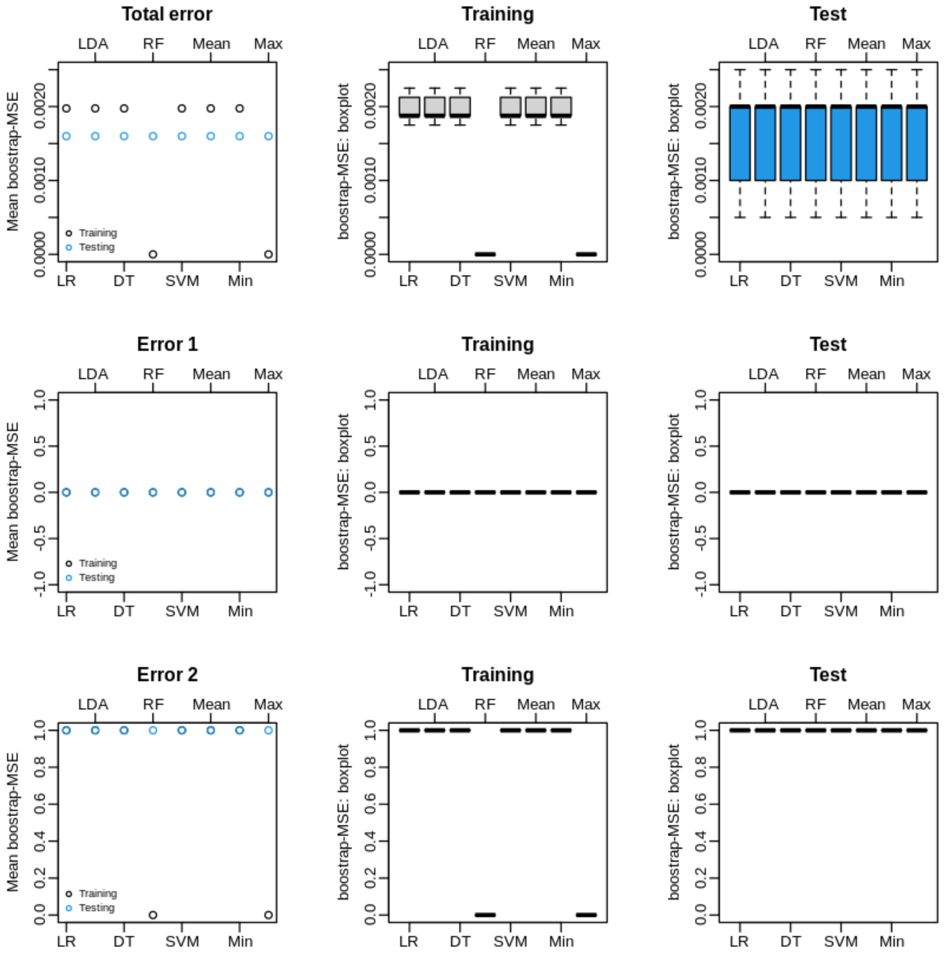


Figure 26– First results according to RNF

The graphs demonstrate varying algorithm performances across different conditions (inclusive of all, healthy, and failure conditions) and types of failures. Upon analysis, it's evident that in certain instances, there were too few failed machines to conduct a reliable analysis. Consequently, the number of operational machines samples will be reduced to 500 while maintaining the same number of failed machines. Additionally, in the initial scenario, algorithms were executed only 10 times. In the forthcoming analysis, a decision will be made to increase this repetition count, allowing for an exploration of its impact on the results.

4.3.3 Classification of operating states according to alternative scenario 1

In this section of the paper, new findings were achieved by decreasing the quantity of operational machines to 500 (see Appendix D) while maintaining a consistent number of algorithms runs at 10.

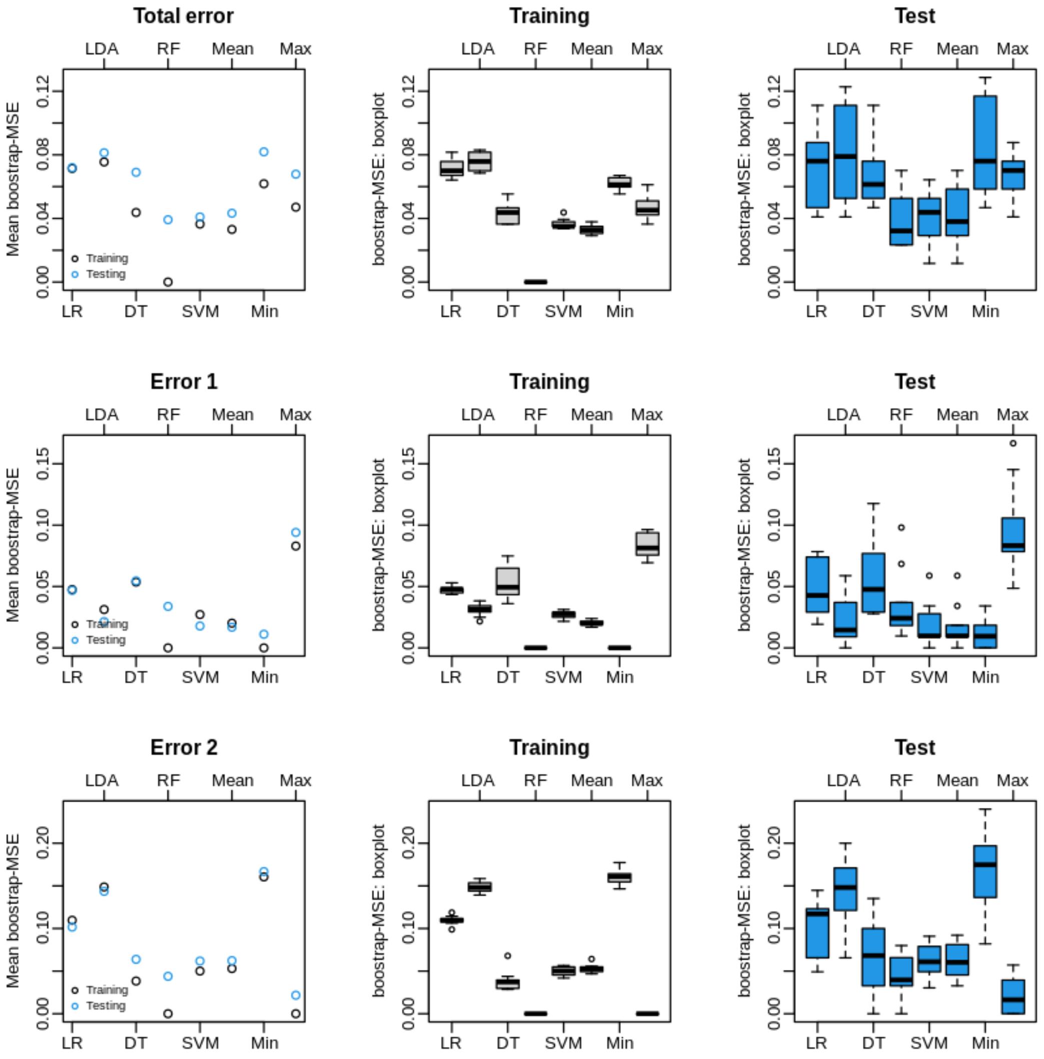


Figure 27– Results according to Machine Failure with 500 operational machines

Figure 27 depicts the results obtained from applying the algorithm across all instances of machine failures. In comparison to Figure 21, within the initial three plots of the Training section, there was an observed rise of 2-4 times in the Mean Squared Error (MSE) values for each algorithm except for RF. A comparable trend was noticed in the

Testing section, accompanied by an expansion in the range between the minimum and maximum quartiles of the boxplots.

In the three remaining graphs displayed in the center, RF and the minimum values of the algorithms nearly reach close-to-perfect prediction scores, resembling the previous conditions observed during training. Nevertheless, akin to the training graph encompassing all machines, there is a pronounced surge in the MSE value. A similar scenario persists in the Test section, where the gap between the minimum and maximum quartiles in the boxplots continues to widen.

In the final three visuals, within both the Training and Test graphs, unlike the scenarios involving all machine states and operational machine states, there is a substantial decrease in MSE values. In both instances, numerous algorithms exhibit error rates approaching zero.

Figure 28 illustrates the outcomes derived from implementing the algorithm across TWF. In Figure 22, noteworthy outcomes were not achieved due to the minimal number of failures in comparison to operational machines. However, with the operational machines reduced to 500 in Figure 28, clearer and more interpretable results were obtained.

Compared to Figure 22, in the first three plots of the Training and Test section, it is evident that there is a notable rise in the MSE value across all instances of machine failure. Additionally, there is an observable widening in the range between the minimum and maximum quartiles of the boxplots in both scenarios.

In contrast to Figure 22, the current graphs exhibit interpretable outcomes. In the training section, RF demonstrates results that are nearly perfect, mirroring many other instances. Despite the dissimilarity in performance among other algorithms, the overall situation is promising due to their notably low MSE values. Similarly, MSE values in the test section are also low. Given the proximity of algorithm values to each other, it becomes challenging to ascertain which one yields the most optimal results.

In the last three visuals, both in the Training and Test graphs, there is a decline observed in nearly all MSE values.

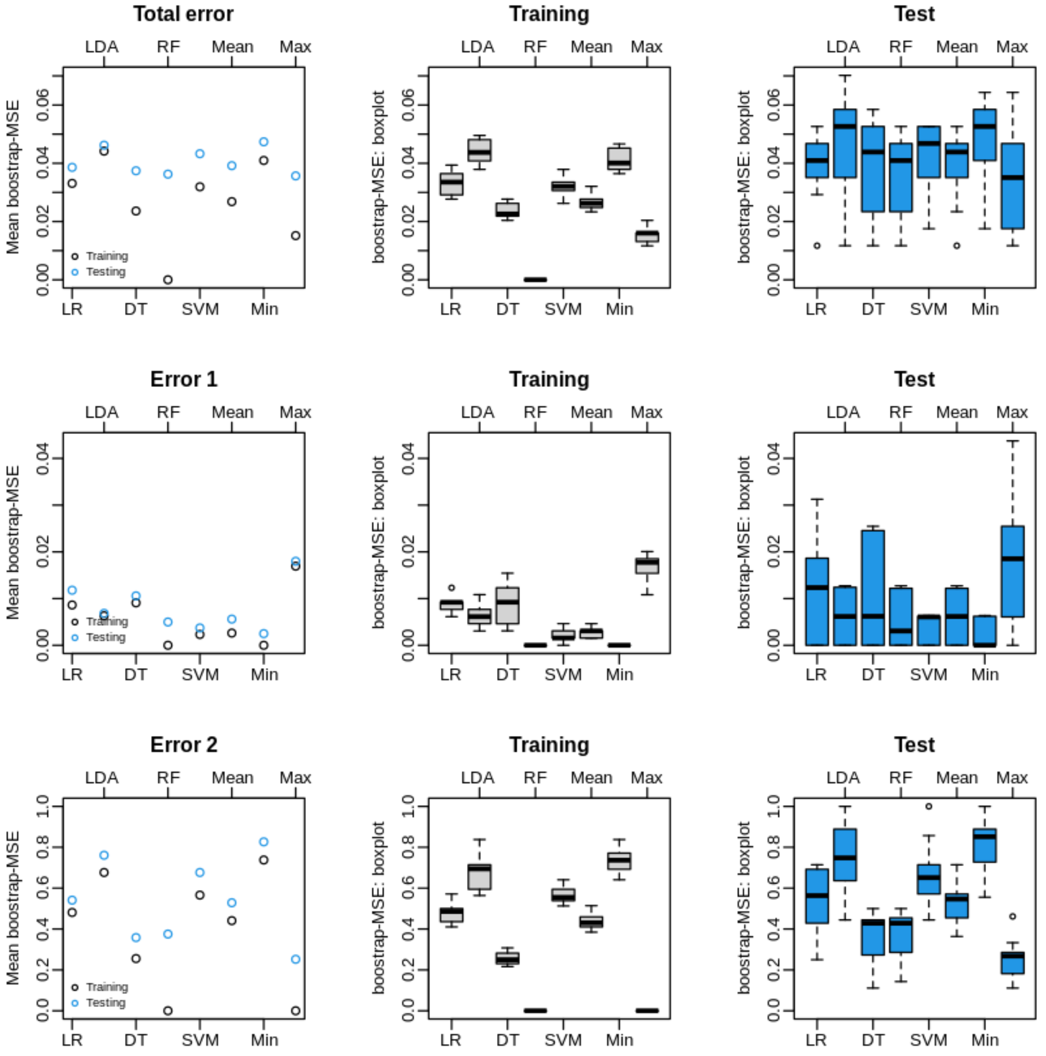


Figure 28– Results according to TWF with 500 operational machines

Figure 29 presents the results obtained from utilizing the HDF algorithm. Contrary to Figure 23, the initial three plots within both the Training and Test sections exhibited a notable rise in the Mean Squared Error (MSE) values for each algorithm. Notably, only RF's MSE value remained unchanged in the Training section. Additionally, an increase in the range between the minimum and maximum quartiles of the boxplots was observed in the Testing section.

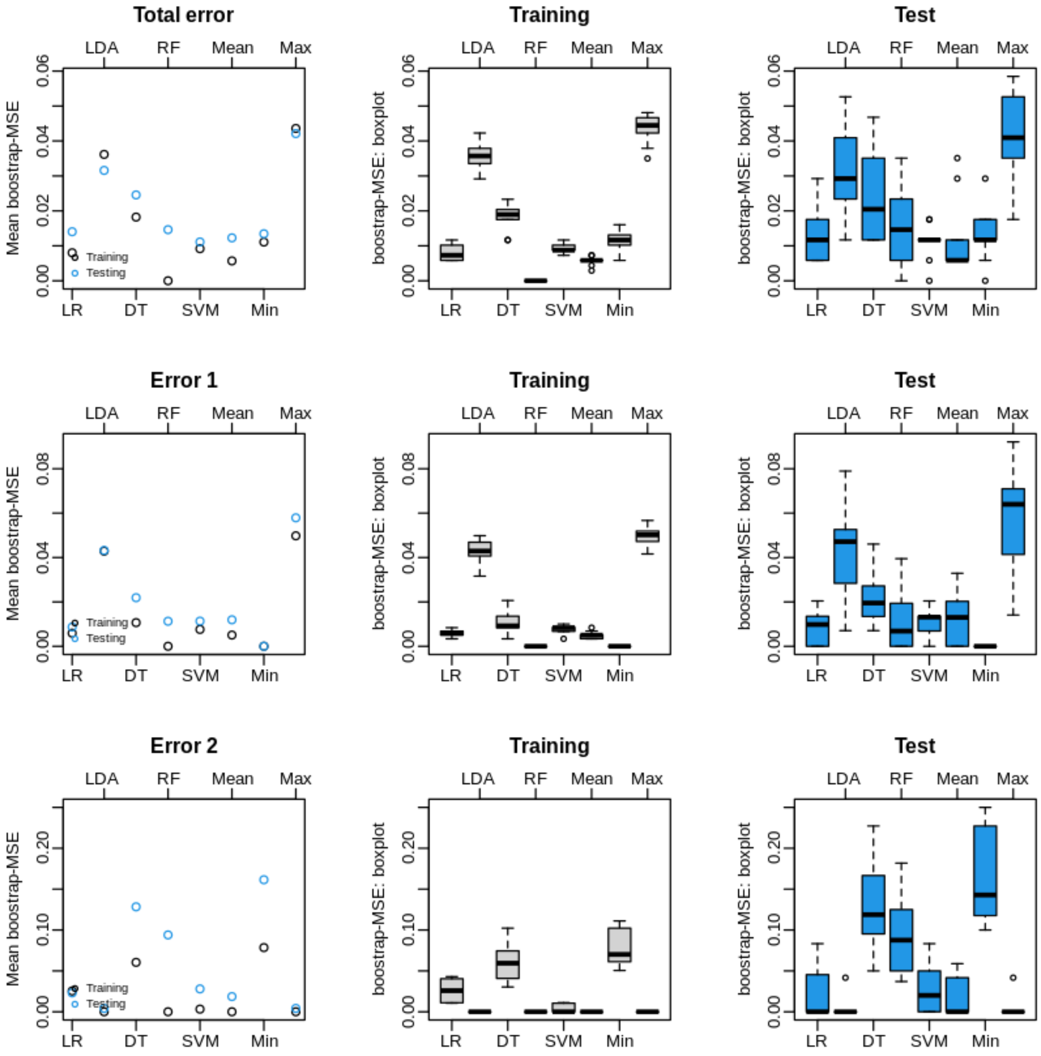


Figure 29– Results according to HDF with 500 operational machines

In the three remaining graphs shown in the middle, RF and the minimum values of the algorithms almost achieve near-perfect prediction scores during Training. Additionally, LR, SVM, and the mean of algorithms also exhibit similar performance. However, during the test phase, in comparison to Figure 23, a notable increase was observed in the MSE values of LDA, RF, SVM, and the Mean of algorithms, deviating from the near-perfect MSE value. This is evident in the Test section, where the gap between the minimum and maximum quartiles in the boxplots continues to widen.

In the last three graphs within the Training section, there's a considerable reduction in MSE values compared to scenarios involving all machine states and operational machine states. Several algorithms have approached perfection. In the test scenario, a noticeable decrease in MSE values was observed for many algorithms, mirroring the trend seen in the Training scenario.

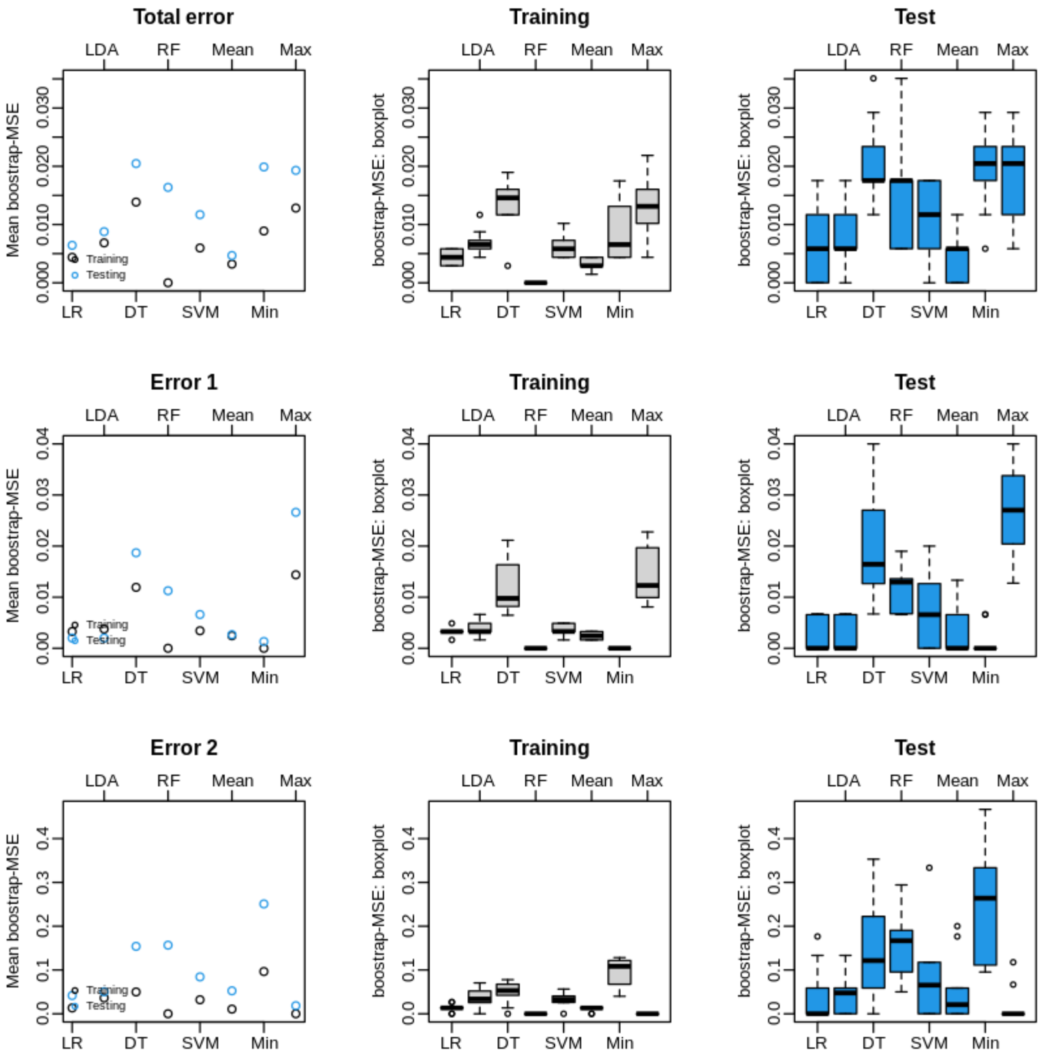


Figure 30– Results according to PWF with 500 operational machines

Figure 30 depicts the outcomes achieved through the utilization of the PWF algorithm. In contrast to Figure 24, there was a noticeable increase in the Mean Squared Error (MSE) values for nearly every algorithm in the initial three plots within both the Training and Test sections. Interestingly, the MSE value of LDA decreased in almost both cases. Additionally, RF's MSE value remained unchanged in the Training section.

In the subsequent three graphs positioned centrally, RF and the minimal values of the algorithms nearly attained predictions bordering on perfection during the Training phase. Additionally, the minimum values of the algorithms maintained identical predictions during the Test phase as in Training. Nonetheless, throughout both phases, a substantial escalation in the MSE values for nearly all algorithms was observed in comparison to Figure 24. Notably, in the Test section, there was a widening gap between the minimum and maximum quartiles in the boxplots.

In the final three graphs, a significant decrease in MSE values is evident compared to scenarios encompassing all machine states and operational machine states during Training. Multiple algorithms have neared perfection. Similarly, in the test scenario, a noticeable decrease in MSE values was observed for many algorithms, reflecting the trend observed in the Training scenario. Additionally, the maximum values of the algorithms have achieved perfection.

Figure 31 illustrates the results obtained from employing the OSF algorithm. Unlike Figure 25, there was a clear rise in the Mean Squared Error (MSE) values across almost all algorithms in the first three plots in both the Training and Test sections. Furthermore, the MSE value for RF stayed consistent and demonstrated perfect predictive performance in the Training section.

In the following three centrally positioned graphs, RF and the mean and minimum values of the algorithms approached predictions close to perfection during the Training phase. Furthermore, the minimum values of the algorithms maintained consistent predictions during the Test phase as observed in Training. However, across both phases, a significant increase in the MSE values for almost all algorithms was noted compared to Figure 25.

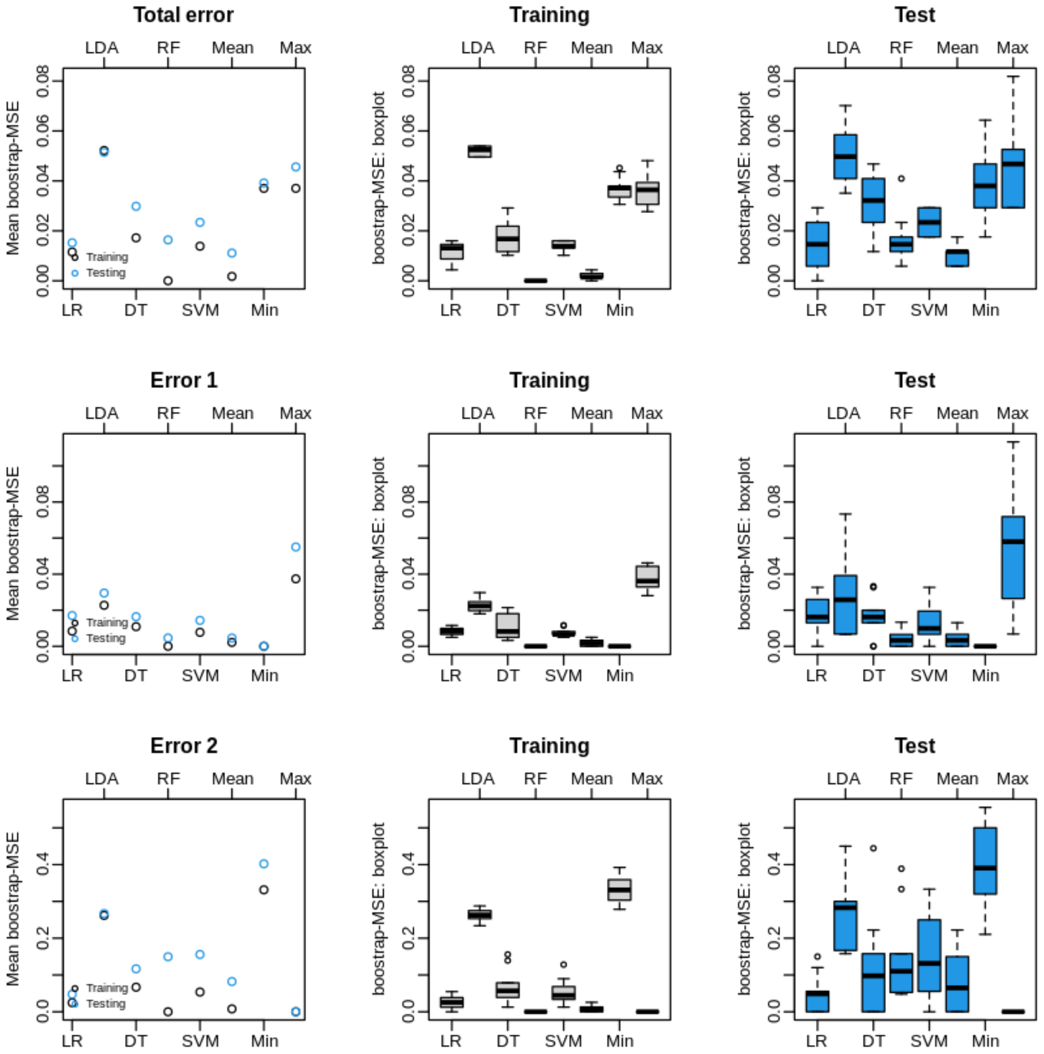


Figure 31– Results according to OSF with 500 operational machines

In the last three graphs, there's a notable reduction in MSE values compared to situations involving all machine states and operational machine states during Training. RF, Mean, and maximum of algorithms have approached perfection. Similarly, in the test scenario, a distinct decrease in MSE values was noted for all algorithms, mirroring the trend seen in the Training scenario. Furthermore, the maximum of the algorithms has reached perfection during the Test phase, like the Training phase.

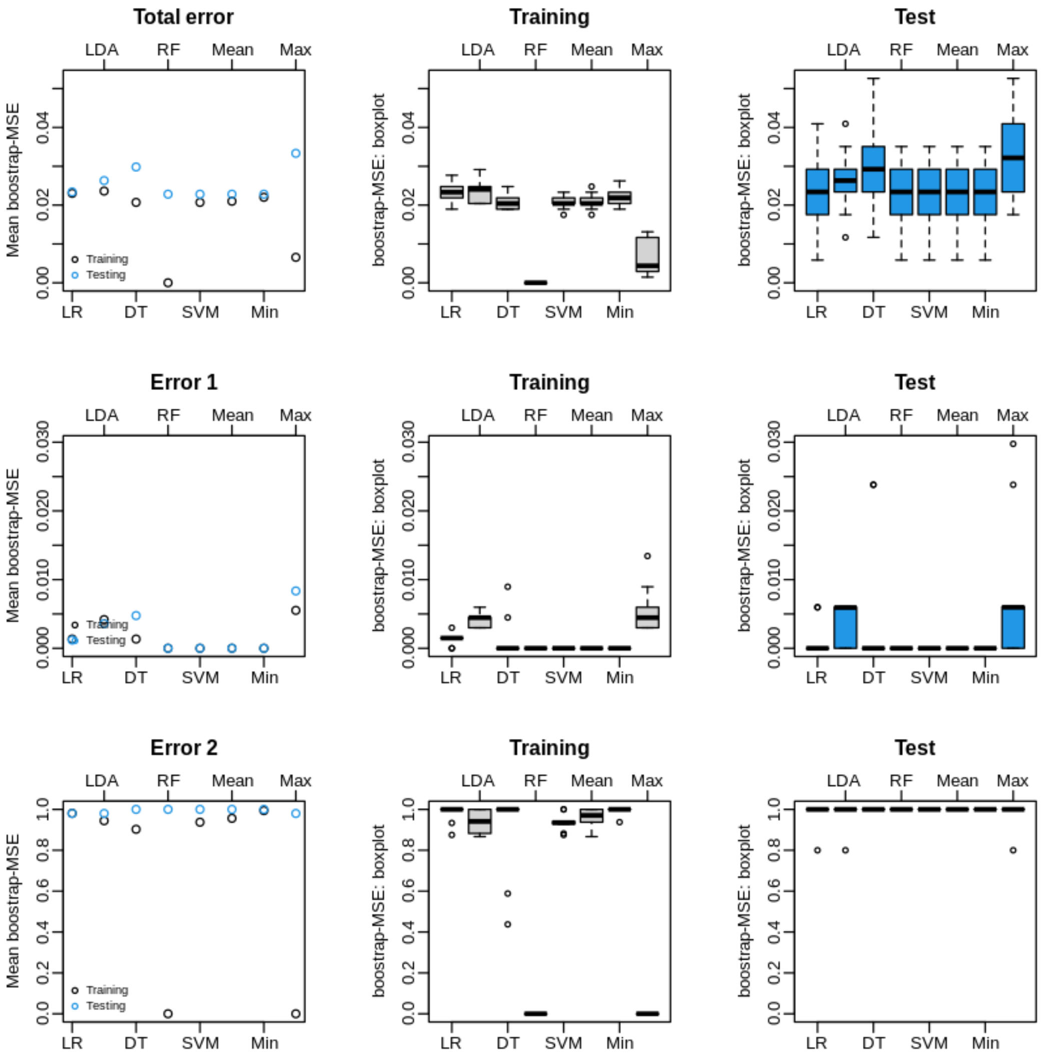


Figure 32– Results according to RNF with 500 operational machines

Figure 32 depicts the results obtained from applying the algorithm across RNF. In Figure 26, significant results weren't attained because of the limited instances of failure when contrasted with operational machines. Nonetheless, in Figure 28, where the operational machines are reduced to 500, although there remain aspects that are challenging to analyze, more understandable outcomes were achieved compared to Figure 26.

When comparing Figure 26, within the initial three plots of the Training and Test section, a significant increase in the MSE value is apparent across nearly all instances of machine failure. Only RF maintained the ideal MSE value, as observed in the initial scenario.

Providing detailed analysis for the middle graphs and the last three graphs is challenging due to the scarcity of available data in the initial scenario. Specifically, in the Training graph for operational machines in the middle section, a minimal MSE value was observed for the LDA and maximum of algorithms, a trend also observed in the Test case. Although minor changes are noted in the last three graphs, the overall situation closely resembles that of Figure 26. In summary, the limited data points make it challenging to conduct a precise analysis.

4.3.4 Classification of operating states according to alternative scenario 2

In this section of the paper, novel findings were made by maintaining a constant number of operational machines at 500 and elevating the consistent number of algorithms runs to 50, in contrast to the previous scenario. In this section, the obtained results will be compared with those from the first and second case scenarios.

Figure 33 illustrates the outcomes derived from implementing the algorithm across all instances of machine failures. When compared to Figure 21, there was an observed increase of 2-4 times in the Mean Squared Error (MSE) values for each algorithm within the initial three plots of the Training section, except for RF. The results are nearly identical to the Training graph in Figure 27. However, the widening trend observed in the range between the minimum and maximum quartiles of the boxplots seen in Figure 27 has diminished for many algorithms in Test scenario, resulting in improved MSE values.

In the three remaining graphs shown in the middle, RF and the minimum values of the algorithms almost achieve close-to-perfect prediction scores, like the first two scenarios during training. Despite a notable increase in the MSE value compared to Figure 21, similar MSE values were obtained as in the first Training graph, as seen in Figure 27. A similar pattern is observed in the testing section.

In the final three visuals, a substantial decrease in MSE values is noted in both the Training and Test graphs compared to Figure 21. This is unlike the scenarios involving all machine states and operational machine states, where very similar results are observed with Figure 27 during Training. Aside from this, the widening trend observed in the range between the minimum and maximum quartiles of the boxplots seen in Figure 27 has decreased for many algorithms in the Test scenario and a decrease in MSE values was observed.

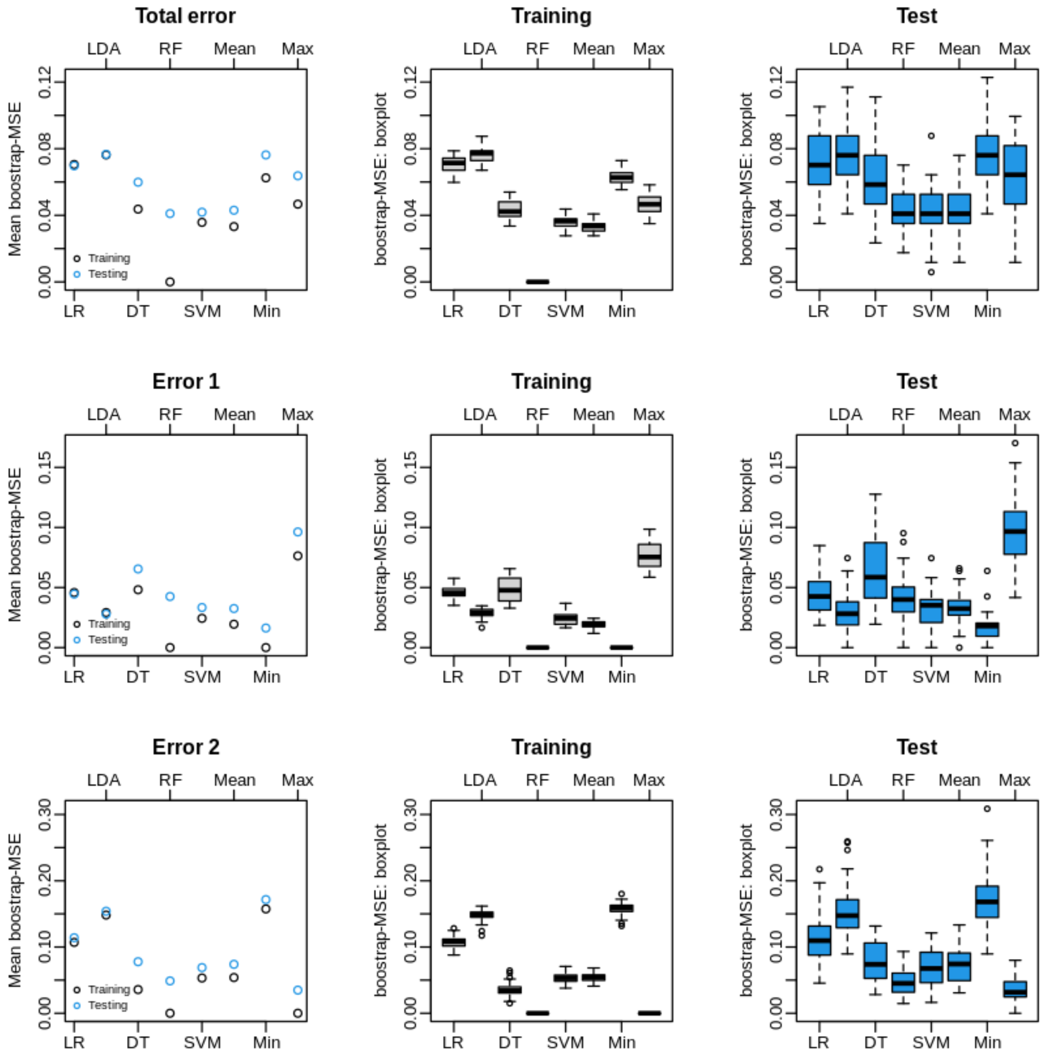


Figure 33– Results according to Machine Failure with 500 operational machines and 50 repetitions

Figure 34 depicts the results obtained from applying the algorithm across TWF. Contrasting with Figure 22, there has been a notable rise observed in the Mean Squared Error (MSE) values for nearly every algorithm within the first three plots of the Training section, except for RF. The outcomes closely resemble those in the Training graph of Figure 28. Nevertheless, the widening trend seen in the range between the minimum and maximum quartiles of the boxplots in Figure 2 has decreased for several algorithms in the Test scenario, leading to enhanced MSE values.

In the remaining graphs illustrating operational and failure scenarios, no discernible results were acquired for Figure 22 that could be elaborated upon.

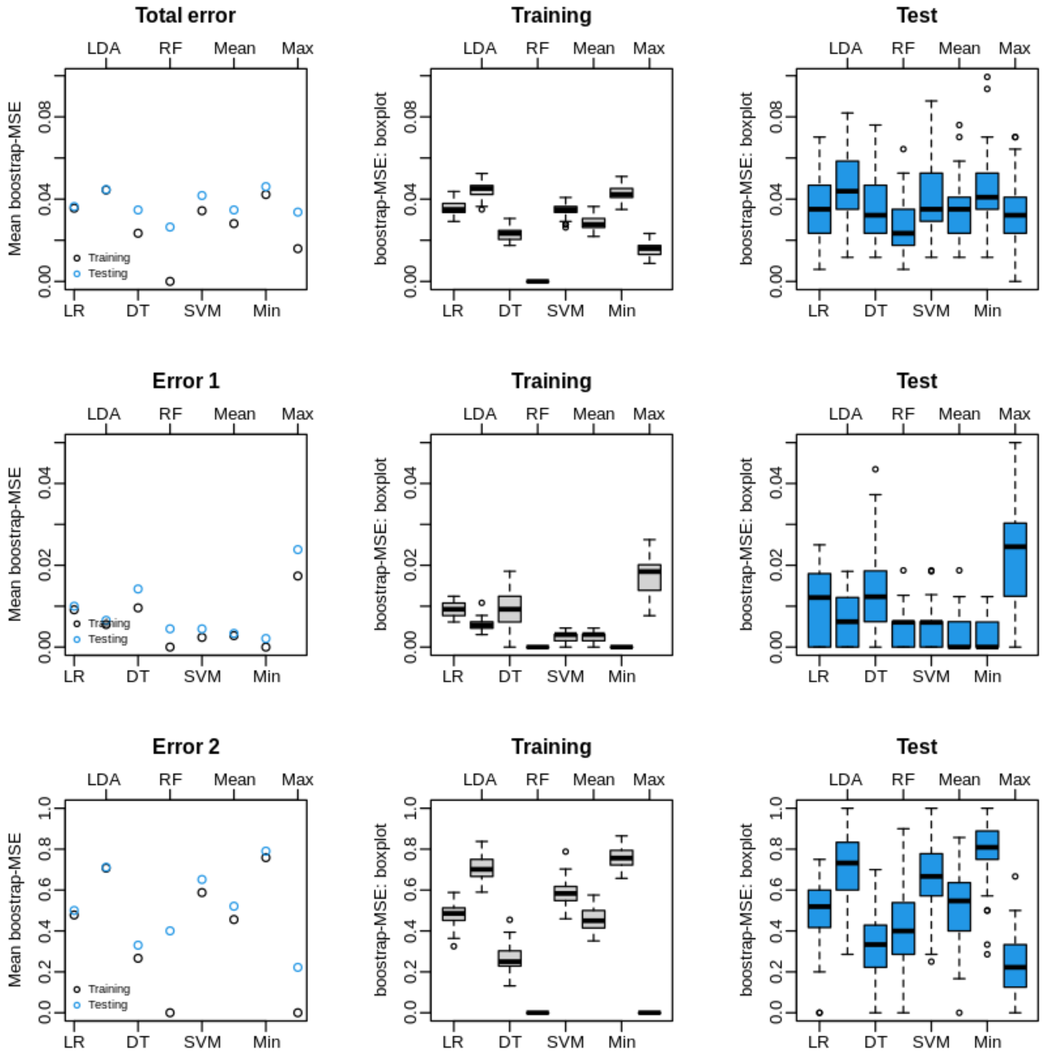


Figure 34– Results according to TWF with 500 operational machines and 50 repetitions

On the other hand, RF and the minimum values of the algorithms nearly reach near-perfect prediction scores for the three remaining graphs depicted in the middle, resembling Figure 28 during training. Comparable MSE values were attained as in the initial Training graph, akin to Figure 28. Additionally, a similar pattern is noted in the testing section.

In the last three visuals, very similar results to Figure 28 during Training are observed. Apart from this, the widening trend seen in the range between the minimum and maximum quartiles of the boxplots in Figure 28 has diminished for many algorithms in the Test scenario. Commenting on MSE values is challenging. While in some cases there was a slight increase or decrease, in others, it remained constant.

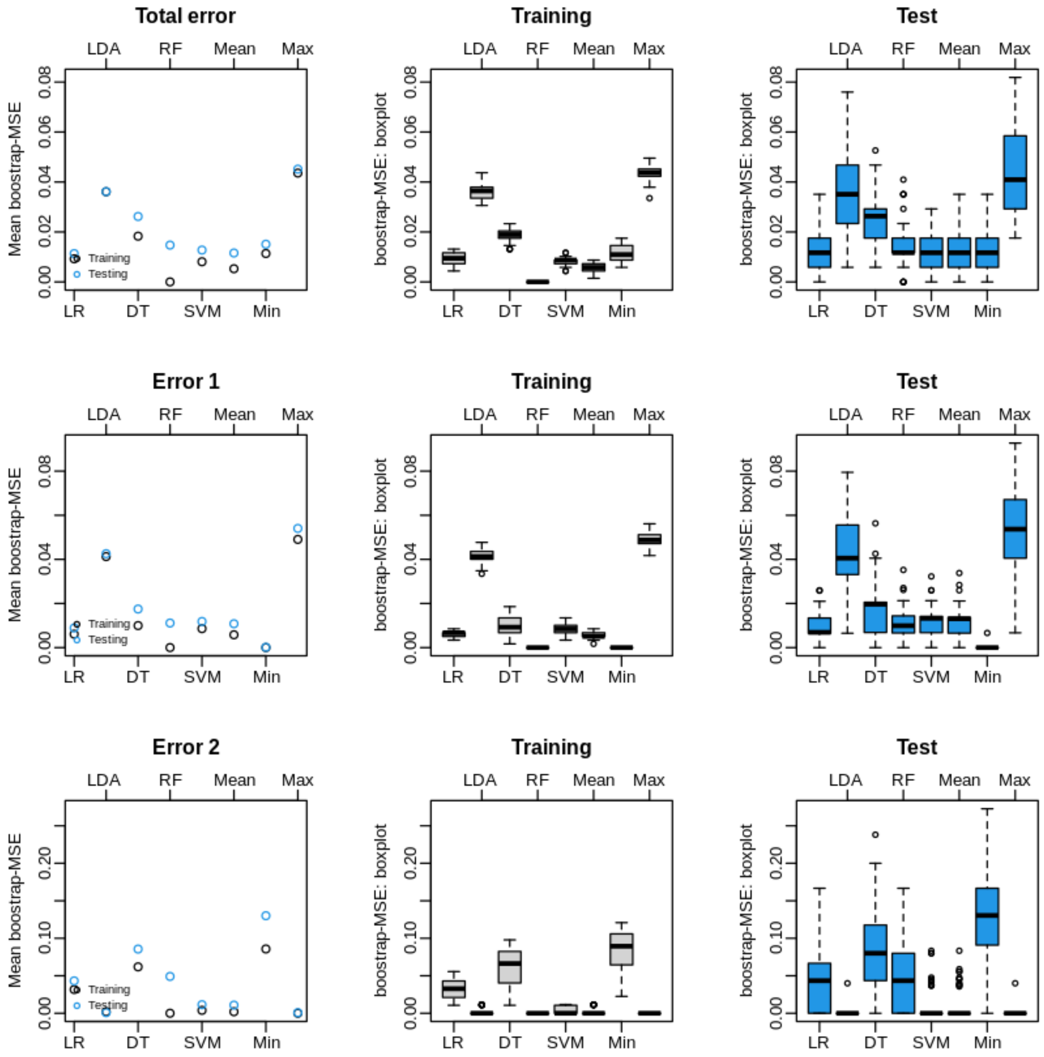


Figure 35– Results according to HDF with 500 operational machines and 50 repetitions

Figure 35 delineates the outcomes derived from applying the algorithm across HDF. In contrast to Figure 23, a conspicuous elevation in Mean Squared Error (MSE) values was noted for each algorithm within the initial three plots of the Training scenario, except for RF. During the test scenario, in conjunction with notable increments in MSE values, algorithms exhibiting MSE values akin to those depicted in Figure 23 are also discerned. The findings closely mirror those of the Training graph in Figure 29. Offering commentary on MSE values during the Test phase poses a challenge. While minor fluctuations were evident in certain instances, in others, they remained static.

In the three intermediary graphs presented, RF and the minimal values of the algorithms nearly attain prediction scores close to perfection, akin to the initial two scenarios during training. Despite a significant uptick in the MSE value in comparison to Figure 23, similar MSE values were acquired as those in the primary Training graph, as depicted in Figure 29. A parallel pattern is evident in the testing section across these two scenarios.

In the last three visuals, a significant decline in MSE values is observed in both the Training and Test graphs compared to Figure 23. This contrasts with the scenarios encompassing all machine states and operational machine states, where very comparable results are seen with Figure 29 during Training. Moreover, the widening trend noticed in the range between the minimum and maximum quartiles of the boxplots observed in Figure 29 has diminished for many algorithms in the Test scenario. Additionally, alongside LDA and the Maximum of algorithms, RF and Mean of algorithms approach perfect MSE values.

Figure 36 illustrates the results obtained by implementing the algorithm across PWF. In contrast to Figure 24, certain algorithms showed an uptick in Mean Squared Error (MSE) values in the initial three graphs of the training scenario, while others demonstrated consistent outcomes. Notably, in the testing scenario, the MSE value for the LDA algorithm remained relatively unchanged, yet there were significant increases in MSE values across nearly all algorithms. These findings closely resemble those depicted in the Training graph in Figure 30. While some MSE values exhibited similarity during the testing phase, there were observed increases in the values for certain algorithms.

In the three intermediary graphs provided, RF and the minimum values of the algorithms almost achieve prediction scores close to perfection, resembling the initial two scenarios observed during training. Despite a notable increase in the MSE value compared to Figure 24, comparable MSE values were obtained to those depicted in the primary Training graph as illustrated in Figure 30. However, superior outcomes were attained for DT and Max algorithm values. Although considerably higher MSE values are apparent in the Test case relative to Figure 24, some decreases are also evident alongside results akin to those seen in Figure 30.

In the last three visuals, a notable decrease in MSE values is noted in both the Training and Test graphs compared to Figure 24. Conversely, there are MSE values very similar to those in Figure 30. While the variance between the minimum and maximum quartiles of the boxplots seen in Figure 30 remains consistent for many algorithms in the Test scenario, there is a significant reduction in the DT and Minimum of algorithms.

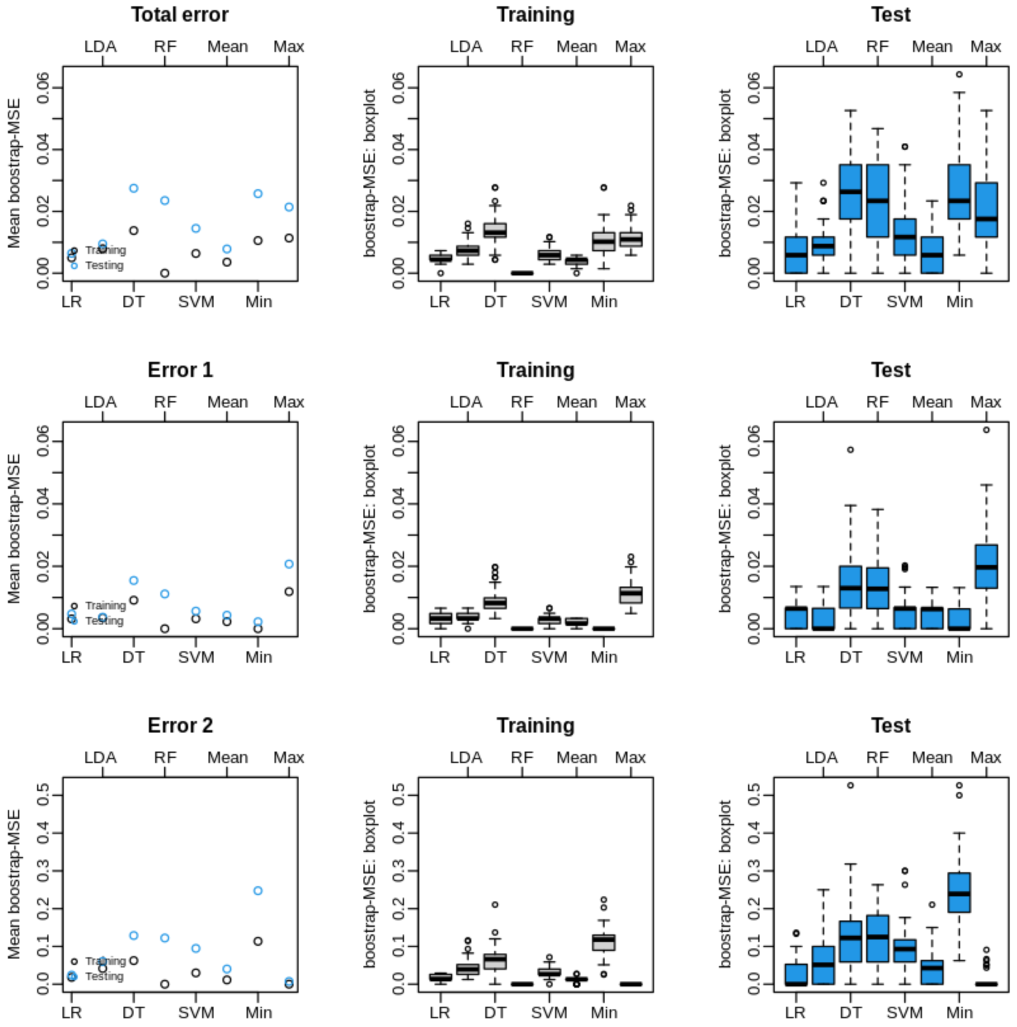


Figure 36– Results according to PWF with 500 operational machines and 50 repetitions

Figure 37 depicts the outcomes of applying the algorithm across OSF. Unlike Figure 25, a noticeable rise in Mean Squared Error (MSE) values was observed for every algorithm within the first three plots of the Training scenario, except for RF. This significant increase in the MSE value is evident across all algorithms in the Test case.

Similar results were obtained in both the Training and Test scenarios in Figure 36, resembling those in Figure 37.

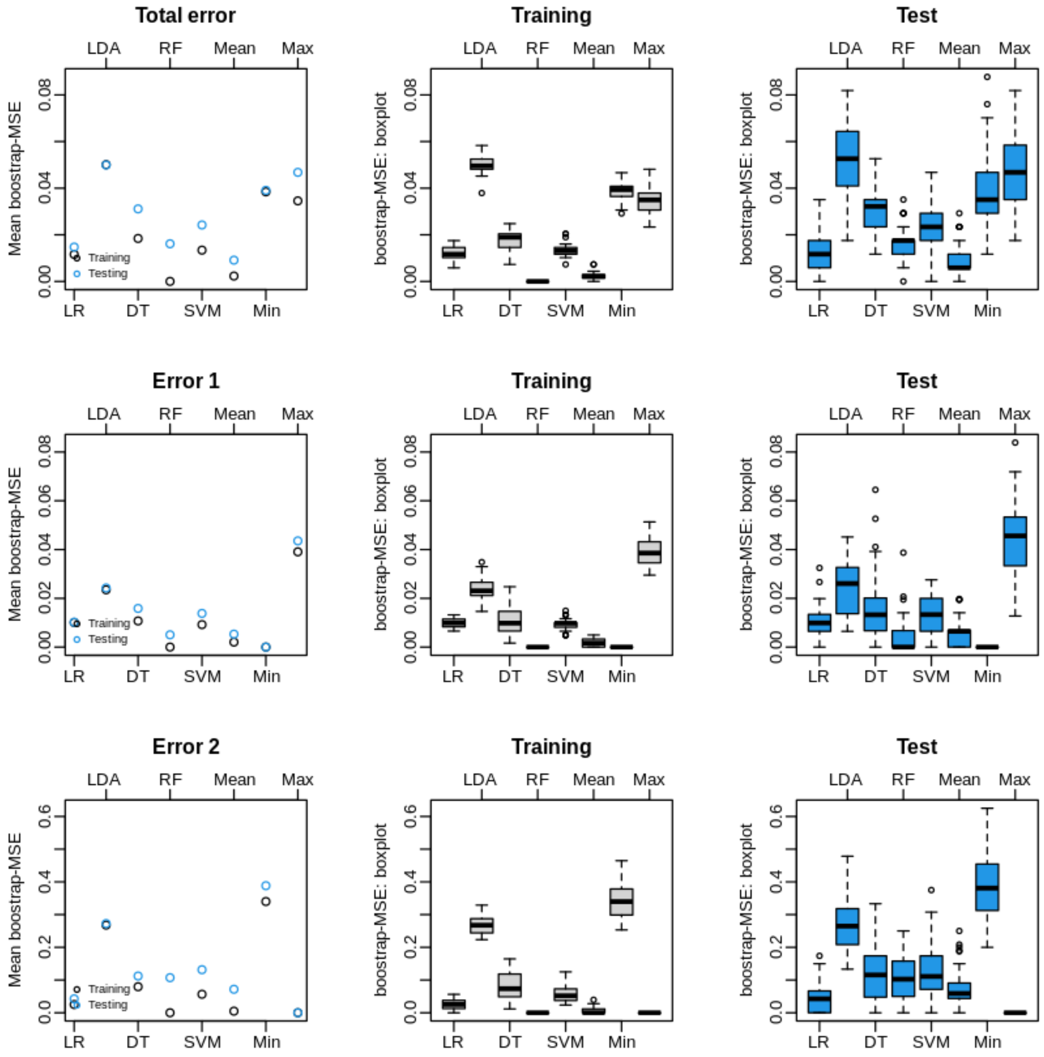


Figure 37– Results according to OSF with 500 operational machines and 50 repetitions

In the three intermediary graphs presented, RF and the minimum values of the algorithms nearly reach prediction scores close to perfection, mirroring the initial two scenarios observed during training. Despite a notable increase in the MSE value compared to Figure 25, very similar results were obtained compared to the situation depicted in Figure 31 during Training. Although considerably higher MSE values are evident in the Test case relative to Figure 25, similar results have also been observed. When comparing Figure 31 with the Test Case, it is challenging to comment on the trend within the range between the minimum and maximum quartiles of the box plots due to the presence of different scenarios.

In the final three visuals, a significant reduction in MSE values is observed in both the Training and Test graphs when compared to Figure 25. Conversely, there are MSE values that closely resemble those depicted in Figure 31. While the variability between the minimum and maximum quartiles of the box plots observed in Figure 31 remains consistent for numerous algorithms in the Test scenario.

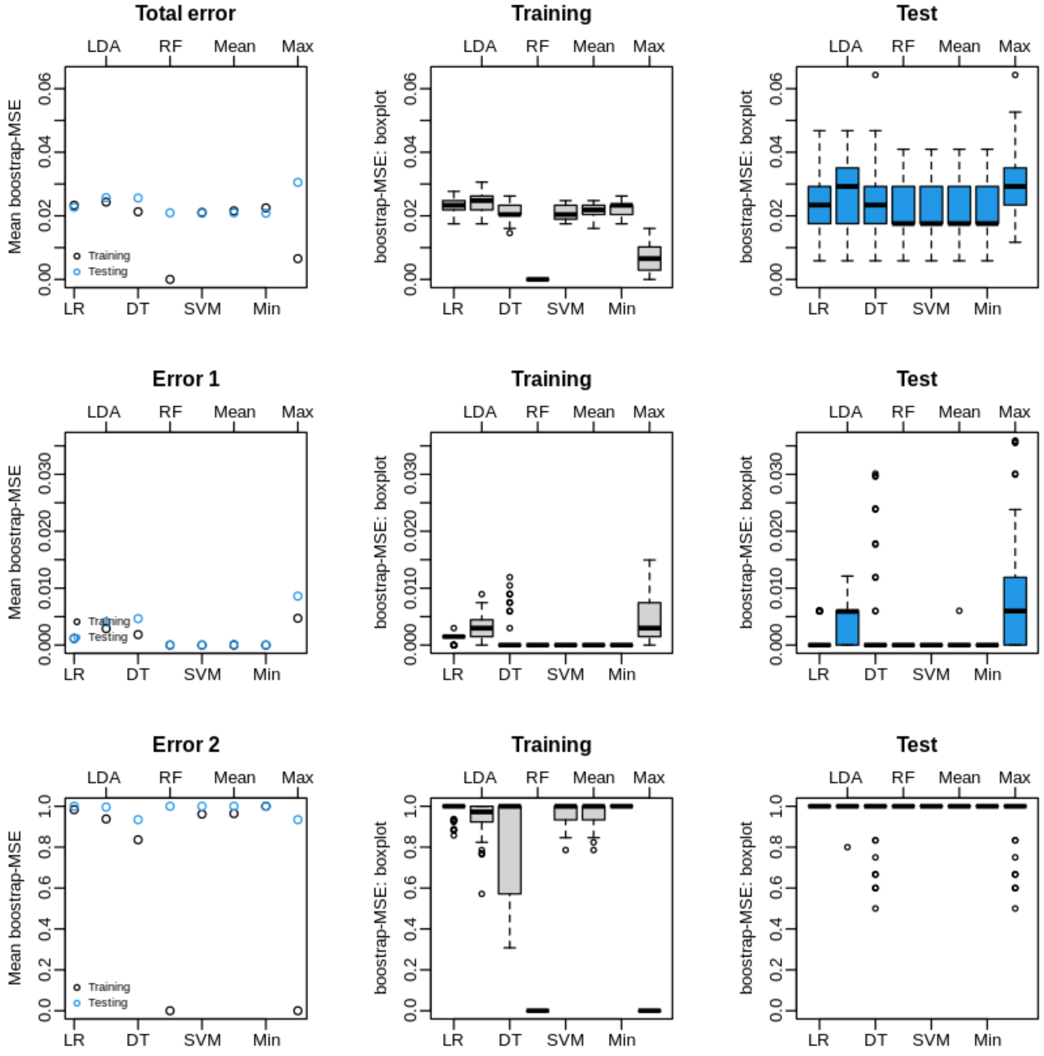


Figure 38– Results according to RNF with 500 operational machines and 50 repetitions

Figure 38 illustrates the outcomes generated by implementing the algorithm across RNF. In contrast to Figure 26, a clear increase in Mean Squared Error (MSE) values was noted for each algorithm in the initial three plots of the Training scenario, except for RF. Furthermore, this notable elevation in the MSE value is apparent across all algorithms in the Test case. Comparable findings were observed in both the Training and Test scenarios depicted in Figure 32, mirroring those in Figure 38. In the test graph, only minor variations were detected in the LDA and DT algorithms.

Performing a detailed analysis for the middle graphs and the last three graphs presents difficulties due to the limited data available in the initial scenario. Drawing conclusions from the Training and Test graphs in Figure 26 is not feasible. Specifically, in the Training graph concerning operational machines in the middle section, minimal MSE values were observed for the LDA and the majority of algorithms, a pattern also evident in the Test case as depicted in Figure 32. Although some minor changes are observed in the last three graphs, the overall situation closely resembles that of Figure 32. In summary, the scarcity of data points makes it challenging to conduct a precise analysis for these three cases.

4.4 Evaluation, comparison, and interpretation of the results

In Section 4.3 displays various machine conditions, including operational and failed machines, for different machine learning algorithms using boxplot graphics generated with R. These conditions encompass both training and testing scenarios, as well as their simultaneous occurrence.

This section utilizes machine failure data encompassing the entire dataset to analyze and interpret the results obtained in the preceding section. A comparison is drawn between these results. The outcomes are depicted through line graphs, separately illustrating Training and Testing cases. They are categorized based on Total error, Error 1, and Error 2, presented across six graphs labeled as Figure 39 through Figure 44.

During the graph preparation process, the three leftmost graphs were constructed, encompassing all three machine states. These graphs display both Training and Testing results derived from the outcome graphs in the previous section. They are based on the average Mean Squared Error (MSE) value for all algorithms. To facilitate comparison, the data is organized into nine distinct cases, each characterized by specific features:

Case 1: Total Error, 500 operational machines, 10 repetitions

Case 2: Error 1, 500 operational machines, 10 repetitions

Case 3: Error 2, 500 operational machines, 10 repetitions

Case 4: Total Error, 500 operational machines, 50 repetitions

Case 5: Error 1, 500 operational machines, 50 repetitions

Case 6: Error 2, 500 operational machines, 50 repetitions

Case 7: Total Error, 9661 operational machines, 10 repetitions

Case 8: Error 1, 9661 operational machines, 10 repetitions

Case 9: Error 2, 9661 operational machines, 10 repetitions

Each graph compares three cases, delineated according to Total Error, Error 1, and Error 2. This arrangement allows for clear separation and comparison across the different error metrics.

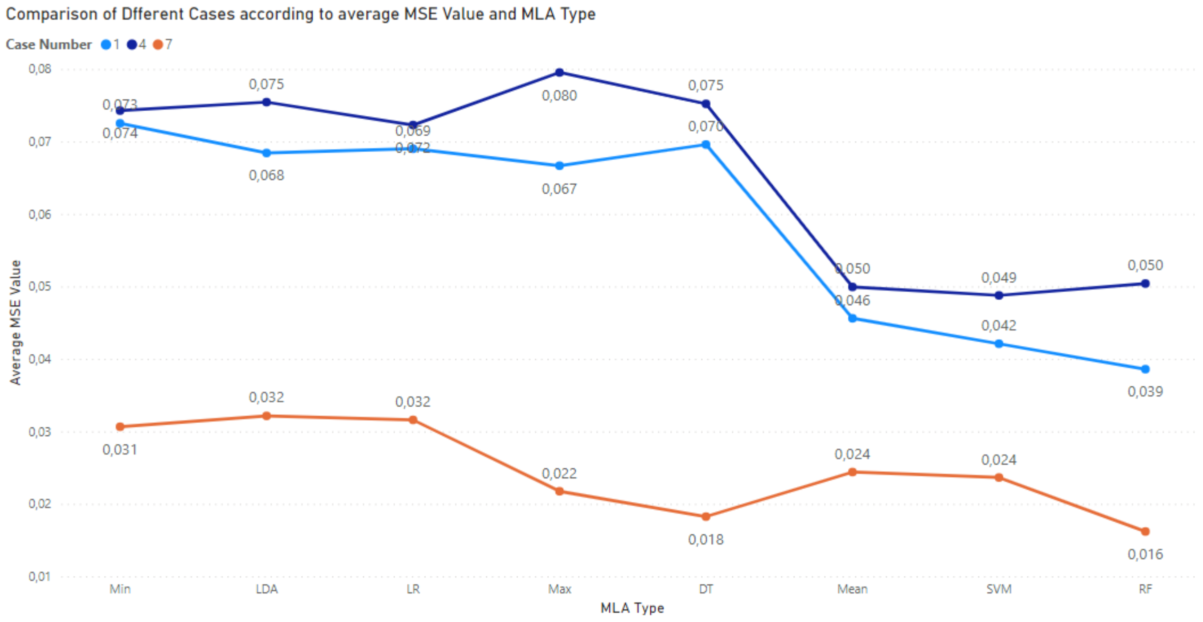


Figure 39– Comparison of average MSE values according to Total Error and Testing case

Figures 39 and 40 depict the outcomes derived from operational and failed machines. These graphs distinctly illustrate that the Mean Squared Error (MSE) value of the Random Forest algorithm notably surpasses that of other algorithms, yielding the most favorable results for this dataset. Following suit are SVM and the Mean of algorithms. Upon scrutinizing these graphs, it becomes evident that the number of repetitions doesn't exert a significant influence on the outcomes. However, a notable impact of the quantity of operational machines is observed. Furthermore, as the count of operational machines rises, the MSE values of the Decision Tree (DT) and Maximum of algorithms exhibit superior performance compared to SVM and the Mean algorithm.

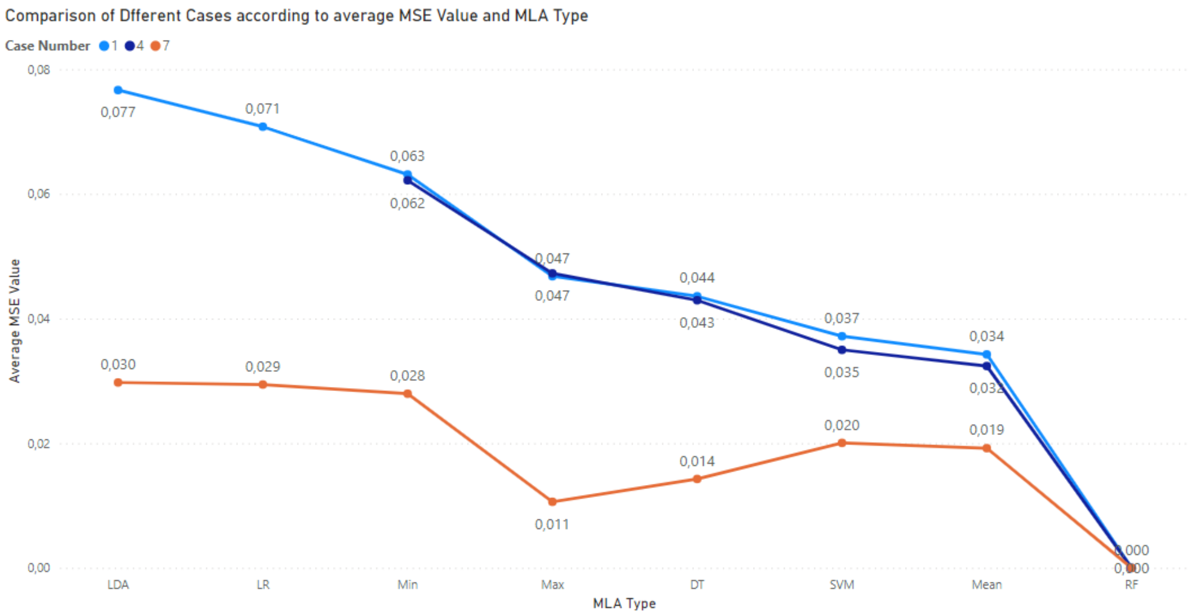


Figure 40– Comparison of average MSE values according to Total Error and Training case

Figures 41 and 42 illustrate the results obtained from operational machines. As anticipated, the Mean Squared Error (MSE) values are notably low, given that these graphs specifically represent outcomes from operational machines. Notwithstanding this circumstance, upon interpretation of these graphs, it can be inferred that the Minimum algorithm value yields the most favorable result for this particular case.

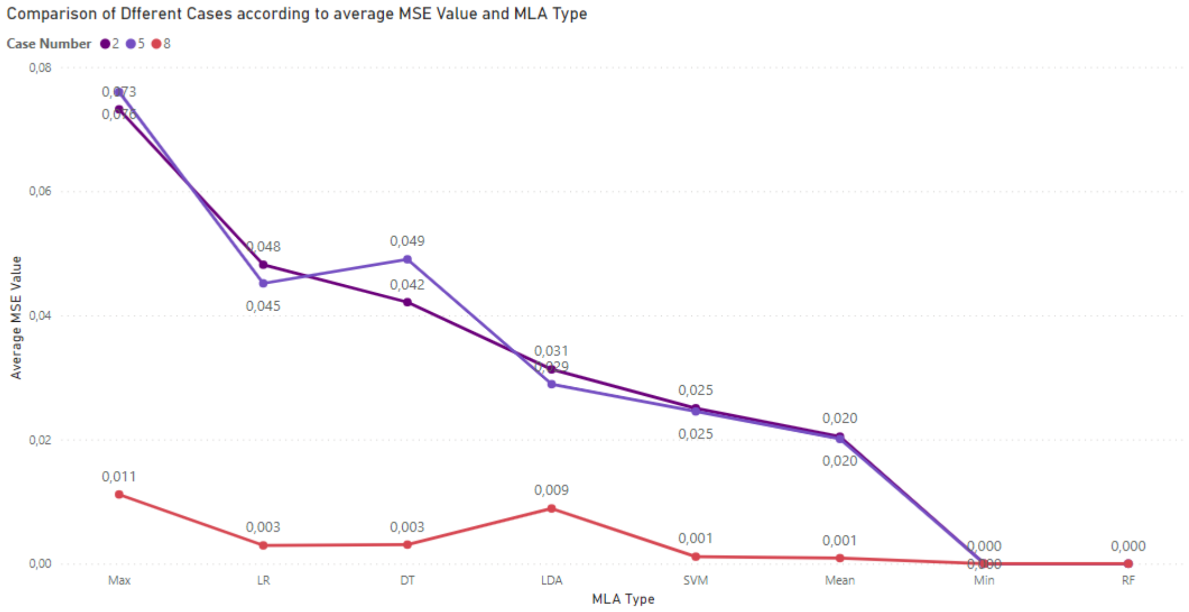


Figure 41– Comparison of average MSE values according to Error 1 and Training case

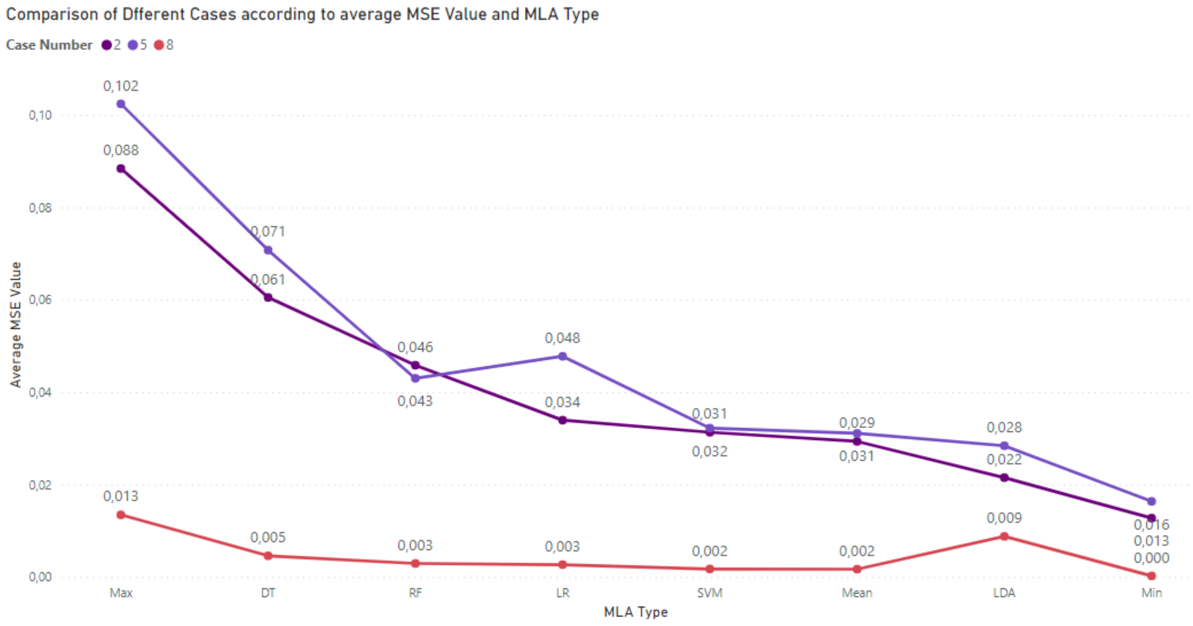


Figure 42– Comparison of average MSE values according to Error 1 and Testing case

Figures 43 and 44 display the results obtained from failed machines. As anticipated, the Mean Squared Error (MSE) values for certain algorithms are notably high, given that these graphs specifically represent the outcomes of failed machines. Despite this expected outcome, upon analysis of these graphs, it can be deduced that the Maximum of algorithms produces the most favorable result for this specific case. The Random Forest algorithm also demonstrates comparable results in this regard.

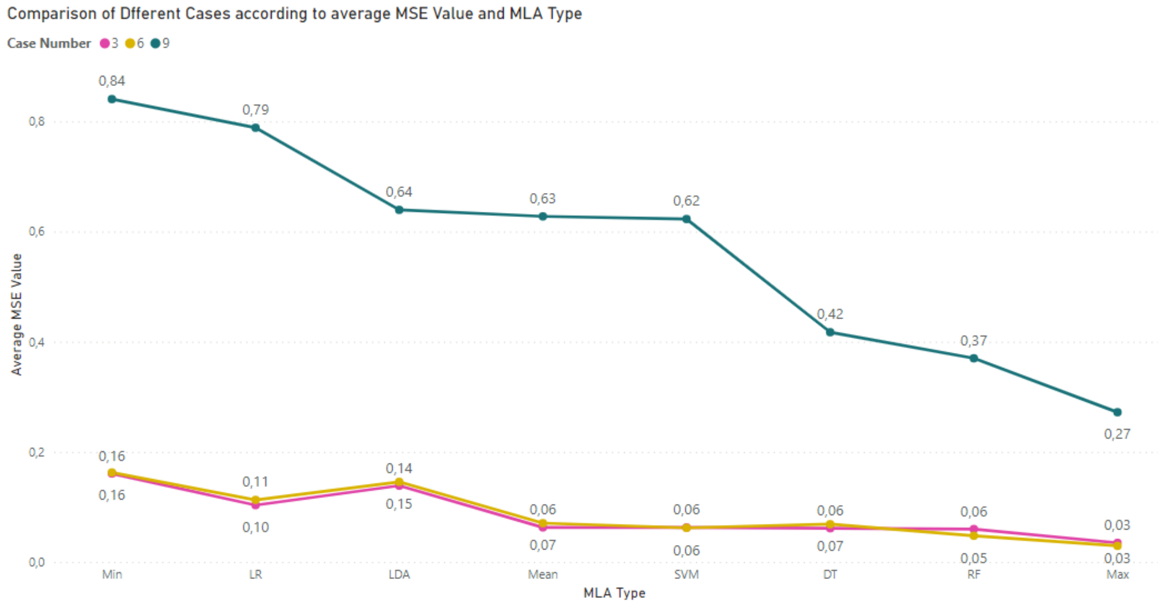


Figure 43– Comparison of average MSE values according to Error 2 and Testing case

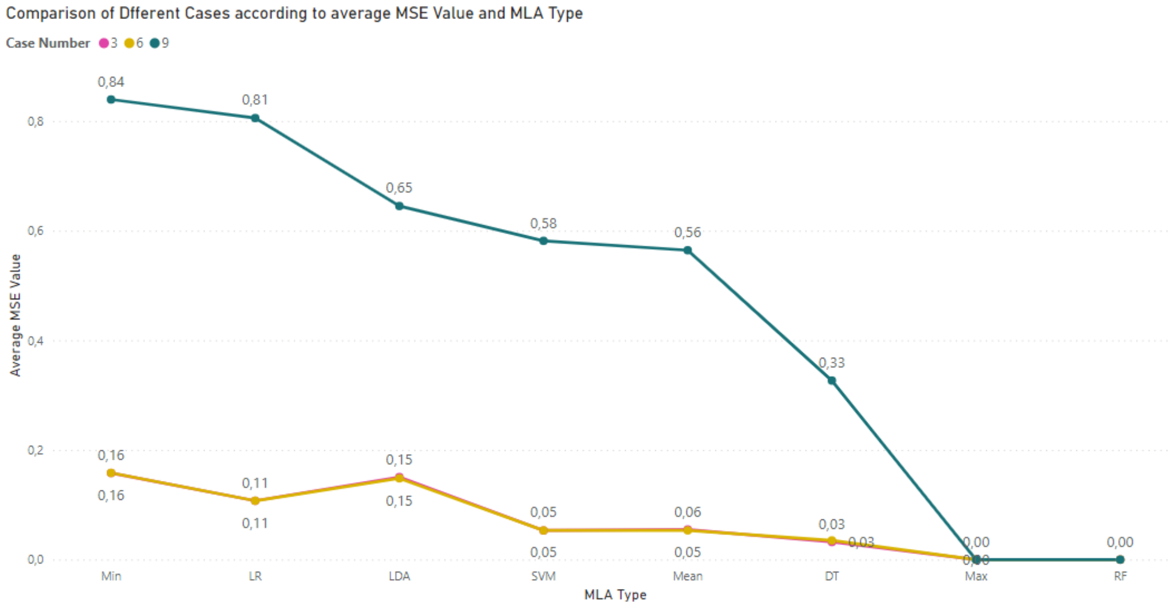


Figure 44– Comparison of average MSE values according to Error 2 and Training case

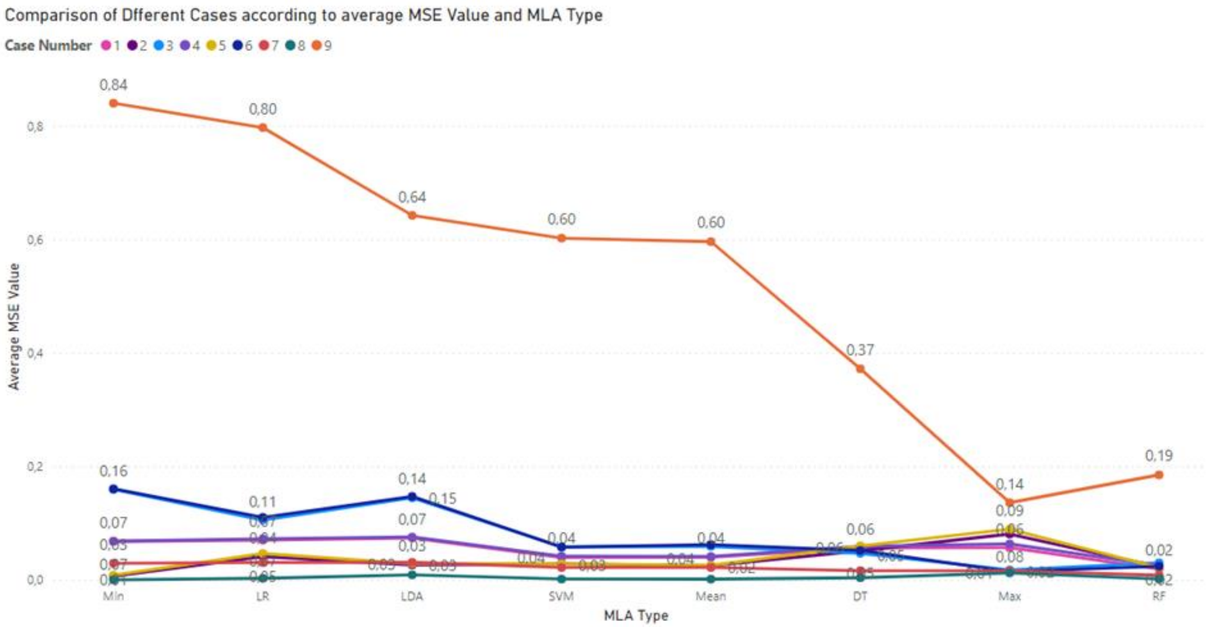


Figure 45– Comparison of average MSE values for all cases

In Figure 45, the results from all cases are compared without discrimination. It is evident that the Random Forest algorithm yields the most suitable results for this dataset and the given parameters. The Maximum of algorithm and Decision Tree algorithms also exhibit favorable performance. Conversely, the Minimum of Algorithm, LR, and LDA algorithms demonstrate the highest MSE values.

These findings suggest that the dataset comprises nonlinear data. Linear algorithms performed poorly in the analysis. Notably, the Minimum, Mean, and Maximum values of the function derived through the consensus ensemble method did not yield substantially different results.

5 Conclusion

The rise of Industry 4.0 marks a new era in manufacturing characterized by interconnected cyber-physical systems, leading to unprecedented levels of efficiency and productivity. Core elements of this shift include Cloud Manufacturing, enabling decentralized production and real-time data sharing, alongside the integration of IoT devices facilitating seamless communication and control across the manufacturing environment. A significant advantage of Industry 4.0 lies in leveraging Big Data to gain valuable insights from vast amounts of information generated within the production process. This data forms the basis for advanced analytics and decision-making, driving enhancements in performance, quality, and cost-effectiveness.

Reliability Engineering within Industry 4.0 plays an important role in ensuring the strength and resilience of cyber-physical systems, minimizing downtime, and maximizing uptime. Predictive Maintenance stands out as a cornerstone, utilizing advanced technologies and methodologies to anticipate equipment failures proactively. Through methods such as Condition-Based Monitoring and statistical techniques, manufacturers can identify potential issues beforehand, avoiding costly unplanned downtime and disruptions to production.

Machine Learning techniques further bolster the predictive maintenance capabilities of Industry 4.0, enabling systems to learn and adapt based on historical data patterns and real-time sensor readings. Continuous analysis and refinement of predictive models empower manufacturers to optimize maintenance schedules, prolong equipment lifespans, and ultimately enhance efficiency and competitiveness.

In the thesis's, sensor data with diverse parameters undergoes scrutiny using the R programming language to evaluate and compare various predictive maintenance methods. To accomplish this classification, an array of statistical methods and machine learning algorithms are employed, comprising Logistic Regression, Linear Discriminant Analysis, Decision Trees, Random Forests, Support Vector Machines, and Consensus functions like Minimum, Maximum, and Mean values.

Upon analyzing all the findings, it becomes apparent that the most suitable algorithm for this dataset is Random Forest, followed by the Maximum of algorithm and Decision Tree algorithms. Conversely, the three algorithms showing the weakest results are Minimum of Algorithm, LR, and LDA.

These results imply that the dataset contains nonlinear data, as indicated by the superior performance of Random Forest and other nonlinear algorithms. Additionally, it seems that the function values obtained through the consensus ensemble method do not significantly influence the overall outcome.

As the Industry 4.0 era is entered, the integration of Reliability Engineering and Predictive Maintenance becomes increasingly crucial for enhancing industrial processes' efficiency. A comparative examination centered on failure analysis and evaluation utilizing machine learning methods will remain pivotal in enhancing operational effectiveness, reducing downtime, and curtailing maintenance expenses across diverse sectors. With advancements in data analytics and machine learning algorithms, we anticipate the emergence of more advanced predictive maintenance approaches, enabling the proactive identification and mitigation of equipment failures prior to their occurrence. This realm of investigation will play a significant role in shaping the future of industrial maintenance methodologies by fostering a transition towards predictive and prescriptive maintenance models to align with the evolving needs of contemporary manufacturing environments.

List of Literature

[1] Krupitzer, C.; Wagenhals, T.; Züfle, M.; Lesch, V.; Schäfer, D.; Mozaffarin, A.; Edinger, J.; Becker, C.; Kounev, S. (2020): A Survey on Predictive Maintenance for Industry 4.0. arXiv: 2002.08224

[2] Francks, P. (2021): The Reconceptualisation of the Industrial Revolution and Why It Matters for Japanese Studies, Vol. 0, No. 0, 1–25

[3] Popkova, E. G., Bogoviz, A. V. & Ragulina,Y.V. :Industry 4.0: Industrial Revolution of the 21st Century, Springer Press, 2019, pp. 21-29.

[4] Groumpos, P. P. (2021): A Critical Historical and Scientific Overview of all Industrial Revolutions, IFAC PapersOnLine Volume 54, Issue 13, 2021, p.p 464-471

[5] Kayembe C. & Nel D.: Challenges and Opportunities for Education in the Fourth Industrial Revolution, Volume 11, No.3,2019

[6] Alguliyev R., Imamverdiyev Y., Sukhostat L.: Cyber-physical systems and their security issues, Computers in Industry, Volume 100, 2018, p.p 212-223

[7] Wang L., Wang X.V.: Cloud-Based Cyber-Physical Systems in Manufacturing, Springer Press, London, 2018

[8] Sobhrajan P., Nikam S.Y.: Comparative Study of Abstraction in Cyber Physical System, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (1), 2014, p.p.466-469

[9] Li B.H., Zhang L., Ren L., Chai X.D., Tao F. et al.: Typical characteristics, technologies, and applications of cloud manufacturing. Comput. Integr. Manuf. Syst. 18(7),2012, pp.1345–1356

[10] Wang L., Wang X.V., Gao L., Váncza J.: A cloud-based approach for WEEE remanufacturing, CIRP Ann. Manufact. Technol. Volume 63(1), 2014, p.p.409–412

[11] Wang L.: Machine availability monitoring and machining process planning towards Cloud manufacturing, CIRP J. Manufact. Sci. Technol. 6(4), 2013, pp. 263–273

[12] Yan J., Yue M., Lei L., Lin L.: Industrial Big Data in an Industry 4.0 Environment: Challenges, Schemes, and Applications for Predictive Maintenance, IEEE access, 2017

[13] Reis M.S., Gins G.: Industrial Process Monitoring in the Big Data/Industry 4.0 Era: from Detection, to Diagnosis, to Prognosis, MDPI, 2017

[14] Manavalan E., Jayakrishna K.: A review of Internet of Things (IoT) embedded sustainable supply chain for industry 4.0 requirements, Elsevier, 2019, p.p. 925-953

[15] Lampropoulos G., Siakas K., Anastasiadis T.: Internet of Things in The Context of Industry 4.0: An Overview, Issue 1/2019, Volume 7

[16] Sensor: Available online at https://www.techtarget.com/whatis/definition/sensor, last reviewed on 10.12.2023

[17] Javaid, M., Haleem, A., Singh, R. P., Rab, S., Suman, R.: Significance of sensors for industry 4.0: Roles, capabilities, and applications. In: Sensors International, 2 (2021) 100110

[18] Hoffmann Souza, M. L., Da Costa, C. A., Oliveira Ramos, G. de, Da Rosa Righi, R. (2020): A survey on decision-making based on system reliability in the context of Industry 4.0. In: Journal of Manufacturing Systems, 56, p.p. 133–156.

[19] Baro-Tijerina, M., Piña-Monarrez, M.R.; Molina Arredondo, R.D. (2021): Reliability Engineering in Industry 4.0. In: Critical Factors in Industry 4.0, p.p. 73-93.

[20] Kiran D.R.: Reliability Engineering. In: Total Quality Management, (2017)

[21] Farsi, M.A., Zio, E.: Industry 4.0: Some Challenges and Opportunities for Reliability Engineering, Vol. 2/ Issue 1/ 2019, p.p. 23-34

[22] Carvalho, T. P., Soares, F. A. A. M. N., Vita, R., Da Francisco, R. P., Basto, J. P., Alcalá, S. G. S.: A systematic literature review of machine learning methods applied to predictive maintenance. In: Computers & Industrial Engineering, 137 (2019) 106024

[23] Shukla, K., Nefti-Meziani S., Davis S.: A heuristic approach on predictive maintenance techniques: Limitations and scope. In: Advances in Mechanical Engineering 2022, Vol. 14(6) 1–14

[24] Endrenyi, J., Aboresheid, S., Allan R.N., et al.: The Present Status of Maintenance Strategies and the Impact of Maintenance on Reliability. In: IEEE Transactions on Power Systems,2001, Vol. 16, No. 4

[25] Bink, R.; Zschech, P. (2018): Predictive Maintenance in der industriellen Praxis. In: HMD-Praxis der Wirtschaftsinformatik, 55 (3), p.p. 552–565

[26] Merkt, O. (2019): On the Use of Predictive Models for Improving the Quality of Industrial Maintenance: an Analytical Literature Review of Maintenance Strategies. In: Proceedings of the Federated Conference on Computer Science and Information Systems, Vol. 18, p.p. 693–704

[27] Krupitzera, C., Wagenhals, T., Züfle M., et al. (2020): A Survey on Predictive Maintenance for Industry 4.0.

[28] Vanli, O. A. (2014): A Failure Time Prediction Method for Condition-Based Maintenance. In: Quality Engineering, 26:335–349

[29] Zhao, S., Yang, F., Ugur E., et al. (2021): A Composite Failure Precursor for Condition Monitoring and Remaining Useful Life Prediction of Discrete Power Devices. In: IEEE Transactions on Industrial Informatics, Vol. 17, No. 1

[30] Cakir, M.; Guvenc, M. A.; Mistikoglu, S. (2021): The experimental application of popular machine learning algorithms on predictive maintenance and the design of IIoT based condition monitoring system. In: Computers & Industrial Engineering, 151 (2021) 106948.

[31] Zhang, W.; Yang D.; Wang, H. (2019): Data-Driven Methods for Predictive Maintenance of Industrial Equipment: A Survey. In: IEEE Systems Journal, Vol. 13, No. 3, September 2019

[32] What is Statistical Modeling?: Available online at https://www.simplilearn.com/tutorials/statistics-tutorial/what-is-statistical-modeling#:~:text=Statistical%20modeling%20is%20an%20elaborate,random%20variables%20in%20this%20process., last reviewed on 25.12.2023

[33] Statistical Modeling: Available online at https://www.heavy.ai/technical-glossary/statistical-modeling, last reviewed on 25.12.2023.

[34] What is a Decision Tree?: Available online at https://www.ibm.com/topics/decision-trees, last reviewed on 25.12.2023.

[35] Jijo, B. T.; Abdulazeez A. M.: Classification Based on Decision Tree Algorithm for Machine Learning. In: Journal of Applied Science and Technology Trends Vol. 02, No. 01, pp. 20 – 28 (2021)

[36] Nick, T. G.; Campbell K. M. (2007): Logistic Regression. In: Topics in Biostatistics

[37] What is logistic regression?: Available online at https://www.ibm.com/topics/decision-trees, last reviewed on 25.12.2023.

[38] Çınar, Z. M., Nuhu, A. A., Zeeshan Q., et al. (2020): Machine Learning in Predictive Maintenance towards Sustainable Smart Manufacturing in Industry 4.0. In: Sustainability 2020, 12, 8211

[39] What is linear discriminant analysis?: Available online at https://www.ibm.com/topics/linear-discriminant-

analysis#:~:text=Linear%20discriminant%20analysis%20(LDA)%20is,helps%20optimize%20machine%20learning%20models., last reviewed on 26.12.2023.

[40] Linear Discriminant Analysis (LDA) in Machine Learning: Example, Concept and Applications: Available online at https://medium.com/aimonks/linear-discriminant-analysis-lda-in-machine-learning-example-concept-and-applications-37f27e7c7e98, last reviewed on 26.12.2023.

[41] Matzka, S. (2021): Künstliche Intelligenz in den Ingenieurwissenschaften, Springer Vieweg

[42] Lubis, A. R., Lubis, M., Al-Khowarizmi: Optimization of distance formula in K-Nearest Neighbor method. In: Bulletin of Electrical Engineering and Informatics, Vol. 9, No. 1, February 2020, pp. 326~338

[43] Ouadah, A., Zemmouchi-Ghomari, L., Salhi N.: Selecting an appropriate supervised machine learning algorithm for predictive maintenance. In: The International Journal of Advanced Manufacturing Technology (2022) 119:4277–4301

[44] support vector machine (SVM): Available online at https://www.techtarget.com/whatis/definition/support-vector-machine-SVM, last reviewed on 27.12.2023.

[45] Support Vector Machine Algorithm: Available online at https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm, last reviewed on 27.12.2023.

[46] Random Forest Algorithm: Available online at https://www.javatpoint.com/machine-learning-random-forest-algorithm, last reviewed on 27.12.2023.

[47] What are neural networks?: Available online at https://www.ibm.com/topics/neural-networks, last reviewed on 29.12.2023.

[48] Artificial Neural Network Tutorial: Available online at https://www.javatpoint.com/artificial-neural-network, last reviewed on 29.12.2023.

[49] Xu, Z., Saleh, J. H.: Machine learning for reliability engineering and safety applications: Review of current status and future opportunities. In: Reliability Engineering and System Safety 211 (2021) 107530

[50] Matzka S.: Explainable Artificial Intelligence for Predictive Maintenance Applications. In: 2020 Third International Conference on Artificial Intelligence for Industries (AI4I), 2020, pp. 69-74, doi: 10.1109/AI4I49448.2020.00023.

[51] Peng R.D.: R Programming for Data Science. In: Leanpub (2016)

[52] Available CRAN Packages By Name: Available online at https://cran.r-project.org/web/packages/available_packages_by_name.html, last reviewed on 28.01.2024.

[53] Gogtay, N.J., Thatte, U. M.: Principles of Correlation Analysis. In: Journal of The Association of Physicians of India, Vol. 65, (2017)

[54] Ringnér, M.: What is principal component analysis?. In: Nature biotechnology, Vol. 26, No.3 (2008)

[55] Mean Squared Error (MSE): Available online at https://statisticsbyjim.com/regression/mean-squared-error-mse/, last reviewed on 13.02.2024.

[56] Alzubi, O., Alzubi, A., Tedmori S., et al. (2018): Consensus-Based Combining Method for Classifier Ensembles. In: The International Arab Journal of Information Technology, Vol. 15, No. 1, January 2018

Appendix

Appendix A and B

```
## Install the packages (it may take a few minutes)

install.packages('neuralnet');require(neuralnet)
install.packages('MASS');require(MASS)
install.packages('corrplot');require(corrplot)
install.packages('rpart');require(rpart)
install.packages('rpart.plot');require(rpart.plot)
install.packages('e1071');require(e1071)
install.packages('randomForest');require(randomForest)
install.packages('LPCM');require(LPCM)
install.packages('GGally');library(GGally)
```

```
## Download and read the data

download.file('https://raw.githubusercontent.com/antoinetordeux/Dat
asets/main/ai4i2020.csv','/content/ai4i2020.csv')
data=read.csv("ai4i2020.csv",header=TRUE)
names(data)
summary(data)
Y=cbind(data[,9:14])==1
X=data[,4:8]
## Plot the results
for(i in 1:5)
  hist(X[,i],main=names(X)[i])
for(i in 1:5)
  hist(as.numeric(Y[,i]),main=names(Y)[i])


sum(YY[,6])
```

Appendix A– Preparation of data for analysis

```
## Correlation analysis

cor(cbind(X,Y))
corrplot(cor(cbind(X,Y)))

## Principal component analysis

PCA=prcomp(X,scale.=T)
summary(PCA)
plot(PCA)
PC=as.data.frame(PCA$x)
cor(X,PC)
corrplot(cor(cbind(PC,Y)))
```

```
plot_circle_correlation=function(a,b){
  plot(cos(seq(0,2*pi,.01)),sin(seq(0,2*pi,.01)),xlab=paste("Princi
pal Composant",a),ylab=paste("Principal
Component",b),type='l',xlim=c(-1,1),ylim=c(-1,1))
  for(i in 1:length(cor(X,PC)[,b])){
    segments(0,0,cor(X,PC)[i,a],cor(X,PC)[i,b],col=i);
    lines(cor(X,PC)[i,a],cor(X,PC)[i,b],type='p',pch=i,col=i)}
    legend("topleft",names(X),pch=1:ncol(X),col=1:ncol(X),cex=.8,bt
y='n')}

plot_component=function(a,b,k){
  plot(PC[,a],PC[,b],xlab=paste("PC",a),ylab=paste("PC",b),pch=1+as
.integer(Y[,k]),col=rgb(as.integer(Y[,k]),0,1-
as.integer(Y[,k]),.2))
  lines(c(mean(PC[!Y[,k],a]),mean(PC[Y[,k],a])),c(mean(PC[!Y[,k],b]
),mean(PC[Y[,k],b])),type='p',pch=c(16,17),col=c(4,2))
  legend("topleft",legend=k,title="Failure",cex=.8,bty='n')}


par(mfrow=c(2,2),mar=c(4,4,4,4))
for(pc in 2:5)
  plot_circle_correlation(1,pc)

par(mfrow=c(2,2),mar=c(3,3,2,2),mgp=c(1.5,.5,0))
for(k in 1:6){
  plot_component(1,2,k)
  plot_component(1,3,k)
  }
```

Appendix B– Correlation analysis and principal component analysis

Appendix C

```
## Classification

MSE=function(M,Y,k){
  mse=mean((M-Y)^2)
  if(k==2) mse=mean((M[!Y]-Y[!Y])^2)
  if(k==3) mse=mean((M[Y]-Y[Y])^2)
  mse}

algo_name=function(){
  axis(1,at=c(1,3,5,7),lab=c("LR","DT","SVM","Min"))
  axis(3,at=c(2,4,6,8),lab=c("LDA ","RF","Mean","Max"))}

B=10

train=matrix(0,B,8)
test=matrix(0,B,8)

plotPred=function(Y){
  par(mfrow=c(3,3),mar=c(3,4,4,2),mgp=c(2,.5,0))
  options(warn=-1)
  for(err in 1:3){
    for(b in 1:B){

      ## Cross-validation sampling
      cc=NULL;n=nrow(X)
      cc[1:n]=T;cc[sample(1:n,.2*n)]=F

      ## Logistic Regression
      algo_REG=glm(Y[cc]~.,data=PC[cc,],family=binomial(logit))
      LRtrain=predict(algo_REG,PC[cc,])>0
      LRtest=predict(algo_REG,PC[!cc,])>0
      train[b,1]=MSE(LRtrain,Y[cc],err)
      test[b,1]=MSE(LRtest,Y[!cc],err)

      ## Linear Discriminant Analysis
      algo_LDA=lda(Y[cc]~.,data=PC[cc,])
      LDAtrain=as.numeric(predict(algo_LDA,PC[cc,])$class)-1
      LDAtest=as.numeric(predict(algo_LDA,PC[!cc,])$class)-1
      train[b,2]=MSE(LDAtrain,Y[cc],err)
      test[b,2]=MSE(LDAtest,Y[!cc],err)

      ## Decision Tree
      algo_DT=rpart(Y[cc]~.,data=X[cc,],method="class")
      DTtrain=as.numeric(predict(algo_DT,X[cc,],type="class"))-1
      DTtest=as.numeric(predict(algo_DT,X[!cc,],type="class"))-1
      train[b,3]=MSE(DTtrain,Y[cc],err)
      test[b,3]=MSE(DTtest,Y[!cc],err)

      ## Random Forest
      algo_RF=randomForest(as.factor(Y[cc])~.,data=X[cc,])
      RFtrain=as.numeric(predict(algo_RF,X[cc,]))-1
      RFtest=as.numeric(predict(algo_RF,X[!cc,]))-1
      train[b,4]=MSE(RFtrain,Y[cc],err)
      test[b,4]=MSE(RFtest,Y[!cc],err)

      ## Support Vector Machine
      algo_SVM=svm(as.factor(Y[cc])~.,data=PC[cc,])
      SVMtrain=as.numeric(predict(algo_SVM,PC[cc,]))-1
      SVMtest=as.numeric(predict(algo_SVM,PC[!cc,]))-1
      train[b,5]=MSE(SVMtrain,Y[cc],err)
      test[b,5]=MSE(SVMtest,Y[!cc],err)

      ## Mean value of the algorithms
      MEANtrain=round(apply(cbind(LRtrain,LDAtrain,DTtrain,RFtrain,SVMtrain),1,mean))
      MEANtest=round(apply(cbind(LRtest,LDAtest,DTtest,RFtest,SVMtest),1,mean))
      train[b,6]=MSE(MEANtrain,Y[cc],err)
      test[b,6]=MSE(MEANtest,Y[!cc],err)

      ## Min value of the algorithms
      MINtrain=apply(cbind(LRtrain,LDAtrain,DTtrain,RFtrain,SVMtrain),1,min)
      MINtest=apply(cbind(LRtest,LDAtest,DTtest,RFtest,SVMtest),1,min)
      train[b,7]=MSE(MINtrain,Y[cc],err)
      test[b,7]=MSE(MINtest,Y[!cc],err)

      ## Max value of the algorithms
      MAXtrain=apply(cbind(LRtrain,LDAtrain,DTtrain,RFtrain,SVMtrain),1,max)
      MAXtest=apply(cbind(LRtest,LDAtest,DTtest,RFtest,SVMtest),1,max)
      train[b,8]=MSE(MAXtrain,Y[cc],err)
      test[b,8]=MSE(MAXtest,Y[!cc],err)
    }

    yl=range(train,test)
    m="Total error\n";if(err==2) m="Error 1\n";if(err==3) m="Error 2\n"
    plot(apply(train,2,mean),xlab="",ylab="Mean boostrap-MSE",xaxt='n',ylim=yl,main=m);algo_name()
    lines(apply(test,2,mean),type='p',col=4)
    legend("bottomleft",c("Training","Testing"),pch=c(1,1),col=c(1,
4),cex=.7,bty='n')
    boxplot(train,xlab="",ylab="boostrap-MSE:
boxplot",xaxt='n',ylim=yl,main="Training\n");algo_name()
    boxplot(test,xlab="",ylab="boostrap-MSE:
boxplot",xaxt='n',ylim=yl,col=4,main="Test\n");algo_name()
  }
}
## Machine failure

plotPred(Y[,1])
sum(Y[,1])
## Failure 1

plotPred(Y[,2])
sum(Y[,2])

## Failure 2

plotPred(Y[,3])
sum(Y[,3])
## Failure 3

plotPred(Y[,4])
sum(Y[,4])
## Failure 4

plotPred(Y[,5])
sum(Y[,5])
## Failure 5

plotPred(Y[,6])
sum(Y[,6])
```

Appendix C– Classification of operating states using statistical methods and machine learning algorithms

Appendix D

```
## Balancing the data set (use of only 500 operational
observations)
XX=NULL
YY=NULL
XX=rbind(XX,X[Y[,1]|Y[,6],])
YY=rbind(YY,Y[Y[,1]|Y[,6],])

XX=rbind(XX,X[!(Y[,1]|Y[,6]),][1:500,])
YY=rbind(YY,Y[!(Y[,1]|Y[,6]),][1:500,])
X=XX
Y=YY
```

Appendix D– Setting the number of operational machines to 500