A Comprehensive IoT-Enabled Predictive Maintenance Framework
A Case Study of Predictive Maintenance for a Printing Machine

Hosseini, Susan

Award date:
2021

EINDHOVEN UNIVERSITY OF TECHNOLOGY

Department of Mathematics and Computer Science
and Department of Electrical Engineering

A Comprehensive IoT-Enabled Predictive Maintenance Framework

A Case Study of Predictive Maintenance for a Printing Machine

Susan Hosseini

Supervisors:
Dr. Majid Nabi (TU/e)
Bert de Swart (NTS)
Laurens van de Laar (NTS)

A dissertation submitted in partial fulfilment of the requirements for the degree of
Master of Science in Embedded Systems

Eindhoven, July 2021

Abstract

The development of smart manufacturing and Industry 4.0 has emphasized the utilization of intelligent manufacturing tools, techniques, and methods such as Predictive Maintenance (PdM). The predictive maintenance function facilitates the early detection of failure and errors in machinery before they reach critical phases. Proper maintenance keeps the life cycle cost down and guarantees proper operations and good order internal logistics. This dissertation proposes a predictive maintenance framework enabling organizations to proactively work against their upcoming maintenance task. Having predicted the failure occurrences with good accuracy allows companies to significantly save costs by, for example, enforcing a controlled system shutdown rather than an emergency shutdown. Our proposed framework has been designed based on the data-driven model, and, therefore, it can be perfectly realized in IoT-enabled smart companies, in which objects and devices are monitored and controlled using intelligent systems connecting to the Internet. Most of the existing work in the domain of predictive maintenance has significantly focused on developing various anomaly detection techniques through simulations. However, in this study, we aim to design an end-to-end predictive maintenance framework and further evaluate our proposed framework using real-world manufacturing data. The obtained results have demonstrated the effectiveness of our proposed framework.

First, we have conducted a Systematic Literature Review (SLR) on existing predictive maintenance research papers. We classified the algorithms and techniques addressed in these studies according to the multiple phases of an end-to-end predictive maintenance project. In the second step, considering SLR results, we propose a plan for designing the PdM framework. The main purpose of this dissertation is to suggest a general predictive maintenance framework that is highly applicable for different use cases. Consequently, we design an end-to-end predictive maintenance framework that covers all the phases of a predictive maintenance project discovered in the SLR section.

The proposed predictive maintenance framework consists of five layers: data acquisition, data preprocessing, predictive analytics, result evaluation, and decision making. Finally, the proposed framework has been implemented into a prototype and tested in an industrial use case. In order to evaluate our proposed framework, we implemented a real-life case study related to the maintenance of a pump installed in an industrial printing machine at NTS Group. In particular, we conducted several experiments to assess the impact of each block of the proposed framework.

In this case study, multiple supervised and semi-supervised learning techniques were employed. Particularly, three regressors were utilized: *Linear Regression*, *Generalized Linear Regression*, *Decision Tree* and three classifiers: *Logistic Regression* (LR), *Decision Tree* (DT), *Random Forest* (RF). For the test dataset, the classifiers performed quite well and obtained significant accuracy of 95%, 99%, and 98%, respectively, for LR, DT and RF. We have also utilized One-Class Support Vector Machine (OCSVM) as a semi-supervised learning algorithm which delivered up to 99% for all of our evaluation metrics. Finally, a technique called *Peak Detection* is used for failure detection, which is based on recognizing the number of peaks in the failure period. This technique also has a promising result to send an alarm at the right time.

Besides these Machine Learning (ML) algorithms, the preprocessing steps such as scaling, feature extraction, feature selection, and dimension reduction are analyzed to achieve better accuracy and fewer error values. In almost all test scenarios, time-domain feature extraction caused less accuracy and higher error due to the low sampling frequency of the collected data. Principle Component Analysis (PCA) and Pearson's Correlation Coefficient (PCC) techniques were analyzed. The investigations revealed that just using PCA or PCC techniques for dimension reduction and finding important features without considering data distribution can decrease the accuracy, or there is no point in using them, just wasting computation resources. Obtained results from multiple tested scenarios indicate the effectiveness of the proposed meth-

odology in supporting predictive analytics in the age of Industry 4.0.

Acknowledgement

Foremost, I would like to thank my supervisor at the Eindhoven University of Technology, Dr. Majid Nabi, for this scientific guidance, support, and encouragement in every step of this work. I am incredibly grateful for giving me the trust to work on this inspiring project. Furthermore, I am grateful to the NTS group for giving me the opportunity to do my research in such a professional and high-tech environment, and special thanks to Bert de Swart and Laurens van de Laar for their supervision, advice, and their valuable critiques during this project. Furthermore, at NTS, I enjoyed collaborating with Anetta, Sergey, Nobahar, and Bernard during some technical challenges.

I am very grateful to all my friends and their endless supports, including Yeshika, Sandeep, Niki, Milad, Nikhil, Mohmmad, Neda, Hamideh. I would like to express my deepest thanks to Shaya and Pegah for their continuous emotional support and inspiration. Thanks to both of you for encouraging me in every step and did not let me give up during this journey.

Most importantly, I would love to express my deepest appreciation and gratitude to my family: my parents, Fatemeh and Naser, and my dear brothers, Hasan and Morteza, who afforded me an inspiring environment that helped me grow and advance in different aspects of life. In addition, I am extremely grateful to my wonderful parents for their unconditional love and kindness, long-life support, and their patience in these years.

Last but certainly not least comes my lovely husband Amin, of whom I have had his patience, friendship, and understanding in happiness and sadness, endless/infinite support and encouragement. I would love to express my deepest gratitude for your everlasting love, motivation to have confidence in me, and for inspiring and helping me get through my Master's study.

Susan Hosseini
July 2021

Contents

List of Figures

List of Tables

List of Abbreviations

Enterprise Resource Planning (ERP)
Manufacturing Execution System (MES)
Internet of Things (IoT)
Predictive Maintenance (PdM)
Industrial Internet of Things (IIoT)
Machine Learning (ML)
Deep Learning (DL)
Support Vector Machine (SVM)
Random Forest (RF)
Logistic Regression (LR)
Decision Tree (DT)
Systematic Literature Review (SLR)
Building Block (BB)
K-Nearest Neighbourhood (KNN)
Root Mean Square (RMS)
Linear Discriminant Analysis (LDA)
Singular Value Decomposition (SVD)
Remaining Useful Life (RUL)
Artificial Neural Network (ANN)
Cross-Validation (CV)
True Positive (TP)
False Positive (FP)
True Negative (TN)
False Negative (FN)
Root Mean Square Error (RMSE)
Mean Absolute Error (MAE)
Key Performance Indicator (KPI)
Ink Supply Unit (ISU)
Head Supply Unit (HSU)
Principle Component (PC)
Generalized Linear Regression (GLR)
Auto Encoder (AE)
One-Class Classifier (OCC)
Overall Equipment Effectiveness (OEE)
Corrective Maintenance (CM)
Preventive Maintenance (PM)
Condition Based Monitoring (CBM)
Open Platform Communications United Architecture (OPC UA)
Message Queuing Telemetry Transport (MQTT)
Reference Architectural Model Industrie 4.0 (RAMI 4.0)
Digital Signal Processor (DSP)
Principle Component Analysis (PCA)
Pearson Correlation Coefficient (PCC)

Chapter 1

Introduction

Nowadays, the manufacturing industry is moving towards creating smart factories that are equipped with state-of-art technologies (such as 3D printing, advanced robotics) [1, 2]. The technology that enables this development is known as the fourth industrial revolution, Industry 4.0. This technology empowers manufacturers with offering new products and services to customers along with higher quality efficiency, and reliability [1, 3].

By utilizing smart devices, organizations can closely track their activities and, then, they have the opportunity to improve their business processes [1, 2, 4] by redesigning them. For example, one of the most critical business processes in the manufacturing domain is equipment maintenance [5]. Internet of Things (IoT) extends equipment maintenance business process to a higher level by introducing the notion of Predictive Maintenance (PdM). Many studies have supported this idea [1, 6, 7], in which the authors have suggested that ineffective maintenance approaches are one of the main reasons behind the rise of operating costs of investments. Accordingly, a predictive maintenance technology can reduce the resource waste due to unnecessary maintenance and eventually ensure that the equipment in a factory is well-maintained and remains in good condition.

Currently, predictive maintenance technology has focused on the healthy condition of hardware and software devices in a factory by employing remote tracking systems. Nevertheless, to adopt these advanced solutions, it is essential to broadly comprehend the trade-off between the advantages that the technology can bring versus the extra costs that are required during the deployment stage of the technology [1, 2, 4]. Since organizations need to purchase required equipment and instrumentation tools, software licenses, and acquiring special knowledge, extra costs will be there for enterprises. Therefore, finding a balance between all the acquiring costs and the competitive edges that predictive maintenance would offer can be seen as a real challenge that needs to be addressed. Thus, this challenge has raised the necessity for extra investigation not only in the industry but also in academia [1, 8, 9, 10]. To bridge this gap, research studies are constantly providing improvement opportunities to the industries in the domain of predictive maintenance. On the other hand, enterprises are continuously bringing new challenges to the researches.

There is another challenge that needs to be investigated, called the lack of initial data. At the beginning of predictive maintenance deployment, due to not having actual data regarding the normal and abnormal behavior devices, the technology might not provide the business added value in a concise term. Consequently, companies may distrust any further investment in the technology.

This work aims to propose a comprehensive predictive maintenance framework, considering deployment simplicity and fast-time-to-value requirements. This framework enables IoT-enabled factories to establish the best strategy to adopt predictive maintenance technology. This framework is evaluated by the NTS Group company located in Eindhoven, the Netherlands. NTS Group (in short, NTS) is a general machine manufacturer for semiconductors, digital printing, human healthcare, and the renewable energy industry, adopting Industry 4.0. NTS develops machines by integrating the required components for the desired application, assembling, and finally installing them for their customers. In addition, NTS provides platforms that can be used by many of its customers.

Moreover, NTS is responsible for maintaining their machines and provide operational services and supports to solve technical issues associated with their devices. One of the important products of NTS

is a platform of components to be used in Digital Printers. Since this platform's cost of production and maintenance is significantly high, NTS requires a predictive maintenance technology for this product. Therefore, the proposed predictive maintenance will be evaluated for the NTS digital printing machine. Due to unexpected expensive shutdown in the production line, one of the essential requirements for NTS is the *accuracy* of the proposed predictive maintenance framework. In addition, the growth of sensor data requires a more solid infrastructure for storing and processing. Therefore, the data management of the proposed framework in a cost-effective manner is also a crucial factor that has to be considered.

In summary, in this work, we aim to have an overview and perception of the predictive maintenance concept to develop a predictive maintenance framework based on Industry 4.0 in an accurate manner. To achieve this goal, we explored the current predictive maintenance frameworks, Industrial Internet of Things (IIoT) concepts, and the challenges they have faced to develop and implement an intelligence maintenance framework. Since this framework will be deployed on a digital printing machine in NTS Group company, we need to investigate the hardware and software architecture of the printing machine and the communication protocols employed in this machine. The next step is finding a critical component and defining a case study for applying PdM on it. For this challenge, we discussed with the experts and maintenance groups in NTS. Our research questions were formed in each step of exploration, and an overview of dissertation outlines has been provided.

This dissertation addresses our initial methodology to pursue research on developing a comprehensive predictive maintenance framework. The following section addresses the gap in the current body of knowledge and formulates the goal of our work and the research questions. According to the defined research questions, the work plan for the rest of this research study has been illustrated. In addition, we enumerated a few risks that we may encounter during this project. Chapter 2 presents the background and related studies that have been done in the domain of Industry 4.0 and predictive maintenance. Moreover, we investigated the case study's architecture and data life cycle considering the proposed predictive maintenance framework at NTS Group.

1.1 Introduction to NTS Group

NTS Group company develops, produces, assembles, and tests complex (opto-) mechatronic systems and mechanical modules, which helps accelerate its customers' innovations and contributes to a more sustainable, healthy, and future-proof world.

As a first-tier system supplier, NTS provides knowledge on production that enables cost-effective manufacturing. NTS is the support organization for selecting systems and modules in which precise motion and positioning are essential. NTS has vast knowledge and know-how of modules and systems for handling, transferring, and positioning machines. NTS focuses on high-tech original equipment manufacturers involved in markets with high levels of product variety, low volumes, and high complexity, such as life sciences, the semiconductor, and analytical and digital printing markets.

Industrial Automation at NTS

NTS builds machines by integrating the required components for the desired application and sell them to their customers. NTS is responsible for maintaining these machines and provide operational supports to solve technical issues associated with machines. These machines are used in the multi-level automation architecture. As it is illustrated in Figure 1.1, multi-level industrial automation architecture [11] comprises four levels:

- Device Level consists of multi-complex devices wired to actuators and sensors. All devices use Ethernet-based protocols such as EtherCAT and ModbusTCP to communicate with each other at this level.
- Control Level has programmable logic controllers or other control boards like DSPs that control the industrial process with the help of devices at the device level. These controllers use PROFINET, EtherCAT, or any other communication protocols for real-time control over Ethernet. To be more precise, controller devices have programs with inputs from the Device Level and take a control

action through program outputs communicated back to the Device Level. Furthermore, they provide information about the underlying process information to the Manufacturing Execution System (MES) level by using the OPC-UA protocol.

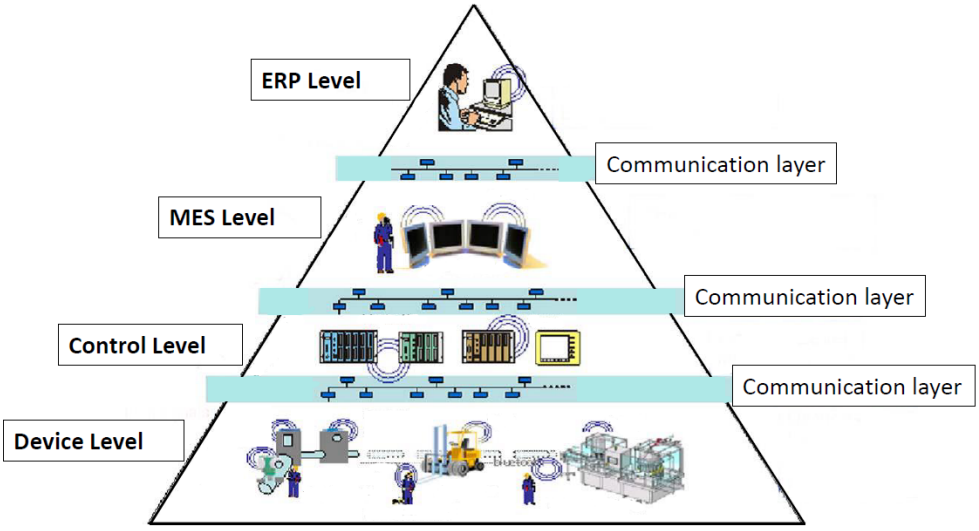

Figure 1.1: Multi-level automation pyramid of NTS [11]

- MES Level monitors and controls the whole industrial process. It means DSPs in the Control Level, which control individual tasks of device-level, will communicate with MES systems. This procedure enables complete control of industrial processing. Communication between the control level and MES level is based on the OPC-UA protocol. On the other side, communication between the MES level and Enterprise Resource Planning (ERP) level can be over the Internet.
- ERP Level provides a high-level overview and control over the business.

1.2 Motivation

The Industry 4.0 paradigm empowers companies to offer their products and services to their customers with higher efficiency, higher quality, and higher reliability. In order to transform a company into an Industry 4.0-compliant company, a combination of technologies need to be adopted. These technologies vary from IoT technology to data science and cloud computing. Altogether, the integration of these technologies empowers factories to reach the ultimate objective of digitization.

One of the main aspects of an Industry 4.0-compliant company is to adopt a predictive maintenance technology. This technology makes use of condition monitoring data that are produced by devices in smart factories. These data can be utilized to detect anomalies (i.e., the behaviors that are deviated from normal operating conditions) in manufacturing business processes. Therefore, predictive maintenance can be employed to detect failures in production processes and manufacturing equipment, products, and services.

In summary, predictive maintenance provides companies with the capability of performing maintenance tasks in a strict manner by making the right part available at the right place at the right time. Therefore, predictive maintenance tasks help companies to:

1. Deeply understand their asset performance patterns,
2. Raise alerts when an abnormal behavior has happened,
3. Proactively raise alerts when an abnormal behavior may happen or is about to happen,
4. Prevent costly downtime of devices, and

5. Eventually, maximize the business added value and production profits.

However, since there is no standard for adopting predictive maintenance, it can be a very complicated and costly process for early adopters of this technology. Therefore, in this work, we aim at addressing this challenge by proposing a predictive maintenance framework. Our proposed Industry 4.0-compliant framework is inspired by the 5C layered architecture [12]. Consequently, the main contribution of this dissertation is defined as the development of a predictive maintenance framework. A successful predictive maintenance framework can run a trade-off between improving the system reliability and reducing the total maintenance cost simultaneously.

NTS Group evaluates the result of this research as they are currently adopting a predictive maintenance technology considering the price of their printing machine and considerable cost for its maintenance. In order to apply the proposed framework to the printing machine, firstly, a set of inner components of the machine need to be chosen. Then, once the eligible components have been identified, a proper IoT infrastructure design must be selected. The selected IoT platform is applicable for PdM applications and can be used for other purposes such as system performance optimization, product quality optimization, innovative production, supply chain efficiency, and optimize resource usage. Subsequently, a set of algorithms and methods supporting predictive maintenance will be taken out. Finally, the last step concerns the exploitation of IoT-enabled monitoring to ensure that predictive maintenance brings enough business added value.

There are a few main challenges for designing a system framework considering NTS printing machine constraints. Firstly, the extraction of relevant information from multiple data sources can be seen as a challenge that needs to be tackled. Secondly, a reliable and accurate predictive maintenance framework must address fundamental technologies such as big data management and computation as well as correlation techniques. These fundamental technologies come with an extra price tag that is considered in the framework. Thirdly, as the computing and storing processes can be done in both on-premise and cloud devices, multiple potential strategies are to be explored in terms of the system's scalability and cost-effectiveness. These strategies can be categorized into the following three groups: (i) Everything in the cloud (ii) Everything in edge devices (iii) Partially in the edge devices and remaining in the cloud. As for the first implementation, we need to realize other KPIs of the framework such as accuracy and error values, firstly we decided to investigate a completely on-premise framework.

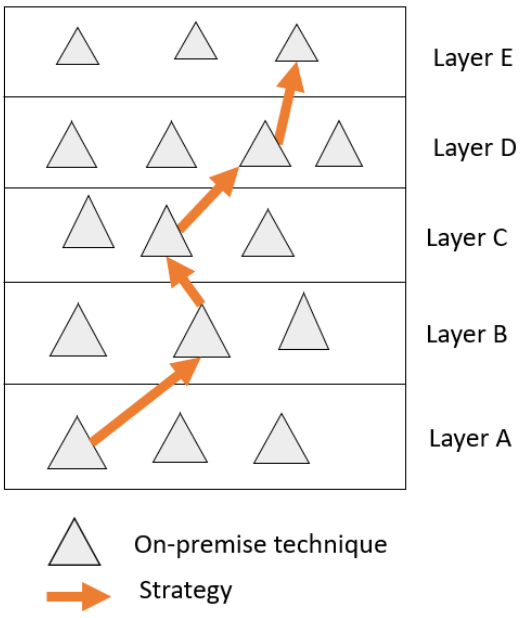

Figure 1.2: Framework, technique and strategy term illustration

Figure 1.2 illustrates what we did precisely in this dissertation. First, we need to design a framework with the specified number of architecture layers based on our requirements. Each layer includes different techniques for achieving the purpose of the corresponding layer. Each technique can be deployed on the cloud or on-premise infrastructure. In this dissertation, an on-premise approach is employed. The important item that is indicated by orange arrows is strategy. It indicates which group of selected techniques in different layers can provide the expected quality for the proposed framework.

1.3 Research Questions

Having identified different layers, techniques, and strategies for implementing predictive maintenance, we formulated the central research question for this work (i.e., *Central-RQ*) as follows:

> ***Central-RQ.*** *What are the efficient strategies to fortify an industry with predictive maintenance?*

To answer this central research question, we aim at developing a predictive maintenance framework, which can be seen as a guideline to select the best strategies. There are two common research methodologies for developing such a framework, namely: *top-down* and *bottom-up* methods. The top-down approach begins with identifying the requirements, while the bottom-up approach starts with an existing body of knowledge.

Since there are some related studies in the development of a predictive maintenance framework in the literature, such as [8, 9, 13, 14, 15], we can conclude that the bottom-up fits our research. Consequently, we conduct survey research on the existing predictive maintenance techniques and investigate the requirements, challenges, and constraints of that predictive maintenance framework. Therefore, we propose a comprehensive predictive maintenance framework by integrating the existing body of knowledge in the predictive maintenance domain. This research methodology leads to the following three sub-research questions. The first research question (i.e., *RQ1*) addresses the available techniques in the literature, which is formulated as follows:

> ***RQ1.*** *What are the available predictive maintenance techniques based on Industry 4.0 paradigm?*

To answer this research question, we systematically start with Industry 4.0 by constituting the layers of this paradigm. A predictive maintenance framework, thus, can be designed in a way that each layer in Industry 4.0 paradigm is mapped to a separated section in the manufacturing process. Nevertheless, all the sections within the predictive maintenance framework must be able to communicate with their adjacent layers. Therefore, the output of each layer is utilized to feed the next layer in the predictive maintenance framework. In addition, several techniques based on the required application can be employed per layer.

A combination of Industry 4.0 layers and multiple existing predictive maintenance techniques can result in a complex mapping process. This complex process can even be more complicated by adding an infrastructure selection dimension (i.e., cloud vs. on-premise) as well as the constraints that each application can bring. Therefore, we need to find a systematic approach to reduce this complexity and to provide transparency. Accordingly, we formulated the second research question (i.e., *RQ2*) as follows:

RQ2. *How can a comprehensive predictive maintenance framework be developed?*

Having developed the predictive maintenance framework, we need to evaluate it against its reliability. Few quantitative KPIs need to be determined. The following three approaches do the process of KPI determination:

1. Based on a set of commonly used KPIs in the literature
2. Based on the consultation with domain experts at NTS Group
3. Based on predictive maintenance applications demand.

Depending on the application, accuracy, availability, and requirements, the determined KPIs can have different impacts. Thus, in order to address this trade-off, we introduce a related KPI. Having defined the KPIs for the performance of the proposed predictive maintenance framework, the third sub-question (i.e., *RQ3*) is, therefore, formulated as follows:

RQ3. *How can the predictive maintenance framework be evaluated against quality criteria such as accuracy and Root Mean Square Error (RMSE)?*

1.4 Contribution

This work contributes to the predictive maintenance research domain by (i) providing an extensive overview of existing techniques, (ii) highlighting the current limitations and challenges of the techniques, (iii) developing a comprehensive predictive maintenance framework based on the Industry 4.0 paradigm, and (iv) evaluating the proposed framework using a real-world industrial use case. In summary, the main contribution of this dissertation can be categorized into the following two subjects:

Contribution 1. *Proposing a comprehensive framework for predictive maintenance based on Industry 4.0*

The following three steps are taken place to achieve the first contribution:

1. Selecting a technique (e.g., model-based or data-driven) to design predictive maintenance framework and its running applications for a given failure prediction scenario (e.g., pump failure in ink supply),
2. Proposing effective techniques for each layer of the framework.
3. Proposing a heuristic-based Machine Learning (ML) algorithms to solve the NP (non-deterministic polynomial) hard problem stated in RQ2,

Contribution 2. *Evaluating the performance of the proposed technique by defining relevant KPI such as accuracy, error values*

To achieve the second contribution, the following three steps are considered:

1. Analysing the performance using both theoretical and experimental approaches,
2. Exploring the remaining useful life estimation topic on the target equipment
3. Developing test scenarios for our experimental evaluation, which can be extended for larger experiments.
4. Comparing several scenarios to investigate the effectiveness of each block in the proposed framework

1.5 Dissertation Outline

This section presents the outline of this dissertation.

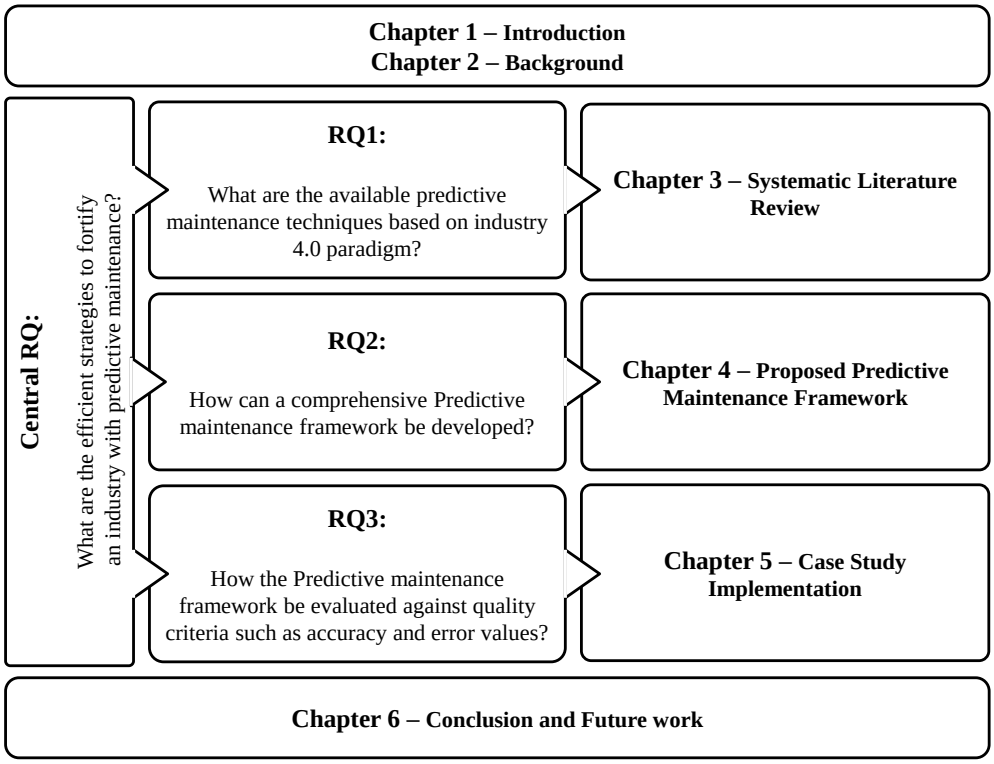

Figure 1.3: A proposed dissertation outline

As illustrated in Figure 1.3, the dissertation is structured as follows:

- Chapter 1 introduces our work in more detail.
- Chapter 2 represents background information and case study which is a printing machine in NTS
- Chapter 3 answers the first research question (i.e., RQ1) by introducing the existing developments in predictive maintenance technology, including and additionally covers the carried out related work.
- Chapter 4 addresses the second research question (i.e., RQ2) by proposing a methodology to introduce a predictive maintenance framework and its resulted framework, respectively.
- Chapter 5 seeks to find a response to the final research question (i.e., RQ3) by introducing a set of KPIs regarding the accuracy and error value of the proposed framework and discussing the result of applying the proposed framework on the case study of NTS Group.
- Finally, Chapter 6 concludes the dissertation and suggests some research directions for future works.

Chapter 2

Background

This chapter aims to deliver a brief overview of Industry 4.0 and predictive maintenance technology by defining the terminologies, framework, strategies, and concepts relevant to the work.

2.1 Industry 4.0

Nowadays, industrial production is facing a new revolution, namely Industry 4.0, which integrates Internet technologies into the industrial manufacturing process, maintenance management, and maintenance strategies.

Industry 4.0 brings flexibility, adaptability to the system, in comparison to the traditional manufacturing production, by employing a selected set of base concepts, described as follows [6]:

- *Industrial Internet of Things (IIoT)*: The IIoT uses the Internet of Things (IoT) technologies in an industrial environment to enhance the processes' performance, safety, reliability, and efficiency. To this end, IIoT collects the sensor data to turn them into actionable information by cost-effectively employing big data analytic tools.
- *Cyber-Physical Systems (CPS)*: CPS refers to a computational and physical system for controlling and monitoring the processes with feedback loops. The CPS architecture consists of five levels, including: (i) Smart connection, (ii) Data-to-information Conversion, (iii) Cyber, (iv) Cognition and (v) Configuration [6].
- *Cloud Manufacturing* [16]: Cloud manufacturing consists of cloud computing, the IoT, service-oriented technologies, and high-performance computing [17], which transforms manufacturing resources and capabilities into manufacturing services.
- *Machine Learning (ML)*: ML is a technique of artificial intelligence that enables computers to learn by detecting the data patterns and adjusting the program accordingly, without explicit programming. The proper implementation of ML into manufacturing processes such as maintenance, scheduling, and quality control, has assured to improve traditional approaches to support the decision making and predictiveness [18, 19].
- *Condition-Based Maintenance (CBM)* CBM is a type of predictive maintenance strategy that uses sensors to evaluate equipment condition over time while it is in operation and suggests maintenance decisions based on the collected data [20]. CBM aims to recommend maintenance only when the data shows a decrease in performance or a failure is predicted, rather than having maintenance at specified intervals [21, 22]. The three main steps of CBM are data acquisition, data processing, and maintenance decision making.
- *Cloud Computing*: Cloud computing plays an essential role in the continuous development of Industry 4.0 by enabling the storage and access to the data over the Internet. Also, it brings multiple advantages to the manufacturing enterprises such as avoid upfront ICT infrastructure costs, focus on core business rather than spending money and time on computer infrastructure, higher applications' speed, improved manageability, less maintenance, and rapid adjustment of ICT resources [23].

2.2 Maintenance Methods

As defined by European Standards' EN 15341 [24], maintenance includes administrative, technical, and managerial tasks during the life cycle of an item intended to keep it in or bring it back to a condition in which it can perform the required function. A maintenance action consists of activities including monitoring, condition analysis, routine maintenance, overhaul, repair, and rebuilding [25]. In recent years, there have been more interests in advanced maintenance strategies, which can be classified into three main categories as Corrective Maintenance (CM), Preventive Maintenance (PM), and Predictive Maintenance (PdM).

2.2.1 Corrective Maintenance

CM strategy considers performing the repair when a failure or breakdown happens to the equipment. Therefore, the asset can continue to operate until the parts start to malfunction as such that the system is no longer operational. In this method, the costs of repair and downtime are considered to be less than the required costs for a maintenance program [24]. Important to note that downtime typically not only relates to the one system that fails but to an entire production line that this machine is working inside it. An assessment of the pros and cons of CM is presented in Table 2.1.

Table 2.1: Pros and Cons of Corrective Maintenance

Corrective Maintenance	
Pros	Cons
• Without planning: there is no scheduling for component maintenance or replacement • Completely broken component: the equipment is utilized till the part cannot be used anymore and totally worn out	• Unhappy customer: unexpected failure cause increasing downtime of the production line and with the dissatisfaction of customer it can lead to financial loss • Profit loss: unexpected downtime causes loss of planned production • No valuable lesson from failure events: when facing unplanned downtime, so it needs to be fixed immediately, so there is no opportunity to find the root cause and plan to avoid the same failure

2.2.2 Preventive Maintenance

PM strategy is used when the failure of an asset is assumed to be costlier than the prevention. It is an approach that employs knowledge of the machine regarding how the components break down. Time-based and risk-based methods are used to schedule inspections and maintenance of the asset to increase the components' life-cycle. Time intervals are estimated from historical data breakdown or supplier recommendations. An assessment of the pros and cons of PM is given in Table 2.2.

Table 2.2: Pros and Cons of Preventive Maintenance

Preventive Maintenance	
Pros	Cons
• Decreased downtime: while replacement parts are exchanged earlier than failure, there is no unscheduled downtime • Well-organized maintenance planning: replacement parts and maintenance service specialists are exist • Enhance machines' lifetime expectation: through exchanging parts earlier than they are broken; the overall function is not at risk • Costs of expectable maintenance	• Capital loss: parts are often exchanged earlier than they are entirely worn down • Enhanced the costs of maintenance planning • Maintenance scheduling: executing checks based on time intervals does not constantly take into account the machine's operational time • Risk associated with unexpected adjustment in equipment working condition/equipment depreciation

2.2.3 Predictive Maintenance

PdM strategy is employed when the equipment breakdown has a critical consequence related to Health, Safety, and Environment (HSE) or operations. In other words, while PM has the goal to minimize downtime, PdM aims to maximize uptime.

Table 2.3: Pros and Cons of Predictive Maintenance

Predictive Maintenance	
Pros	Cons
• Maximum uptime: protect the component completely from failure, with knowledge of the health condition of the asset • Flexible maintenance planning: according to the need of the system and spare parts, maintenance technicians can be scheduled • Minimize the cost: reduce the expense of downtime and avoid the unnecessary cost of replacing parts • Components optimum utilization: machine parts are used until shortly before they are no longer operational • Enhance machines' lifetime expectation: utilizing exchanging parts earlier than they are broken; the overall function is not at risk	• Costs of maintenance: high investment and working expenses • temporary costs: requirements-based cares provide less repair cost probability • Costs of predicting false positives: need to equipped the production line for unnecessary shutdown • Enhanced requirement for flexibility: require to adjust to real-time maintenance services and solution

In general, we can mention that the condition of the equipment is evaluated and compared to a healthy operating state. Maintenance is carried out when defined indicators warn that the equipment is deteriorating and the breakdown probability increases. PdM is realized with the utilization of several condition monitoring techniques. There exist a wide variety of offline/online monitoring techniques related to the application. Other than visual inspections, the most used condition monitoring techniques are vibration

monitoring, process parameter monitoring, oil-debris monitoring, and acoustic emission monitoring.

Considering the growth of the availability of data and computing powers will allow operators to evolve beyond condition monitoring to anticipate problems before they happen, making PdM an attractive and life-changing item in manufacturing systems [26]. We can define PdM as the utilization of a Remaining Useful Lifetime (RUL) [27] model to evaluate the state of the machine and predict the RUL of the components. In this work, we are concerned with PdM as defined here. An evaluation of the pros and cons of PdM is shown in Table 2.3.

2.3 Industrial Internet of Things and Smart Maintenance

The industry-relevant items (e.g., material, sensors, machines, products, supply chain, and customers) can be connected by taking benefit from the Internet of things (IoT) and Cyber-Physical System (CPS), which means these necessary items are going to exchange information and control actions with each other individually and autonomously [28].

Industry and academia demand a complete structure of these technology applications to show manufacturing development with different performance levels. First, we need to determine what we have currently in the manufacturing system compared to the smart factories [29]. A brief comparison between current and Industry 4.0 factories is shown in Table 2.4. To bridge the gap between current manufacturing systems

Table 2.4: Comparison of today's factory and an Industry 4.0 factory [12]

Data source		Today's factory		Industry 4.0	
		Attributes	Technology	Attributes	Technology
Component	Sensor	Precision	Smart sensors & fault detection	Self-aware Self-predict	Degradation monitoring &
Machine	Controller	Producibility & performance	Condition-based monitoring & diagnostics	Self-aware Self-predict Self-compare	RUL prediction Up time with predictive health monitoring
Production system	Networked system	Productivity & OEE	Lean operations: work and waste reduction	Self-configure Self-maintain Self-organize	Worry-free productivity

and Industry 4.0, IoT needs to be combined with data science and modeling capabilities. This combination helps to reach the ultimate objective of digitization, which is supporting decision making to act on the physical systems [1] optimally.

Now, we need to look at how maintenance fits into Industry 4.0. One of the main challenges of Industry 4.0 is the lack of an international standard for implementation. Ref. [30] describes a standard Reference Architecture Model for Industry 4.0 called "RAMI 4.0", which can help with this problem. According to "Platform Industry 4.0", the following challenges are creating sub-models for individual processes, creating a common language, and specific recommendations for implementation. RAMI 4.0 reference architecture is illustrated in Figure 2.1. The model breaks down complex processes into understandable modules, ensuring that all participants involved in Industry 4.0 discussions understand each other. RAMI 4.0 maps all the players of the connected industry employing three axes of definition:

- "Layers: functional, business, information, communication, integration, asset."
- "Life Cycle value stream: development, production, maintenance usage."
- "Hierarchies levels: product, field device, control device, station, work centers, enterprise, connected world" [30]

Conducting this dissertation within the RAMI4.0 reference architecture, we can mention that we are concerned with raising a field device in the maintenance instance from the asset layer to the digital layers. Considering essential standards relevant for Industry 4.0, several models are identified employing five to nine-layer models [31].

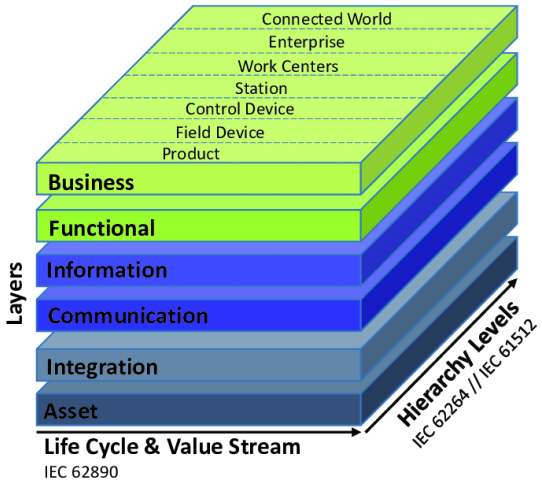

Figure 2.1: RAMI 4.O reference architecture for Industry 4.0 [30]

A 5-level CPS structure is proposed in [12], namely, the 5C architecture, which provides a roadmap for developing and deploying a CPS for industrial applications. A CPS contributes in two main functional components: (a) the state-of-art connectivity that ensures real-time data acquisition from the physical asset and information assessment from cyberspace; and (b) intelligent data storage, management, and analytics capability that constructs cyberspace. However, such a requirement is very abstract and not specific enough for implementation purposes in general. In contrast, the 5C architecture presented in Figure 2.2 clearly defines- through a sequential workflow manner- how to construct a CPS from the initial data acquisition to analytics to the final value creation.

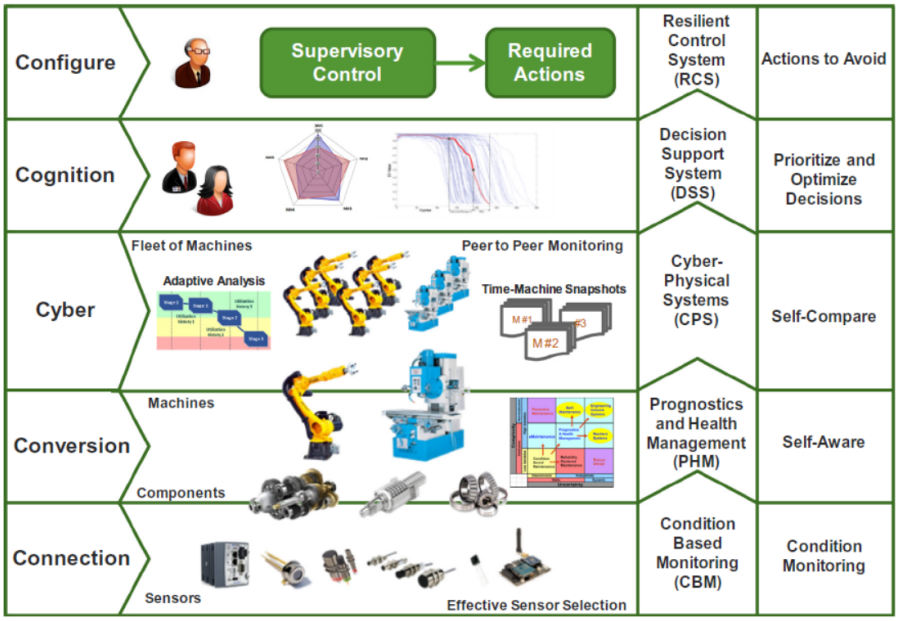

Figure 2.2: 5C architecture and its related applications and techniques [12]

Since for developing PdM, base on RAMI 4.0 [30], first, we need to choose the equipment to develop PdM for that; the first layer can be a physical assessment and choosing the critical component. The next step is to analyze the system, the failure condition of the corresponding component, and the sensors related

to that equipment. Taking into account these steps from RAMI 4.0 [30] and '5C' architecture [12], for developing a smart factory, we need to develop that process based on seven layers to provide us more transparency and reduce complexity:

1. **Physical layer:** A physical item (asset/machine/system) must be chosen to apply the PdM application.

2. **Data acquisition layer:** Each item, e.g., a machine, can generate data about itself, so we must choose between various types of sensors relevant to maintenance applications (data can be collected from different equipment).

3. **Connection layer:** The data can be transferred into specific cyberspace (cloud and edge devices).

4. **Conversion layer:** The collected data are of high volume, and variety needs preprocessing to reduce and clean the resources needed for computation.

5. **Computation layer:** Signal analytics utilizing software and algorithms such as ML.

6. **Cognition layer:** Creation of maintenance decision support with specific diagnostics and prediction of machine health.

7. **Configuration layer:** Movement from cyber to physical space where intelligence is transformed into action looped back to the application, e.g., adjust a parameter on a machine on the factory floor from the cloud.

This 7-layer architecture, considering the NTS printing machine, can utilize to develop an end-to-end PdM architecture in the case study. In Section 1.1, we will look at on NTS printing machine and control management system.

2.3.1 Predictive Maintenance Models Classifications

It is illogical for potential industry users to assess each specific model separately. Therefore, a classification system is required to evaluate similar model variants based on their advantages and disadvantages. Strategies for diagnostics and prognostics can be divided into four main modeling groups, which are: Knowledge-based, Model-based, Signal-based, Data-driven, Hybrid based [32, 33].

Knowledge-based

These evaluate the similarity between an observed situation and historical data of previously defined failures and determine the life expectancy from previous events. Knowledge-based includes two categories as following: a) Expert systems, b) Fuzzy systems [33].

- **Expert:** A software program is an expert system that resembles the performance of human experts in a particular domain. It consists of a knowledge-based including aggregated experience from subject experts and a rule base for applying that knowledge to specific problems known to the software system. Rules are formulated as exact IF-THEN statements; these primarily rely on heuristic facts obtained by one or more experts during multiple years of experience [33].
- **Fuzzy:** Logic models are most effective when one or more of the input variables are continuous, a mathematical model is not available or not implementable, and data contains high noise levels [34]. The multiple uses of fuzzy systems for control state that it can be an appropriate method for predicting RUL. They can also provide results within complete or inaccurate data, as is commonly found in practice. Nevertheless, they can explain their reasoning and, by defining fewer rules, are simpler to adapt than expert systems. Unfortunately, they too rely on the availability of a suitable expert to specify the rules underlying system behavior and develop the fuzzy sets representing each variable's characteristics [33].

The knowledge-based system needs to be updated and maintained as more knowledge is obtained or configurations change. This continuously changing system can be problematic. However, these problems can be partly reduced by integrating with fuzzy logic. The knowledge-based needs to be updated and maintained as more knowledge is obtained or configurations change. For example, in [35] they used an expert system with fuzzy logic and neural networks to predict the remaining useful life of gearboxes.

Model-based

A model-based prognostic approach is based on mathematical models of system behavior obtained from physical laws or probability distribution. For example, conventional model-based prognostics include mathematical methods based on Wiener and Gamma processes [36], hidden Markov models [37], Kalman filter [38], and particle filter [39]. Physical models evaluate an output for the remaining useful life of a component by solving a deterministic equation or set of equations gained from big empirical data. Some of this data will have been converted into common scientific and engineering knowledge, while other data must be acquired through specific laboratory or field experimentation [40]. One of the disadvantages of model-based prognostics is that an in-depth understanding of the underlying physical processes that lead to system failures is required. Another disadvantage is that it is assumed that underlying processes follow certain probability distributions, such as gamma or normal distributions [41].

Signal-based Models

In contrast, using explicit input-output models for fault diagnosis, the Signal-based methods utilize measured signals. The faults in the process can be found in the measured signals, whose features are extracted, and a diagnostic decision is then made based on the symptom analysis and historical knowledge of the symptoms of the health of the system.

Signal-based fault diagnosis methods can offer an extensive application in real-time monitoring and diagnosis. There are two ways to extract the feature signals for pattern/symptom investigation: A) The first one is the time-domain such as mean, trends, standard deviation, phases, slope, and magnitudes (e.g., peak and root mean square, B) The second one is the frequency-domain like a spectrum. By knowing that, we can categorize signal-based fault diagnosis methods into three approaches: time-domain signal-based, frequency-domain signal-based, and time-frequency signal-based [42].

- Time-Domain Signal-Based Methods: This approach is based on the time wave-form itself. Traditional time-domain analysis computes characteristic features from time wave-form signals as descriptive statistics such as mean, peak, peak-to-peak interval, standard deviation, crest factor, high order statistics: root mean square, skewness, kurtosis. The mentioned features are named time-domain features. Time synchronous average is one of the popular time-domain analysis approaches. TSA utilizes the ensemble average of the raw signal over several evolutions in an attempt to eliminate or decrease noise and effects from other bases to improve the signal components of interest [42].
- Frequency-Domain Signal-Based Methods: This method recognizes changes or faults by applying spectrum analysis tools like discrete Fourier transformation. Here, motor current signature analysis can be introduced as a reliable method to diagnose motor faults. MCSA applies the stator current spectral analysis to detect the rotor faults belonging to the broken rotor bars and mechanical balance. Thus, the MCSA approach became more attractive to researchers [43, 44] because it does not need motor access. The recent development of current-based spectrum signature analysis for fault diagnosis is presented in [45, 46].
- Time-Frequency Signal-Based Methods: In some situations, the computed signals are usually transient and dynamic under the required time section (e.g., machines under load torque oscillations, varying load, unbalanced supply voltages, or an unloaded condition). For the mentioned conditions, analysis of the stationary quantities is not easy to monitor or detect faults by time-domain or a frequency-domain method. Therefore, appropriate time-frequency decomposition means are desirable for real-time monitoring and fault diagnosis because of the time-varying frequency spectrum of the transient signals. Extracting feature information in nonstationary signals made this method an effective means for fault diagnosis and monitoring [47]. There are multiple time-frequency analysis

approaches for diagnosing the machinery fault, which can be named as short-time Fourier transform, wavelet transforms, Hilbert–Huang transform, and Wigner–Ville distribution [42].

Data-driven

This approach can be considered a kind of signal-based model, although more data mining techniques are employed in the data-driven model. The growth of industrial data gives us the chance to conduct maintenance projects for the development and deployment of the data-driven PdM, which employs advanced computational techniques to provide helpful information regarding the condition of equipment acquired from the growth of operational data [48]. The data-driven PdM system consists of two main steps: first, a learning process (i.e., model training) is needed based on historical raw sensor signals; second, the trained model is applied to predict targets and make decisions. In addition, each phase consists of the following three sub-steps [6]:

1. Data acquisition and preprocessing, which can be single sensory or multi-sensory,
2. Feature engineering, which contains feature extraction, concatenation, and selection,
3. Model training and predicting, in which a well-trained model will be generated with the optimal parameters.

The model can predict the real-time data flow. Nevertheless, the data-driven strategy has been widely used for industrial manufacturing employing ML [49] and Deep Learning (DL) algorithms [50].

Hybrid-based

Different strategies have different strengths that allow them to recognize specific fault modes and not others. As a result, some researchers have combined various strategies to detect more fault modes with better accuracy. In most cases, fuzzy logic was integrated with a data-driven method in the hybrid system [51].

In summary, as model-based and data-driven are the newest techniques compared to knowledge-based, we are trying to employ one of these trend techniques. Although, due to the complexity of developing a mathematical model of the system, the model-based technique is out of our scope, and our main focus will be on the data-driven method.

2.3.2 Machine Learning

From the perspective of maintenance, a system can be seen as a self-maintained/self-aware machine when it can self-assess its health conditions. In order to avoid possible faults, the machine also should apply similar information from other peers for smart maintenance decisions [52]. Intelligent analytics can be applied to each machine for achieving intelligence. The ability to measure the machine's health conditions and deliver the assessment outcome considers as self-aware for a mechanical system. Employing the data mining technologies, the evaluations can be achieved in order to investigate the collected data from the particular machine and its ambient conditions [24].

ML and DL algorithms are applied in data-driven PdM for industrial manufacturing to diagnose the fault [6, 49, 50]. Classical ML algorithms, as shown in Figure 2.3 section (a), (e.g., Support Vector Machine (SVM), Random Forest (RF), Logistic Regression (LR), Decision Tree (DT)) usually need to gather a huge quantity of data from the health conditions and several failure status situations for model training. Then, the feature engineering is derived from the time, frequency, and time-frequency domain [53, 42], and the machine's health is achieved by applying the extracted features. Nevertheless, DL, as depicted in Figure 2.3 section (b) (i.e., various neural network models), avoids the mentioned complex feature engineering and can be known by applying an end-to-end learning method, which is accomplished by means of adding deep layers between the prediction outcomes and the raw data. Therefore, the deep models can be seen as a "black box," which delivers the input estimation result, considering this as the important difference between ML and DL. For all of these thoughts, both ML and DL have been extensively applied in the PdM's application.

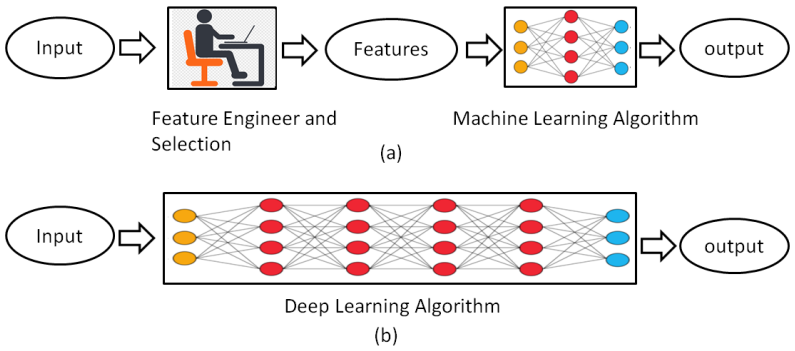

Figure 2.3: Flow of machine learning (a) and deep learning (b), reconstructed from [6]

For ML algorithms, sufficient data harvesting and feature engineering help improve algorithm performance. For DL algorithms, the deeper network architecture and the higher dimensional feature vectors have a significant impact on optimizing the task metric [6].

2.3.3 Predictive Maintenance Framework

Related projects that attempt to create a framework for predictive maintenance are outlined in the following. The IMS Watchdog Agent™ tool kit [54, 55] is a toolbox of algorithms to assess and predict the performance of a machine by using modeling and forecasting indicators without considering the decision making, and it is not involved the association with enterprise systems or MES. Henceforth, the response to foreseen failures includes human communication and manual synchronization between the enterprise systems and the Watchdog Agent™ instances, considering that the Watchdog Agent™ is a monolithic solution and does not propose a variable interface to the shop floor. One of the predictive maintenance applications which were designed for industrial processes is SIMAP [56], based primarily on neural networks to model and forecast the indicators. Moreover, SIMAP comprises functionality for scheduling the actions of maintenance. Nevertheless, SIMAP has only been applied in a small setting for a wind turbine until now. The scalability of the decision making module to larger setups is doubtful, and still, it is not involved in the association with enterprise systems or MES. Such integration was recognized in the PROTEUS project [13], a generic platform for e-maintenance, i.e., it incorporated current maintenance management applications to permit a comprehensive workflow. Since PROTEUS was intended to support maintenance operations itself, the goal of PROTEUS is different from that of our framework.

A manufacturing big data ecosystem was proposed in [57] in order to present the problems of big data ingestion, management, and analytics for fault detection in PdM at IoT-based intelligent factories. The presented method applied a real manufacturing big data ingestion procedure on the whole framework. The approach achieved many things on the Apache Spark platform, such as effective data management, guaranteed data security, and real-time data analytics through the deployment of a data lake, NoSQL database, encryption protocol. In addition, the MapReduce-based DPCA approach was explained for the fault detection model in this reference. The report confirmed that the presented big data ecosystem could alarm the system in a real-time fashion several days before an actual event. MapReduce Principal Component Analysis (PCA) model was presented in [57] to recognize and diagnose the fault. PCA model works only with unlabeled data without benefits from any data labeling functionality.

A system framework presented in [8] based on Industry 4.0 concepts, which consists of the fault analysis and treatment process for predictive maintenance in machine centers. The framework comprises different modules: sensor selection and data acquisition module, data preprocessing module, data mining module, the decision support module, and maintenance implementation module. Therefore, it displays a whole framework without discussing the communication layers between different modules of this framework. In addition, the cloud or on-premise infrastructure was not investigated. In [14], investigated a systematic framework that took advantage of using the cyber-physical systems. The framework is a five-level architecture for operating CPS in the manufacturing process. The process from acquiring data until generating meaningful information and the decision making process for the end-user are covered in

this architecture. The 5C architecture applies new calculating and communication methods such as cloud computing to offer connectivity between machines. Moreover, an adaptive clustering method has been presented to achieve the necessities of this architecture with more innovative analytical methods, which can automatically recognize new working regimes.

The base concepts, materials, and methods for developing an Industry 4.0 architecture were presented in [9]. The main focus of this research is on predictive maintenance, whereas relying on low-cost principles in order to be reasonable by Small Manufacturing Enterprises. A cost-effective, easy-to-develop cyber-physical system architecture was presented as the result of this investigation. This PdM system supplies data in the cloud where the Recursive Partitioning and Regression Tree model technique performs to predict the rejection of machined parts based on a quality threshold.

PdM framework taking into account the Big data challenges was investigated in [58]. It is shown that the traffic load over a network can be diminished significantly by applying cloud computing and cloud storage facilities. A mobile agent-based approach for predictive maintenance in cloud manufacturing is also presented in this report. The mobile agent-based approach is an emerging technique that allows a new paradigm for predictive maintenance as remote services instead of traditional centralized methods and offers distributed maintenance services within the manufacturing enterprises. Furthermore, a mobile agent can arrange different services (e.g., signal processing algorithms) to adjust variable operations and tasks in a dynamic manufacturing situation. Mobile agents allocate signal processing algorithms (e.g., feature extraction) to the cloud nodes instead of transmitting raw sensing measurements to the central server, and especially in Big Data Era, they can decrease the traffic load over the network. The described mobile agent-based approach codes the feature extraction algorithms in the mobile agents, direct them to the cloud nodes for locally processing raw data, and transports the extracted features to the central server. Such a new approach could greatly decrease data transmission and, consequently, advance system efficiency.

A maintenance platform has been developed in [15] by applying lambda architecture [59] and taking into account the data-driven model. The computing process and data storage in edge and cloud nodes are positioned in this platform. Updating the learning model is the approach that the architect can advance the accuracy. Lambda architecture is a suitable example while consider evolving a framework that aids from edge-cloud were calculating.

Chapter 3

Systematic Literature Review

This chapter aims at addressing the first research question of this study. This question enquires available predictive maintenance techniques based on Industry 4.0 paradigm. To answer the first research question, we aim to survey the existing PdM frameworks and their available techniques using the Systematic Literature Review (SLR) methodology to identify, classify, and analyze a set of studies.

SLR is a famous method that is broadly used to recognize, analyze, interpret and assess the existing body of knowledge in a specific research interest [60, 61]. Reviewing the existing literature based on a predefined review protocol is the main advantage of the SLR methodology, which produces impartial, precise, and robust review results. This predefined protocol is also employed to confirm the reproducibility of the obtained results. For the implementation of SLR, we follow the Kitchenham guideline [60].

The rest of this chapter is categorized into three sections. First, Section 3.1 presents the review protocol that was employed as a basis for conducting our survey. Section 3.2 discusses the steps and intermediate results that lead to the selection of the final set of studies. Subsequently, Section 3.3 reports the obtained results.

3.1 Review Protocol

In this section, we present the review protocol that was employed to conduct our study. This protocol specifies the research questions as well as the sources, search strings, and the criteria that were employed in order to select relevant primary studies. First, a broad search was conducted to find a set of papers proposing frameworks and their techniques for PdM. Subsequently, we extracted and synthesized data from this set according to data extraction fields.

The first step in an SLR is to construct the research questions. We formulated three research questions to acquire knowledge about the existing PdM frameworks and their techniques within research communities.

- ***SLR-Q1***. What are the available PdM frameworks?
- ***SLR-Q2***. Which PdM frameworks are in the context of Industry 4.0?

The next step in SLR is to define a search strategy that first identifies the initial set of primary studies. To this end, we define the *Search Sources* as shown in Table 3.1. These sources were chosen since they reasonably cover most of the scientific publications (e.g., journal papers, conference proceedings, and workshop papers) in the field of computer science. As also suggested by [61], these databases guarantee to provide the confidence level for coverage of all the required primary studies. After identifying the research questions and having a relevant literature database, we employ selection criteria to include or exclude a primary study. These criteria aim to narrow down the obtained results by excluding the studies that were not relevant to the proposed research questions. As suggested in [60, 61], the selection criteria consists of a set of *quality criteria* and a set of *exclusion criteria*. For qualitative evaluation, we consider the following points into account: Firstly, whether the authors propose a framework, secondly, if the work uses a data-driven approach, and lastly, whether the scope of the work is towards an industrial application. Therefore,

Table 3.1: Utilized Electronic Databases

Index	Database	Institution
1	ACM Digital Library [62]	Association for Computing Machinery (ACM)
2	IEEE Xplore [63]	Institute of Electrical and Electronics Engineers (IEEE)
3	ScienceDirect [64]	Elsevier
4	MDPI [65]	Molecular Diversity Preservation International
5	SpringerLink [66]	Springer

we apply the following questions to select papers that meet the quality requirements:

- ***QC1***. Is there a framework proposal?
- ***QC2***. Is the framework based on a data-driven methodology?
- ***QC3***. Does the scope of the paper cover an industrial application?

Furthermore, it is necessary to remove all studies that are not relevant to the scope of this dissertation. The following exclusion criteria apply in this process of removal:

- ***E1.*** Works not related to PdM.
- ***E2.*** Works not related to ML.
- ***E3.*** Works that do not present frameworks.
- ***E4.*** Works dated before the year 2010.

3.2 Review Conduction

This section presents the steps and intermediate results that lead to select the final set of studies. To begin this selection procedure, we formulate specific search strings, which used for each database, as described below:

- ***ST1***: Abstract: ("predictive maintenance" AND "framework") AND Keyword: ("predictive maintenance") AND All Metadata: ("machine learning")
- ***ST2***: Abstract: ("predictive maintenance" AND "Industry 4.0") AND Keywords: ("predictive maintenance") AND All Metadata: ("machine learning" AND "framework")

It should be noted that the survey was executed on January 29, 2021. A total of 144 studies were taken from the five scientific databases after applying the time consideration criterion (E4). Taking into account the rest of the selection criteria (ECs and QCs), we then read the important parts of the papers, such as abstract, introduction, and figure analysis, to select the final candidates. This amounted to a total of 27 papers. Figure 3.1 shows the number of papers obtained in each of the databases selected in the initial search stage and the final search stage.

3.2.1 Publication Distribution Along the Years

Figure 3.2 shows the number of articles published between 2010 and 2020 (using the extraction criteria of this paper). This search confirms that PdM is a new maintenance technique since before 2017, the number of published papers is 4. On the other hand, after 2017, a growing interest in this research area was noted. This fact is probably associated with the increase in the size of data that industrial equipment and the recent advances in ML algorithms are generated.

3.2.2 Citation Analysis

The number of citations is important for an article since it determines how many times other studies have cited an article. However, as the release date of most papers being 2020, maybe this approach is not as

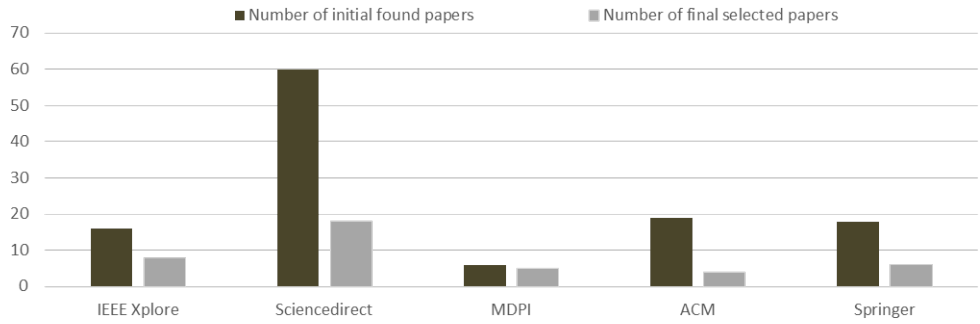

Figure 3.1: Initially found papers vs. final selected papers per each database

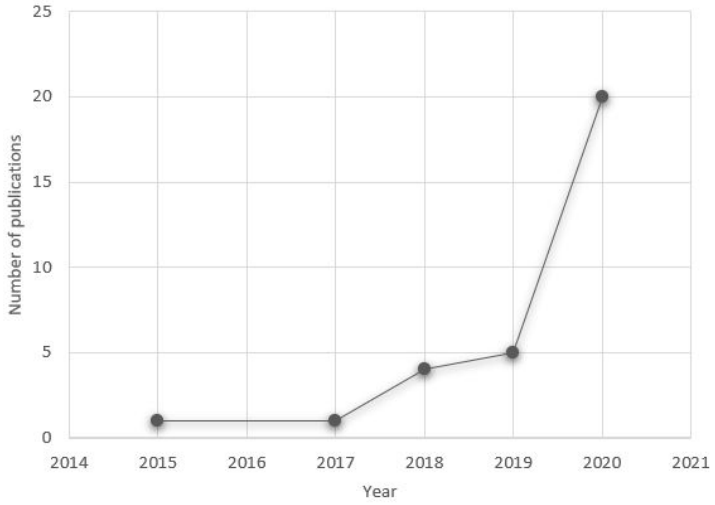

Figure 3.2: Number of papers per year

helpful as expected. Therefore, to perform a citation analysis, the Web of Science platform was selected to determine the number of citations of the selected papers in this review. The citation analysis reveals that in the top 11 cited articles, the work published by B. Bagheri et al. [14], which relies on Cyber-physical systems architecture for self-aware machines in Industry 4.0 environment, received the maximum number of citations (citations=254). Moreover, an article published by Zhe. Li et at. [24] received much attention from the scientific community. This paper presented a framework for formulating a systematic approach and obtaining knowledge based on Industry 4.0 concepts for predictive maintenance. The citation analysis also revealed that the average number of citations of all the research papers is 20.88.

3.3 Review Results

So far, this chapter has focused on the review protocol used to conduct SLR and the reason behind the 27 studies being selected for further evaluation. In this section, we analyze the contributions of these studies, and in each subsection, each defined research question in Section 3.1 is answered.

This research question contributes to the scientific community by identifying the available PdM frameworks. To this end, we suggest investigating the Building Blocks (BBs) of each proposed framework and classify them by identifying the commonly used BBs and their relevant techniques. By analyzing the selected 27 papers, we found the most commonly used BBs. The frequency of repetition of these blocks is

listed in Table 3.2.

Table 3.2: The frequency of frameworks's building blocks

BB	repetition(%)
Data acquisition	48
Data preprocessing	11
Data processing	15
Feature engineering	37
Predictive analytics	15
Decision making	22

After an analysis of the research papers between 2010 and 2021, using the extraction criteria, Table 3.3 was built. It contains an overview of the most recent papers for PdM, where each line is related to a paper. The first column, "Reference," contains the paper reference. The second column, the publication year of papers. The third column shows the citation of the papers. The fourth column, "Database," represents the scientific digital library that each paper is chosen from it. The fifth column, "Search String," shows that with the help of which search string each paper is found. The sixth column, "Equipment," shows the used equipment for maintenance prediction. The seventh column, "Sensor Data," shows the data type used in the ML learning algorithm, which can be Single-sensory (Single) or Multi-sensor (Multi). Finally, the eighth column, "Preprocessing," shows which techniques are employed for preprocessing the data set for feeding the predictive algorithm. These techniques are grouped into four categories. The first group, Data Cleaning, addresses the process of removing or modifying corrupted data that are received from the sensors. The second group, Dimension Reduction, addresses the transformation process of data from a high-dimensional space into a low-dimensional space without losing any meaningful properties of the original data. The third group, Data Enrichment & Correlation, addresses the enrichment process in order to fill the missing information from the original data; this process can be done using the additional data or by finding the correlation among the existing data. Finally, the fourth group, Feature Engineering, addresses the process of extracting special features from raw data, which these features can subsequently be used for predictive analytics. Table 3.4 summarizes these four preprocessing groups.

Table 3.3: Summary of the SLR study

Reference	Year	Citation	Database	Search String	Equipment	Sensor Data	Preprocessing	Predictive Analytics
[67]	2021	0	MDPI	ST1	HVAC System	Multi	C,E	DL
[68]	2020	36	Science Direct	ST2	Hot Rolling Machinery	Multi	E, F	ML
[69]	2020	29	Science Direct	ST1	Chiller	Multi	None	ML/DL
[70]	2020	26	IEEE	ST1 & ST2	Turbine Compressor	Multi	C,E	ML
[71]	2020	10	MDPI	ST1	None	None	F	ML
[72]	2020	6	Science Direct	ST1	Pump	Single	F	ML
[73]	2020	5	MDPI	ST2	Robot Box	Single	F	ML
[74]	2020	3	Science Direct	ST1	Bearings	Single	F	ML
[75]	2020	3	Springer	ST1	None	None	F	ML/DL
[76]	2020	3	Springer	ST1	Magnet Brushless DC Motor	Single	D, F	ML
[77]	2020	2	Science Direct	ST1	Coil	Single	C,D,E,F	ML
[78]	2020	1	Science Direct	ST1	Pump	Multi	E, F	ML/DL
[79]	2020	0	IEEE	ST1	Robot arm	Multi	C,E,F	ML
[80]	2020	0	MDPI	ST1	Stator Armator	Multi	C,D,E	ML/DL
[81]	2020	0	MDPI	ST2	Milling Machine	Multi	C,D,E,F	ML/DL
[82]	2020	0	Science Direct	ST2	Heaterbands, Thermocouples	Multi	F	ML/DL
[83]	2019	10	Science Direct	ST2	CNC	Multi	C,D,E,F	ML/DL
[84]	2019	7	Science Direct	ST1	Robot Box	Multi	C,D,E,F	ML/DL
[85]	2019	5	IEEE	ST1 & ST2	Nozzle	Multi	C,D,E,F	ML
[86]	2019	1	Springer	ST1	Gearwheel box	Multi	F	DL
[87]	2019	0	ACM	ST1	Turbofan engine	Multi	C, F	ML
[88]	2018	57	Science Direct	ST2	None	None	F	ML
[89]	2018	20	Springer	ST1	Energy Storage System	Multi	C,D,E,F	ML
[90]	2018	4	IEEE	ST1	Transformer	Multi	D, F	ML
[91]	2018	4	Springer	ST1	Gas Turbine	Multi	C, D	DL
[92]	2017	83	Springer	ST1	CNC	Multi	C,D,E,F	DL
[93]	2015	254	Science Direct	ST2	None	None	F	ML/DL

Table 3.4: Preprocessing column descriptions

Key	Description
C	Data Cleaning
D	Dimension Reduction
E	Data Enrichment & Correlation
F	Feature Engineering

The last column of Table 3.3, Predictive Analytics, shows which data mining method is utilized for a prediction process that can be traditional ML or DL. With the review accomplish, it can be verified that PdM is being used for the most diverse equipment of the most varied areas.

Table 3.3 shows that each PdM application uses specific equipment. The equipment includes turbines, motors, compressors, pumps, fans, milling machines, among others. Another interesting aspect that emerges from observing Table 3.3 is that there is a preference for multi-sensory data collection in order to detect anomalies in the machines. Through the mentioned aspects of the most recent papers for predictive maintenance (Table 3.3), it is possible to answer research questions SLR-Q1 and SLR-Q2 (described in Section 3.1). To elaborate, the modular design for a PdM framework, considering the industrial application (of the use cases mentioned in papers) and for the most used data-driven approach, there is no preference for equipment to perform PdM strategies. Finally, the main characteristics of the most used techniques and how they are employed in the PdM applications are presented.

Chapter 4

Proposed Predictive Maintenance Framework

Traditional manufacturing automation can be considered model-based manufacturing. Experts obtain experience by making physical observations such as noise recognition and visual inspection from manufacturing systems. Together with these experiences, human intelligence will derive physical models using theoretical, experimental, and numerical methods [75]. Although outstanding achievements have been made and applied in various applications, such as simulation and performance evaluation, these model-based methods have limited accuracy and range due to plenty of simplifications and assumptions. Moreover, considering that the human experts are not assured impartial towards all obtained experiences, we can see that there is no enough accuracy in driving the physical models as well. On the other hand, modern manufacturing is data-driven, in the sense that data generated through manufacturing activities are fully utilized to enhance manufacturing quality positively and thus enrich flexibility and autonomy of the system [75].

This study aims to formulate a systematic approach and obtain knowledge for fault detection, interpretation, and prediction based on Industry 4.0 concepts. Therefore, a system framework is designed in the following sections that include the entire fault analysis and treatment process for predictive maintenance in industrial equipment based on data mining and Industry 4.0 concepts.

4.1 Design Principle

Designing and developing an intelligent system to enable the health status of machines is one of the big challenges for the Industry 4.0 paradigm. PdM has been introduced as a key subject for Industry 4.0, where its application allows for a reduced unscheduled downtime and a consequent improvement in productivity and a reduced production cost [71]. The utilization of vendor-specific solutions for predictive maintenance purposes and the diversity of technologies in brownfield for condition monitoring of industrial equipment reduces the flexibility and interoperability required by Industry 4.0. Considering Industry 4.0 key technologies such as big data, cyber-physical systems, IoT, and cloud computing, a framework to provide a general overview for the end-to-end data life cycle is needed. Therefore, we conduct our framework design based on Industry 4.0 concept and two reference architecture (5C and RAMI 4.0) that was discussed in Chapter 2. Horizontal and vertical integration over the whole value network helps with designing a robust and scalable framework. The system design method allows the developer to select the technologies compatible with their corporate guidelines/specific implementation challenges.

Based on the literature reviewed in Chapter 3, the most commonly used techniques and concepts are considered in designing the framework. In the majority of the papers, only a part of the techniques are focused, and they lack providing an end-to-end architecture for PdM. Therefore, we aim to cover all the aspects, concepts, and techniques to achieve a scalable and robust end-to-end framework. The proposed framework is designed based on a data-driven approach since we consider the data life cycle in the factory from the shop floor to the management floor.

In the design process based on the data-driven methodology, multiple steps are considered, such as data collection, design of an efficient and suitable dataset, and employing the most promising algorithms in the field of machine learning. Finally, maintenance strategies can be implemented based on the result of machine learning. Each step is considered as a building block that, with the combination of these building blocks, we can apply a data analysis process on industrial systems. Some of the most popular techniques found in review papers in predictive maintenance for industrial machines were included in the proposed method.

4.2 Framework Overview

Monitoring systems in industrial machines may require data mining methods for fault diagnosis and prognosis according to different monitoring purposes or components. Therefore, a systematic framework based on data mining to achieve fault diagnosis and prognosis for industrial machines is imperative. Figure 4.1 shows that a system framework is formulated for predictive maintenance based on Industry 4.0 concepts. This framework can monitor plant-floor assets, link the production and maintenance operations systems, obtain data, collect feedback from a local/remote customer site, integrate it into upper-level enterprise applications, discover hidden information about impending failures, and generate maintenance knowledge. It can also monitor the state of manufacturing processes and predict the condition of the equipment.

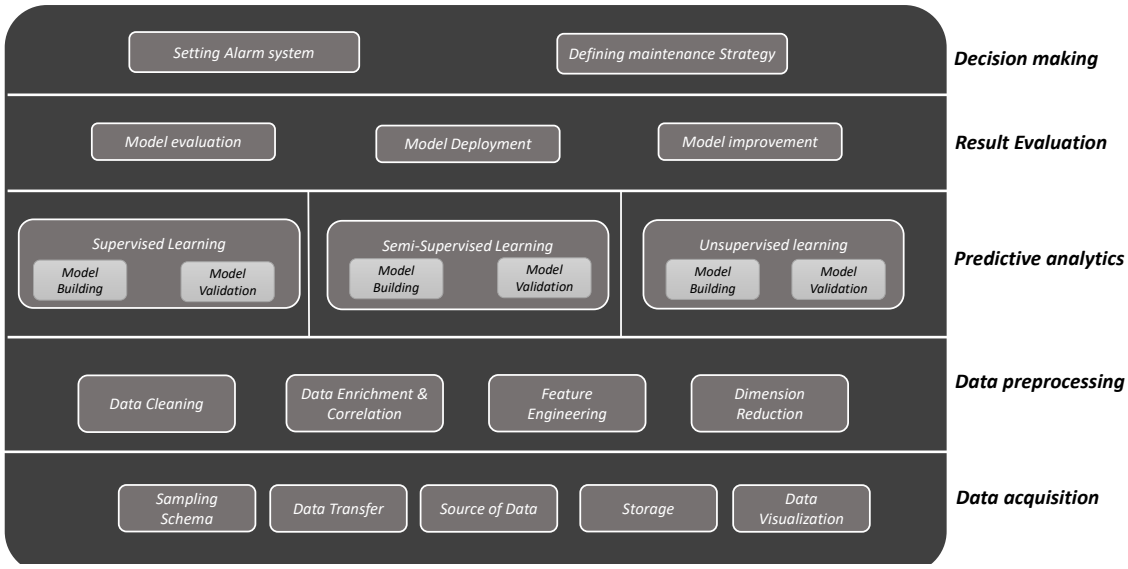

Figure 4.1: End-to-end predictive maintenance framework

The system can make a maintenance decision to prevent failures occurrence effectively to ensure equipment, personal safety and reduce the economic loss caused by failures. In addition, it can use fault diagnosis, performance assessment of the degrading level, and fault prognosis models to achieve near-zero-breakdown performance and improve the company's productivity. The framework includes five main building blocks: **Data Acquisition, Data Preprocessing, Predictive Analytics, Result Evaluation** and **Decision Making Blocks**. All these building blocks have clear ordinal relations and specific functions in the system.

The framework is based on many key techniques of Industry 4.0 concepts, such as CPS, IoT, Big data, machine learning, and cloud computing. In the following sections, each layer with the most used techniques is described.

4.3 Data Acquisition Block

To construct the PdM system for an industrial machine, first, a data acquisition module is developed with three sub-blocks, as shown in Figure 4.2.

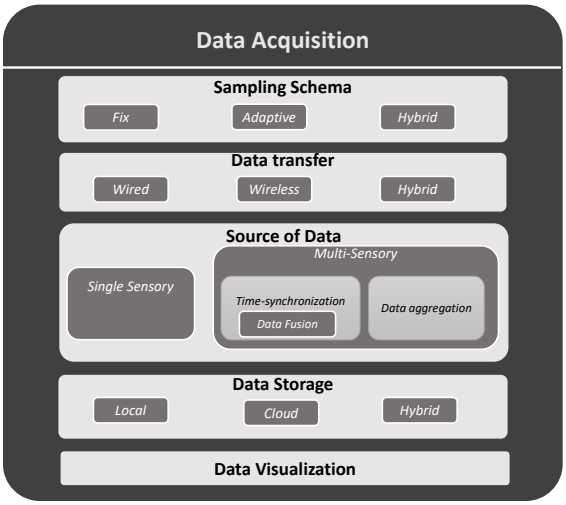

Figure 4.2: Data acquisition block

This block is utilized to gather, store, and forward the collected data. Data acquisition is one of the essential steps in the PdM process as it can affect the quality of data and the performance of PdM. The desired design for the data acquisition block is dependent on the type of engineered systems and their operating conditions. There are four decision variables for an optimal design of data acquisition, and these are as follows:

- Sampling schema
- Data transfer (wired, wireless, hybrid)
- Source of data (single-sensory, multi-sensory)
- Data storage (cloud, local, or hybrid)

In this building block, we also consider the visualization functionality called data visualization. Since we are looking for abnormalities, we can always get hints from raw data visualization. The sampling frequency method can be different. It depends on the signal types and their variation in time (in an industrial application, we are working with time series data). For example, when changing the signal is not observable, using a simple fixed sampling frequency with a longer interval is preferred.

Similarly, in the case of a signal with abrupt changes, a higher sampling frequency is better. Therefore, an adaptive sampling method can be utilized to optimize bandwidth usage. In addition, a combination of fixed and adaptive sampling can also be an optimized solution.

Since the use case is of an industrial environment, the location and accessibility of the corresponded equipment are important for collecting sensory data. This equipment can be wired-, wireless- and a hybrid. Furthermore, accessibility, sensor expenses, accuracy, and sufficient data transmission speed are parameters included in deciding the type of data transferring method to have.

Data acquisition can be single-sensory or multi-sensory. In many PdM scenarios, it is needed to investigate the impact of several sensor data on system failure and not just a single sensor data. With the help of company experts, it can be decided which parameters may impact more on the system breakdown. Moreover, for investigating the most important parameters, some mathematical techniques can be applied to find the correlation between parameters; this helps with understanding the healthy and unhealthy condi-

tions of the system. A significant concern in the multi-sensory method is how to align all the data coming from different sensors. For instance, pressure-, temperature- and vibration data should be collected approximately simultaneously to analyze the effect of these parameters on the machine's health condition. Hence, data aggregation and data fusion techniques should be applied to make a robust and consistent dataset (specifically when we are using dynamic sampling frequency). Nevertheless, data collection methods are not defined in this framework since they depend on the available ICT infrastructure at the company or user preferences.

4.3.1 Data Storage Block

Data storage block involves storing the data in storage mediums after collecting the data from sensors. There are different storage methods (e.g., cloud or local) that depend on the user's preference and infrastructure availability. The data can be stored in a database locally or in the cloud and then transmitted to the control and monitoring center through a wired or wireless network. An industrial IoT application harvests a high volume of time series data. Therefore the architecture of the storage module should be designed in terms of capacity, accessibility, and cost. For handling big data, we can consider distributed architecture (e.g., edge and cloud) and also popular file/data format for storing a dataset.

4.4 Data Preprocessing Block

The data preprocessing step can effectively clean the raw data, reduce the data dimension, and store it back in the warehouse for knowledge discovery. Therefore, massive data can be converted into features or statistical values as input variables of the data mining process [92]. Raw data measured from sensors can be processed to generate more convenient features that represent the health states of a system.

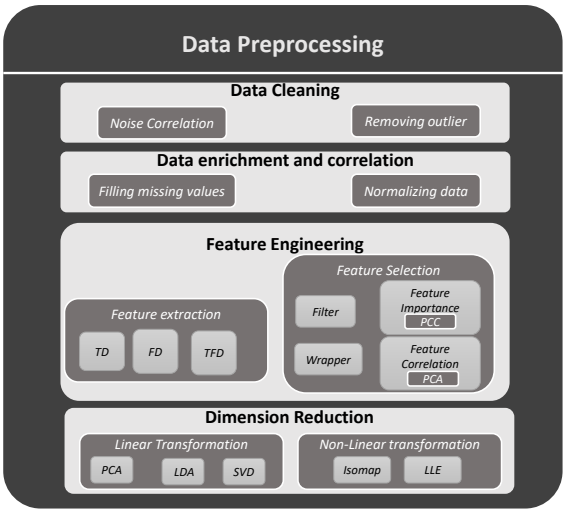

Figure 4.3: Data preprocessing block

As presented in Figure 4.3, the data preprocessing step can be divided into four detailed steps:

- Data cleaning
- Data enrichment and correlation
- Feature engineering
- Dimension reduction

For a given situation, each step can be selectively applied while data preprocessing.

4.4.1 Data Cleaning

Data cleaning is the procedure of detecting and correcting corrupt or inaccurate records from the database by smoothing noisy data, identifying or removing outliers (points with behavior quite different from the others), and resolving inconsistencies [92]. The handling of noisy or corrupted data includes identifying features beyond the expected standard (called outlier) or other unplanned behaviors. The root causes are diverse, such as measurement variations of equipment or human interference, among others. The solution can simply remove the value if the sample is recognized as an irregularity or resolving it by utilizing binning, clustering, or other methods. Outlier elimination is the simplest solution to enhance the quality of ML training. Although, before removing such a value, it should be carefully studied that if these outliers are happening because of a new measurement approach or it is just noise [80]. An outlier may represent an opportunity for a discovery, which might research new directions not noticed before. Data inconsistency correction is also a part of the cleaning task. Inconsistency is the presence of incompatible values in the same attribute, which in many cases may be caused by the combination of multiple databases.

An example will be if each database uses various scales to measure power. One table could utilize watt and the other kilowatt. A combination of the values would be inconsistent. The inconsistency correction can be done manually, automatically, or even considering other types of normalization (see Data enrichment and correlation in the following section).

4.4.2 Data Enrichment and Correlation

This preprocessing step consists of two main steps: filling missing values and normalization. Filling missing value deals with the vacancy of data, which happens if one or more features do not exist. The problem can be resolved by eliminating the attribute or eliminating the entire sample if this may create a problem with other sample attributes. There exist other solutions with more complicated techniques, such as assigning the mean, a moving average, or even the minimum or maximum values to those missing values [80]. Raw data needs to be scaled to avoid particular variables dominating the predictive method. Autoscaling standardizes the variables in a way that ensures each variable is given equal weight before the application of the detection method [57]. For instance, it will avoid an excessive difference between the maximum and minimum values (e.g., 0.001 and 10000). It is important for many algorithms such as the neural networks and K-Nearest Neighbourhood (KNN) algorithms. Here, we always execute a min-max normalization [84].

4.4.3 Feature Engineering

Manufacturing sensors monitor production processes that are usually happened periodically and distinguished by a particular span. The feature engineering module is designed to transform and process the raw sensor data to extract the signal's main feature. A feature engineering component preprocesses incoming raw data into values with more significant meaning for predictive analytics, for example, the current average value instead of simple amplitude. This block consists of two main modules: Feature generation and Feature selection.

Feature Extraction

Generally, the features can be extracted from three domains: (i) time domain, (ii) frequency domain, and (iii) time-frequency domain.

Time domain data processing relates to feature extraction of the time series data, for instance, the peak, mean, and Root Mean Square (RMS) value [94]. For example, Song et al. [95] discovered a linear relationship between the AR parameter and the surface roughness and distinguished the vibration time series using the autoregressive moving average model. Campatelli and Scippa [96] predicted the cutting force coefficients by analyzing the time-domain behavior of the cutting force signal. Ertekin et al. [97] measured the RMS of the AEDC signal, which was detected as the most sensitive feature for wearing of the tool. The average RMS feature of the current signal also adds to the estimation of tool wear [97].

Frequency domain data processing can extract more inherent features from cyclic data series, mainly when data carries background noise which is hard to recognize in the time-domain. For example, in [98],

Altintas et al. analyzed the cutting force and chatter stability during the dynamic cutting process using Nyquist law in the frequency-domain. The investigation of tool vibrations employing fast Fourier transform was proved an effective means for the prediction of surface roughness [99]. By analyzing the motor current in the frequency-domain, the sensorless automated condition monitoring was achieved for predictive maintenance of machine tool [100]. In the FFT method, the complete frequency spectrum is presented with the average frequency composition. Almost, the sensory data is dynamically varying over time.

Therefore, **time-frequency** domain data processing gives a more reasonable outcome by partitioning the time series data into short time intervals for frequency analysis [101]. Specifically, wavelet analysis and short-time Fourier transform are the two prevalent techniques to analyze cutting force [102], vibration, current, and sound signal. In Figure 4.4, a demonstration of common statistical features is presented.

No.	Name	Equation	No.	Name	Equation	No.	Name	Equation
1	Maximum	$s_1 = max(x_i)$	8	Mean square	$s_8 = \frac{1}{N}\sum_{i=1}^{N} x_i^2$	15	Variance frequency	$F_2 = \frac{1}{N}\sum_{j=1}^{N}(x_j - \bar{x})^2$
2	Minimum	$s_2 = min(x_i)$	9	Root mean square	$s_9 = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}$	16	Standard deviation frequency	$F_3 = \sqrt{\sum_{j=1}^{N}\left((f_j - F_1)X_j\right) \Big/ \sum_{j=1}^{N} X_j}$
3	Median	$s_3 = \begin{cases} x_{(N+1)/2}, N\ is\ odd \\ \frac{x_{N/2} + x_{(N+1)/2}}{2}, N\ is\ even \end{cases}$	10	Skewness	$s_{10} = \frac{1}{N}\sum_{i=1}^{N} \frac{(x_i - s_5)^3}{s_7^3}$	17	Root mean square frequency	$F_4 = \sqrt{\sum_{j=1}^{N}(f_j^2 X_j) \Big/ \sum_{j=1}^{N} X_j}$
4	Peak-to-peak	$s_4 = max\lvert x_i\rvert - min\lvert x_i\rvert$	11	Kurtosis	$s_{11} = \frac{1}{N}\sum_{i=1}^{N} \frac{(x_i - s_5)^4}{s_7^4}$	18	Spectral skewness	$F_5 = \sum_{j=1}^{N}\left(\frac{f_j - F_1}{F_3}\right)^3 S(f_j)$
5	Mean	$s_5 = \frac{1}{N}\sum_{i=1}^{N} x_i$	12	Skewness factor	$s_{12} = \frac{1}{N}\sum_{i=1}^{N} x_i^3 \Big/ \left(\sqrt{s_8}\right)^3$	19	Spectral kurtosis	$F_6 = \sum_{j=1}^{N}\left(\frac{f_j - F_1}{F_3}\right)^4 S(f_j)$
6	Variance	$s_6 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2$	13	Kurtosis factor	$s_{13} = \frac{1}{N}\sum_{i=1}^{N} x_i^4 \Big/ \left(\sqrt{s_8}\right)^4$	20	Spectral power	$F_7 = \sum_{j=1}^{N}(f_j)^3 S(f_j)$
7	Standard deviation	$s_7 = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2}$	14	Mean frequency	$F_1 = \frac{1}{N}\sum_{j=1}^{N} f_j$	21	Wavelet energy	$F_8 = \sum_{j=1}^{N} \omega t_\phi^2(j)/N$

Figure 4.4: Summary of statistical features in time-domain and frequency-domain according to [103, 104]

Feature Selection

Feature selection is choosing and eliminating given features without changing them. The task of feature selection, which is known to be NPhard [105, 106], entails the search for an optimal subset of features in such a way that this chosen subset can best represent the original data set. In addition to removing irrelevant and redundant features, the resultant feature subset, with a fewer number of features, will lead to reduced computational cost and simpler models. In feature selection techniques, it is intended to remove features with missing values, low variance, and highly correlated features. In paper [86], the work applies wrapper and filter techniques, and in paper [71], the H2O gradient boosting machine [107] calculates the importance of each feature. Feature importance refers to techniques that assign a score to input features based on how useful they are at predicting a target variable. These techniques can be possible to apply in case of the availability of the labeled data.

Wrapper methods wrap the feature selection process in the classifier itself. The computation involved for the filter methods is relatively straightforward and less intensive compared to the wrapper methods. Filter methods are robust against overfitting when compared with the wrapper methods [86].

4.4.4 Dimension Reduction

We generally do not want to deliver a large number of features directly into a machine learning algorithm as they are expensive to store, causing slow-down computations, large samples are required to avoid overfitting, and in algorithms like K-nearest neighbors, distances in high dimensions are distorted. Extracted condition indicators exhibit a variety of correlations between one another. It might cause inefficient classification of health states since many classifiers require independent features to obtain a satisfactory accuracy. Overcoming this drawback, there are two general approaches which are linear and non-linear methods. The most popular linear methods are PCA, LDA (Linear Discriminant Analysis), and SVD (Singular Value Decomposition). The non-linear ones are isometric mapping (isomap) and locally-linear embedding. The

most used technique in literature such as [57, 76, 108] is PCA. Based on the reviewed literature in the previous chapter, the most suggested method to reduce dataset dimensions is to use the PCA technique. PCA is utilized to reduce the feature vector's dimension space and find the correlation between the features. The central idea of PCA is to reduce the dimension of interrelated features while preserving the variance in the data by projecting the feature vectors onto a new set of variables called principal components [84].

4.5 Predictive Analytics Block

Predictive analytics block is the heart of this framework. With this block, the aim is to predict the correct label (healthy/unhealthy) for new incoming sensor data in the real-time prediction phase. In this step, first, the preprocessed dataset from the previous building block is split into a training dataset and test dataset. In machine learning, the training dataset is utilized for training the model to learn healthy and unhealthy patterns, while the testing dataset is used to validate the model and tune its parameters like the anomaly threshold. Although, there are some challenges we deal with in this step for identifying healthy and unhealthy data.

Despite the ease of collecting data from experiments from industrial machines, it is not feasible to run industrial equipment for years just to collect wearing data. The only allowable and available data from such industry machines for some experimental study is the data recorded during regular operation. It is also impossible to simulate the machine failure mode as it might cause permanent damages and most likely violate the machine's guarantee. These are real scenarios that can happen everywhere, and it needs to be deal with constraints and restrictions.

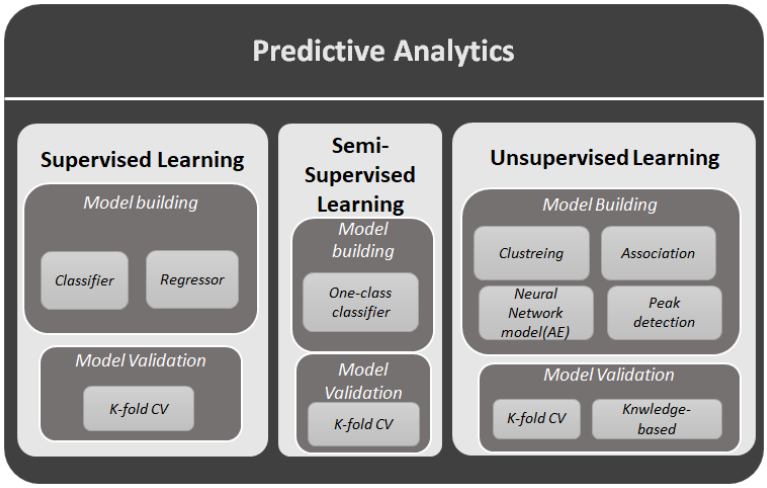

Figure 4.5: Predictive analytics block

Therefore, in the proposed framework, the Predictive Analytics Block as presented in Figure 4.5 is divided into three main sections: (i) supervised learning, (ii) semi-supervised learning, and (iii) unsupervised learning.

In machine learning, a contrast has traditionally been made between two major tasks: supervised and unsupervised learning [109]. In supervised learning, the algorithm is presented with a set of data points consisting of a given input x and an output value y. Then, the goal is to construct a classifier or regressor that can estimate the output value for previously unseen inputs. Output values in unsupervised learning are not provided. Alternatively, these algorithms reveal an existing underlying structure from the inputs. A well-known example of unsupervised learning, unsupervised clustering, works based on the structure retrieval to map the given inputs (such as vectors of real numbers) to a set of groups such that similar inputs are mapped to the same theme [110]. The semi-supervised learning method is a branch of machine learning that aims to combine these two tasks. Typically, semi-supervised learning algorithms try to enhance performance

in one of these two tasks by employing information generally correlated with the other. For example, in classification situations, some extra data may be used to facilitate the process. For clustering approaches, the learning procedure might benefit from the knowledge that specific data points belong to the same class.

Semi-supervised classification approaches are mainly related to scenarios where labeled data is rare. In those cases, it may not be easy to construct a reliable supervised classifier. This situation occurs in application domains where labeled data is expensive or difficult to obtain, like computer-aided diagnosis, expensive industrial equipment, and drug discovery. If adequate unlabelled data is available and under specific data distribution assumptions, the unlabelled data can help construct a more reliable classifier. In practice, semi-supervised learning approaches have also been implemented to scenarios with no notable shortage of labeled data. If the unlabelled data provide further information relevant to prediction, they can potentially be used to achieve enhanced classification performance [110]. Therefore, our proposed PdM considers these three learning approaches and based on the availability of labeled data, any of these methods can be chosen.

When there is historical data and records about the health condition of the system and failure events, the machine learning model is trained with labeled healthy and faulty data. There are plenty of algorithms to train our model, and this also can be categorized into groups: classifier and regressor. Regression and classification are categorized as supervised machine learning. Both are utilizing the same concept known datasets (referred to as training datasets) to make predictions. Basically, classification aims to predict a label, and regression is about predicting a quantity. One looks for predicting health conditions with a classifier, and with a regressor, one looks for the Remaining Useful Life (RUL) of equipment.

In both supervised and unsupervised techniques for model building, several popular ML algorithms are used.

Table 4.1: Machine learning used in reviewed literature

Machine Learning algorithm	used in PdM (%)
Support Vector Machine (SVM)	24.3
Random Forest (RF)	18.9
K-Nearest Neighbour (KNN)	16.2
Decision Tree (DT)	13.1
Artificial Neural Network (ANN)	10.8
Gradient Boosted Tree (GBT)	8.1
K-mean Clustring, CNN, LR, Gaussian Naive Bayes	5.4
LM, RPART, AE, RNN, Peak detection	2.7

Table 4.1 reveals a preference for some ML learning methods based on the literature reviewed in Chapter 3. For example, the most employed ML algorithm is SVM, RF, KNN, followed by neural network based methods (i.e., ANN - Artificial ANN). Other ML algorithms are used for a specific use-case; for example, in [67] due to lack of labeled value, an auto-encoder algorithm is employed. Furthermore, in [82] peak detection technique is used for unsupervised machine learning. In the next subsections, we describe the main characteristics of the most used ML methods and how they are employed in PdM applications.

4.5.1 Common Supervised Learning Algorithm

Support Vector Machines

SVM is a broadly used and known ML method for performing classification and regression tasks because of its high accuracy [111]. SVM is a set of supervised learning approaches that perform regression analysis and pattern recognition. Initially, SVMs were non-probabilistic binary classifiers. However, now they are also employed in multi-class problems. Here, SVM creates n-dimension hyperplanes that divide data ideally into n groups/classes. One of the main characteristics of SVM is the high precision in the separation of different classes of data, being able to determine the best point for separating classes of data [112].

Random Forests

RF is an ensemble learning algorithm composed of multiple Decision Tree classifiers, and the category of its output is determined jointly by these individual trees [113]. The RF is provided with many significant advantages. For instance, it can handle high dimensional data without feature selection; trees are independent of each other during the training process, and the implementation is relatively simple. In addition, the training speed is usually fast, and at the same time, the generalization ability is strong enough.

4.5.2 Common Unsupervised Learning Algorithm

K-means

The k-means model is a common clustering algorithm that employs an unsupervised approach to determine a set of clusters [114]. The primary purpose is to discover the k partitions (or clusters) of the dataset so that "close" samples to each other are correlated with the same cluster, and "far" samples from each other are correlated with different clusters [115]. The k-means algorithm is easy to implement. In addition, it presents good performance and handles large data sets (as long as the number of clusters k is small). Moreover, it can change the centers of the clusters by retraining when new samples are available. Another essential feature of the k-means algorithm is that it tends to minimize inter-class variance and increases the extra class distance [116].

Artificial Neural Networks

Artificial Neural Networks (ANNs) are intelligent computational techniques inspired by the biological neurons [117]. An ANN is composed of several processing units (nodes or neurons) with a relatively simple operation. Communication channels usually connect these units with an associated weight; they only operate their local data indicated through their connections. The intelligent behavior of ANNs comes from the interactions between the processing units of the network. ANNs are one of the most common and applied ML algorithms, and they have been proposed in many industrial applications, including soft sensing [118] and predictive control [119]. Firstly, their main advantages are that no expert knowledge to make decisions is needed since they are based only on the historical data (as the k-means model). Secondly, even if the data is inconsistent, they do not suffer degradation (i.e., ANNs are robust). Thirdly, building an accurate ANN for a particular application can be implemented in real-time without changing its architecture with every update. However, some disadvantages of ANNs are that, firstly, networks can reach conclusions that deny the rules and theories established by the applications. Secondly, training an ANN can be time-consuming. Lastly, they are the "black box" method (it is impossible to know why the ANN model has reached an output prediction), and a vast data set is needed for an ANN to learn correctly.

After training the model by train dataset, it is time to validate the trained model with the test dataset in the model validation module.

4.5.3 Model Validation

While training a model, we need to know whether it works and can trust its predictions. Could the model hardly memorize the data it is fed with, and therefore unable to make good predictions about coming samples or samples that it has not seen before? in this block, we want to find a solution for this challenge after training the model.

Methods for evaluating a model's performance are divided into two categories: namely holdout and Cross-validation. Both methods use a test set (i.e., data not seen by the model) to evaluate the model performance. It is not recommended to use the data we used to build the model to evaluate it because our model will simply remember the whole training set and consistently predict the correct label for any point of the training set known as overfitting.

The purpose of the evaluation is to test a model on different data than what it was trained on it. This provides an unbiased estimation of learning performance. In this approach, the dataset is randomly divided (for time-series data, it is not randomly) into three subsets: **Training set** which is a subset of the dataset used to build predictive models, **Validation set** that is a subset of the dataset utilized to evaluate the

performance of the trained model. It gives a test platform for tuning a model's parameters and selecting the optimized performing model. All modeling algorithms do not need a validation set. The last one is **Test set**, or unseen data, which is a subset of the dataset used to evaluate the possible future performance of a model. If a model fits the training set much better than it fits the test set, overfitting is probably the cause. The approach is helpful because of its simplicity, flexibility, and speed. Nevertheless, this technique is often related to high variability since differences in the training and test dataset can lead to meaningful differences in accuracy estimation.

Cross-validation is a technique that comprises partitioning the original dataset into a training set that is utilized for training the model and an independent set that is used to evaluate the analysis. The most common cross-validation technique is **k-fold cross-validation**, where the original dataset is partitioned into k equal size subsamples, called folds. The k is a user-specified number, usually with 5 or 10 as its preferred value. This is repeated k times, such that each time, one of the k subsets is used as the test set/validation set, and the other k-1 subsets are put together to form a training set. The error estimation is averaged over all k trials to get the total effectiveness of our model.

For example, when performing five-fold cross-validation, the data is first partitioned into five parts of (almost) equal size. Then, a series of models are trained. The first model is trained by utilizing the first fold as the test dataset, and the remaining folds are utilized as the training dataset. The procedure is repeated for each of these five divisions of data, and the accuracy estimation is averaged over all five cases to get the total effectiveness of our model. In this procedure, every data gets to be in a test dataset just once and k-1 times gets to be in a training dataset. This significantly decreases bias, as we are using most of the data for fitting. It also notably reduces variance, as most of the data is being utilized in the test set. Changing the training and test sets also add to the effectiveness of this method.

Since the most used technique in literature is Cross-Validation (CV), and considering the advantages of this method, we used this technique in our PdM framework. This approach can be computationally expensive, nevertheless it is the right approach if the number of samples is small [84].

4.6 Result Evaluation Block

Model evaluation metrics are required to quantify model performance. The choice of evaluation metrics depends on a given machine learning task (such as classification, regression, ranking, clustering, topic modeling, among others). The performance of the ML model should be evaluated in terms of predictive

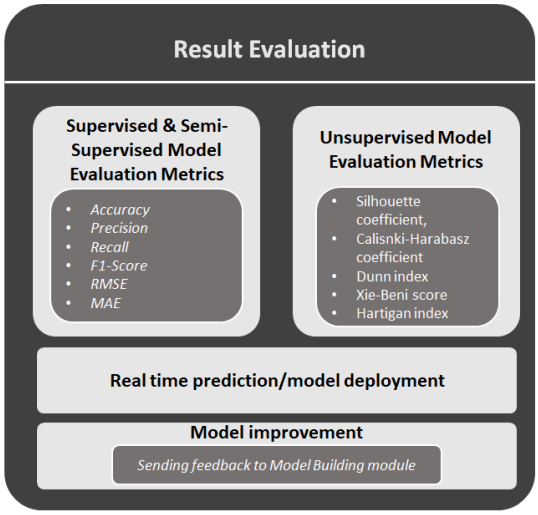

Figure 4.6: Result evaluation block

accuracy and model interpretability. Predictive accuracy of the proposed approach was performed according to the following measures: Accuracy, Recall, Precision, F1-score. These metrics are calculated by

Eqs. 4.1–4.4, respectively. True Positives (TP) indicate the number of times the classifier correctly predicts the normal stage of operation of the equipment. False Positives (FP) corresponds to the number of times the faulty signal was misclassified as normal. A True Negative (TN) is an outcome where the model correctly predicts the unhealthy condition of the asset. False Negative (FN) informs how many times the model incorrectly predicts the unhealthy condition for the equipment. The accuracy is the percentage of correct predictions, not taking into consideration the difference between the normal stage of operation and the faulty stages. Metrics such as precision can be used to evaluate the performance of the approach. A system with many false alarms leads to a discredited system. Hence, the system also needs a good recall to make it more reliable, avoiding permanent damages to the pump [72]. The formulas are as follows [120]:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{4.1}$$

$$Recall = \frac{TP}{TP+FN} \tag{4.2}$$

$$Precision = \frac{TP}{TP+FP} \tag{4.3}$$

$$F1-Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4.4}$$

Where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.

In further steps, when a company in a situation to choose the best model for deploying in the production line, the essential question is how to achieve the most accurate model. For answering this question, first, we should discuss another aspect of manufacturing. We need to look for the business challenge a company is trying to solve using the model. For example, if a model's accuracy is 99%, it does not mean this model is the most accurate model for our business challenge. The other metrics introduced earlier, such as precision, recall, and F1-score may be more useful in our business case. For example, if we do not have a correct measurement of false positives, the result shows the system is in bad condition, which is not; it will cause sending an alarm to shut down the production line. Depends on the cost of shutting down and restarting the production line, this false positive causes financial loss for a company. For tackling this issue, precision is an effective indicator. If we look back at the precision formula and analyze how it is calculated based on the confusion matrix as presented in Figure 4.7, finally we have this formula for precision as follows:

		Predicted	
		Negative	**Positive**
Actual	**Negative**	True Negative	False Positive
	Positive	False Negative	True Positive

Figure 4.7: Confusion matrix with considering precision calculation

$$Precision = \frac{TP}{TP+FP} = \frac{TP}{TotalPredictedPositive} \tag{4.5}$$

Precision shows how precise a model. It means how many predicted positive are actual positive. Therefore, we can conclude that precision is a good measure to determine when the costs of FP are high.

Another business challenge is FN. Imagine the model predicts a normal/healthy condition for the machine, which is not valid. The consequence of this predicted negative can be very destructive for sensitive and expensive industrial equipment. For solving this issue, we need to utilize another metric which is recall. If again look at the confusion matrix as depicted in Table 4.8, we can rewrite the recall formula as follows:

		Predicted	
		Negative	**Positive**
Actual	**Negative**	True Negative	False Positive
	Positive	False Negative	True Positive

Figure 4.8: Confusion matrix with considering recall calculation

$$Recall = \frac{TP}{TP+FN} = \frac{TP}{TotalActualPositive} \tag{4.6}$$

Recall measures the number of TP among the actual positive. Therefore, if there is a high cost due to FN, recall is the relevant and practical metrics to select the best model.

Most business challenges involve detecting more TN, and accuracy performs quite well to indicate TNs. However, FP and FN commonly have business costs; therefore, F1-score might be a better metric to employ if we need to find a trade-off between recall and precision. In addition, F1-score is a more useful metric in the situation of a large number of actual negatives). Therefore, in industrial cases, we can not risk and evaluate our models just by one of these metrics, such as accuracy. Therefore, we should consider the company's business challenges when utilizing these evaluation metrics to choose the best mode.

Moreover, there are some other evaluation metrics such as Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) [121]. These are used to calculate the distance between the input vector X (which is the label of data) and the predicted vector Y. RMSE and MAE are defined as below equations. Let $X = (x_1, x_2, ..., x_n)$ and $Y = (y_1, y_2, ..., y_n)$.

$$RMSE(X,Y) = \sqrt{\frac{\sum_1^n (x_i - y_i)^2}{n}} \tag{4.7}$$

$$MAE(X,Y) = \frac{\sum_1^n | x_i - y_i |}{n} \tag{4.8}$$

For example in PdM result analysis n is the number of samples for testing. X is the real RUL of the equipment, and Y is the predicted RUL of the equipment. The smaller the RMSE and MAE are, the more accurate the prediction results are and represent the better performance of the model.

So far, we have introduced the evaluation metrics for supervised learning. There are, however, different evaluation metrics that we can apply for unsupervised machine learning, such as the Silhouette coefficient, Calisnki-Harabasz coefficient, Dunn index, Xie-Beni score, and Hartigan index [122]. Since most unsupervised learning methods want to identify different clusters among data (e.g., KNN), applying these metrics aims to investigate if the model can find diverse clusters. For example, Silhouette Score measures how close each point in one cluster is to points in the neighboring clusters, thus helping in figuring out clusters that are compact and well-spaced out. In addition, for specific unsupervised approaches such as the peak detection method, the experts' decision should evaluate the model's performance. Finally, these metrics have been used to evaluate the performance of unsupervised learning techniques in predictive maintenance work.

4.6.1 Model Deployment

Deployment is the approach by which it is possible to integrate a trained machine learning model into a production environment to make practical business decisions based on actual data. In the machine learning life cycle, this phase can be one of the most challenging parts. Primarily, an enterprise's IT systems are not compatible with conventional model-building languages. Thus, for reaching a fully integrated framework, we need to force programmers and data scientists to rewrite some parts of their work.

For the purpose of start using a model for practical decision making, coordination between software developers, IT teams, data scientists, and business experts is needed to guarantee that the model works

reliably in the company's production environment. This reveals a major difficulty due to a discrepancy between the programming language used for a machine learning model and the languages the production system can understand. Rewriting the model's code needs quite a long time and can change the timeline and schedule in terms of business. In order to gain the most value out of machine learning models, it is essential to deploy them integrated into production so that an organization can start applying them to make practical decisions.

In order to enhance the accuracy of the machine learnings' results, feedbacks of models need to be regularly updated. Therefore, we can design another module called Model Improvement. This module can collect feedback after the deployment is done to improve the fault prediction model. The feedback is gathered from the facility management team (the users) to report false alerts or undetected failures. This feedback is collected and stored via a maintenance and management system. After the feedback is collected, a procedure for error evaluation is carried out where the model's errors are inspected. Following this, the model is then updated using new training data, and its parameters are tuned to reduce the error ratio. The improvement of the model is not a systematic approach; the procedure of updating the model using the collected feedback should be done via a proper schedule and by a machine learning specialist. In order to give the facility management team a quicker response, an anomaly threshold can be designed as an external parameter where the user can directly change the setting without a need for a total update to the model. If the anomaly score (which is an indicator of the model's accuracy) exceeds a defined threshold, it can send an alert to the maintenance management system that the model needs to be updated.

4.7 Decision Making Block

The primary purpose of this module is to visualize the result of machine learning and provide an optimized strategy according to the achieved result. Generally, a diagram of Key Performance Indicator (KPI), also

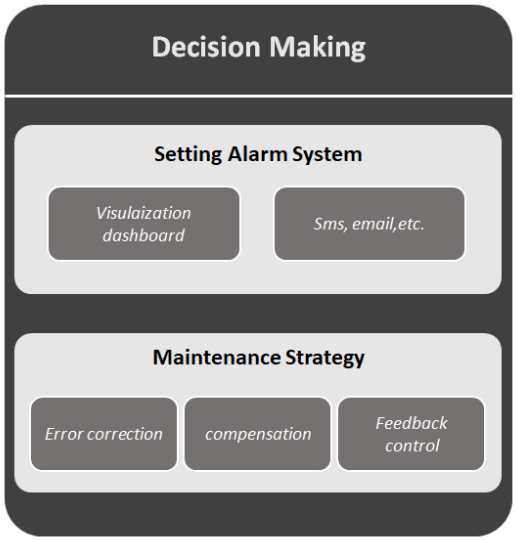

Figure 4.9: Decision making block

called a spider chart, can be used for presenting the situation of equipment and also some visualization tools such as Grafana [123] and Power BI [124]. The equipment conditions can be defined in several levels, such as green, yellow, and red. Green indicates a healthy condition, yellow as a need for inspection, and red for a critical situation. In the framework, the KPI may be formed according to the outputs of the ML result. The diagram will enable operators or managers to evaluate the performance visually, and subsequently, an optimized maintenance schedule can be provided according to the evaluation result. The Maintenance strategy module can do maintenance planning and scheduling optimization with the help of integrating the results into the equipment's control system.

In this module, maintenance will be implemented after the decision makers choose the strategy for maintenance. It can be considered as the purpose of the CPS. The physical world is transferred into the virtual one for communication, computation, analysis, and decision making via the previous modules. In this module, reaction to the physical world according to the results of those modules and to further implement maintenance to achieve a particular purpose is followed, for example, in the process of minimizing the cost of maintenance, realizing the zero-defect manufacturing, or reducing breakdown. Moreover, this module may also include error correction, compensation, and feedback control, which is based on the results from the Predictive analytics module to continue to run the equipment and process in a normal condition. Different techniques can be used to correct and compensate for the errors.

The next chapter presents the investigation of the proposed PdM framework on a printer machine at the NTS group as a use case.

Chapter 5

Case Study Implementation

In this chapter, a use case implementation of the proposed PdM framework is presented. The selected use case is an Ink Delivery System installed in a printing machine. This ISU has several components such as valves, pumps, heater. The higher probability of pump failure compared to other components in the ISU is an opportunity for executing PdM on it. The pump-related data was collected by NTS'IoT software. The layers of the framework are implemented as described in previous chapters. Then, different comparisons are made to analyze the impact of each of these layers on the predictive maintenance results.

An important aspect of the implementation is that the Spark code-blocks are designed to be generic. With few modifications, we can use these code blocks for different data analytics applications. In our case study, applying predictive maintenance on the printing machine allows us to manage different data analysis challenges in Apache Spark, such as collecting data, feature engineering, dimensionality reduction, regression analysis, and binary classification. In this work, two predictive analytics tasks were handled:

- Predict the RUL of the equipment.
- Predict failure events (faulty values) of the equipment.

The experimental results are obtained on an Intel Core i7 machine with 32 GB of main memory running Windows 10 with Spark (and MLlib) 3.1.1 [125] and MySQL Workbench 8.0.25 [126]. The primary tool that was used for the implementation of the predictive maintenance framework is Apache Spark (version 3.1) [125]. Apache Spark (in Python is also known as PySpark) is an open-source distributed platform for fast data processing.

The rest of this chapter is organized as follows. First, in Section 5.1, the use case is introduced that is a pump installed in the NTS printing machine. As discussed in Chapter 4, the framework is composed of (a) data acquisition,(b) data preprocessing, (c) predictive analytics, (d) result evaluation e) Decision making. In the following sections, each block implementation is presented. Accordingly, Section 5.2 to Section 5.6 present the implementation for our use case per each layer of the proposed framework.

5.1 NTS Use Case Description

As discussed earlier, the proposed predictive maintenance framework is implemented on one of the machines of NTS Group, called the digital printing machine, in this work. Figure 5.1 depicts the architecture of the printing machine, in which multiple levels such as communication and control levels are identified. First, sensor data are transmitted to the Digital Signal Processing(DSP) boards then, data packets are sent to the PC for further monitoring and analysis. In addition, NTS has a plan to initialize cloud communication for conducting advanced data analysis such as PdM.

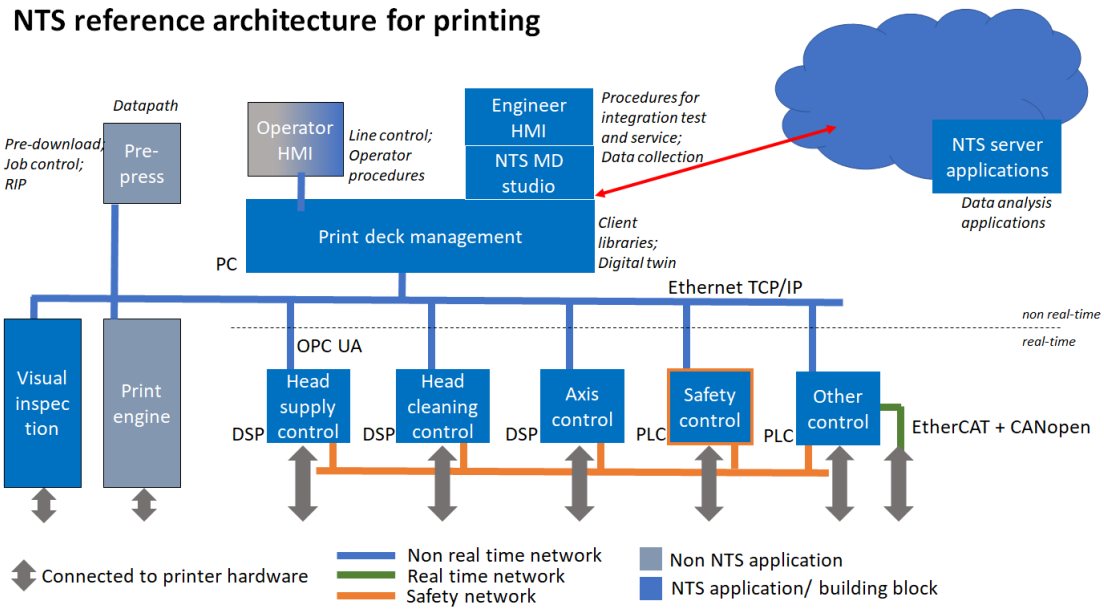

Figure 5.1: Architecture of digital printing machine of NTS Group

5.1.1 NTS Printing Machine Communication

DSP control board is utilizing a processor from Texas Instruments, a *T.I. TMS320F280049C* and provide a closed control loop for ink supply control. Considering DSP to the PC, different equipment from different vendors should work together under the same and efficient communication protocol. Therefore, at this level, NTS decided to utilize OPC UA [127] and its delivering raw signal data from DSP to the PC as presented in Figure. 5.2. The Figure shows all the collected sensory data, and in soft real-time, it shows the condition of the printing machine. At the red box of Figure 5.2 a list of all parameters collected from DSP is demonstrated, and at the blue box is a schematic of the control system.

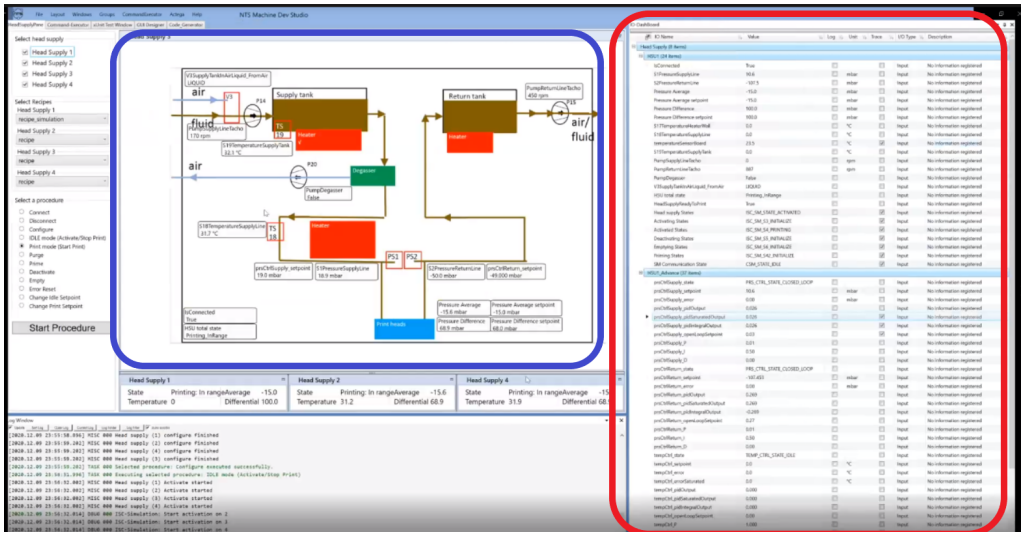

Figure 5.2: DSP to PC communication and monitoring the printing machine

5.1.2 Selected Component for Predictive Maintenance

NTS printing machine, as a target for applying predictive maintenance, has many different components. As depicted in Figure 5.3, with feeding temperature-controlled ink (right side) into supply tanks and a controlling system (left side), it can do the printing. The red box in Figure 5.3 shows four ink supply units of the printing machine.

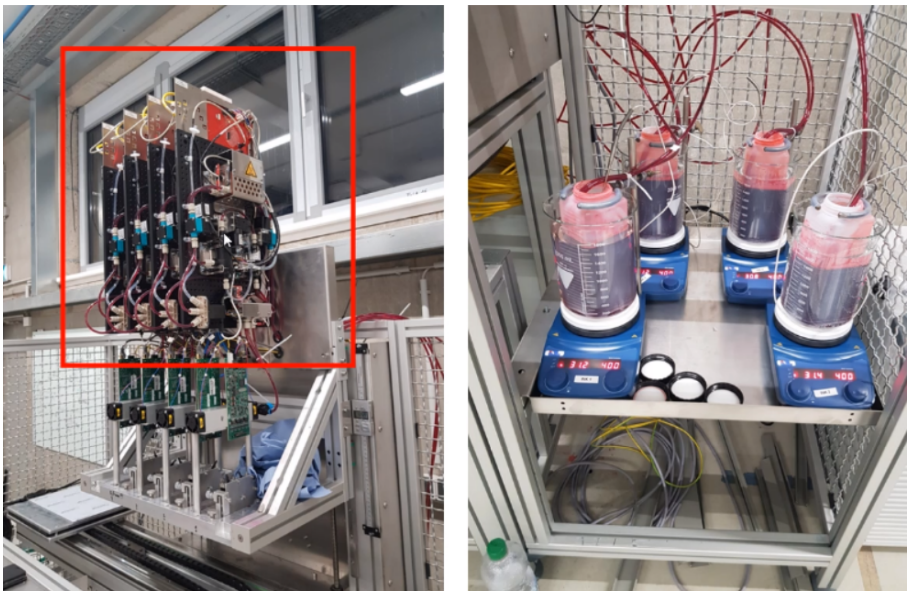

Figure 5.3: Test setup with ink supplies

In Figure 5.4, list of components that are installed in the ink supply unit of the printing machine are shown, such as valves, pumps, pressure sensors, and PCB board. Now we need to investigate the essential component of the ink supply unit. After consulting with NTS, some critical parts of this control system were proposed by them as follows: **Monitor pumps & sensors:**

- Track pump setpoint against other parameters
- Monitor degasser pump & filter performance
- Track changes to machine fingerprints
- Valves: low failure rate, but with 8x6x7=336 valves, this may still be relevant

Develop broad statistical model:

- Link with other data for machine service
- Predict print head performance
- Anticipate maintenance on pumps, filters, and other

Optimize production with print head cleaning

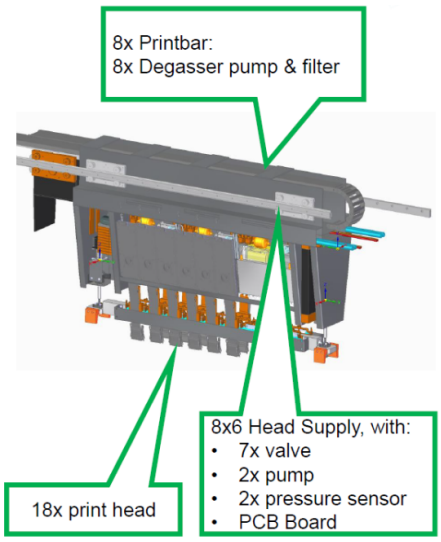

Figure 5.4: Printbar/Ink supply

Since we have relevant information for condition-based monitoring of the pump, monitoring pumps are selected among these potentially predictive maintenance application scenarios.
As discussed earlier, Ink Supply Unit (ISU) is the most important part of the printing machine and can be considered the system's heart. As shown in Figure 5.5, two pumps are installed in ISU (red boxes), supply pump and return pump. These two pumps are working together to provide a predefined ink level in the ink tank.

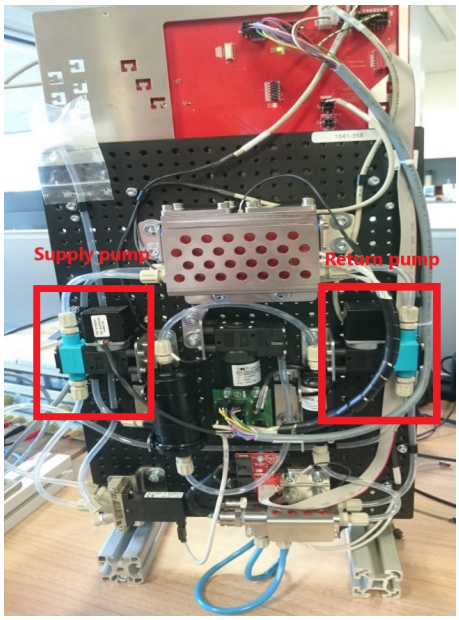

Figure 5.5: Ink supply unit prototype

The supply and return pumps are P14 and P15 in the Figure, respectively, shown with blue arrows. The pumps are connected to the supply tank and return tank, respectively, adjust the Ink level and required pressure in both tanks. Figure 5.6 shows the control system of ISU that controls the printing process.

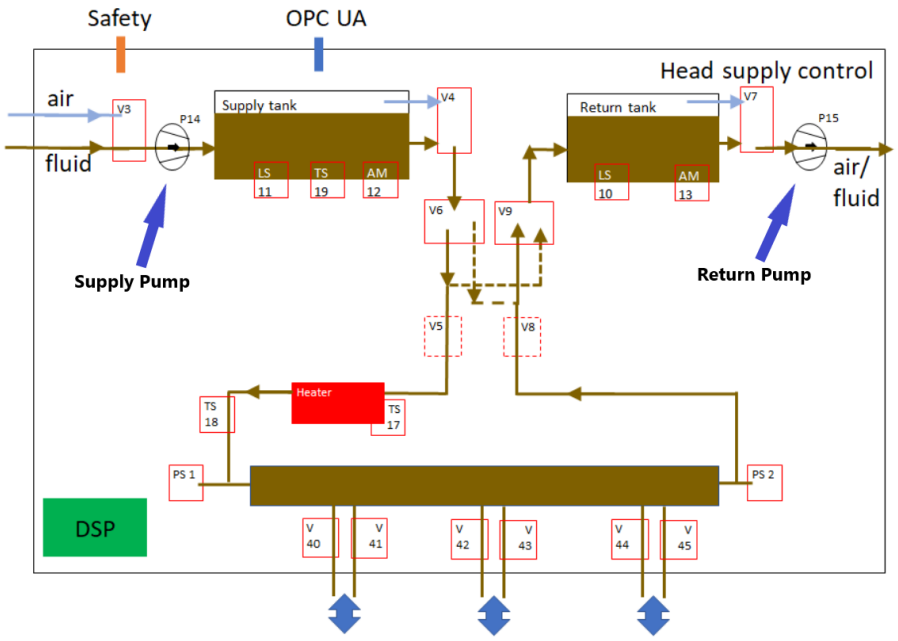

Figure 5.6: Schematic diagram of ink supply control Unit

Pump Characteristics

The pump is a kind of diaphragm liquid pump. A filter is utilized in the NTS printing machine beside the pump to protect it from ink contamination. As it is shown in Figure 5.7, this pump uses Eccentric rod force to move the diaphragm for inhaling and exhaling the fluid. Diaphragm liquid pumps are based

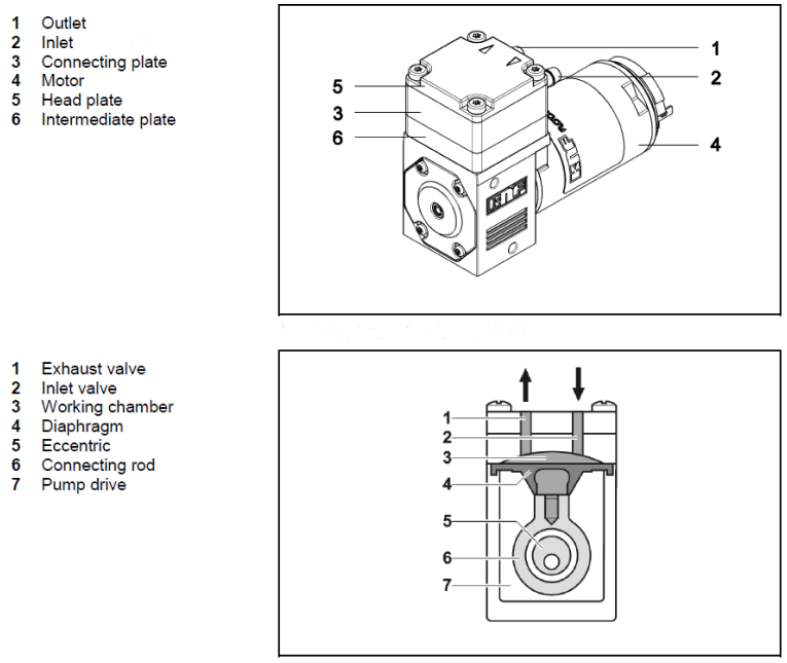

Figure 5.7: Diaphragm liquid pump [128]

on reciprocating displacement pump technology. An elastic diaphragm (4) is moved up and down by the eccentric (5) and the connecting rod (6). During the downstroke, the diaphragm sucks in the medium

through the inlet valve (2). During the upstroke, it forces the medium out of the pump head through the exhaust valve (1)."The diaphragm hermetically seals off the working chamber (3) from the pump drive (7)" [128].

Pump Failure Mode

Even though there are many different types of pumps, most of them share common failure modes; in other words, a phenomenon that occurs leads to inefficiency, failure of a mechanical component, or damage to the entire pump. Every failure mode is an unwanted occurrence that may not directly impact the function of the pump. However, in the longer term, the occurrence of the failure modes without proper action may lead to malfunction, such as leakage [129, 130].

Failure modes can be either hydraulic failures or mechanical failures. Examples of problems of hydraulic failures are cavitation, pressure pulsations, radial thrust, and suction and discharge recirculation [129]. Some mechanical failures are bearing failure, seal failure, inadequate lubrication, excessive vibrations, and fatigue [131]. One of the common problems for this kind of pump is the wearing and bearing problem that the symptom changes in speed and vibration signals. For example, because of misbehavior of the pump, the rod place is changed, and it shows some anomaly in vibration or speed of the motor. This kind of pump is designed for continuous operation. KNF pumps are designed for continuous operation. Quick start and stop cycles may adversely affect the service life of the brushed motors [128].

In NTS printing machine, there are some root causes for pumps' failure, such as:

- Eccentric: If any displacement occurs for the Eccentric Rod, it causes extra vibration to the pump. Therefore, it can be detected by spectrum analysis from captured data of accelerometer (tachometer) and make FFT or WT.
- Power consumption increased due to particle contamination of the diaphragm. Since contamination causes the diaphragm to get heavier, the Eccentric rod needs more power for the movement of the diaphragm/filter.
- Wearing of the diaphragm is one of failure too that cause diaphragm will become less effective Overtime (less ink flow per up/down movement).
- Voltage or current fluctuation.
- Impulse operation: this pump is designed for continuous operation, and impulse operation can affect the pump's life cycle.

We need to investigate the failure root cause of the equipment in our use case pump, to understand better which techniques can help interpret the PdM result. For example, we can employ a knowledge-based technique to evaluate unsupervised learning model performance. In addition, by identifying the different failure root cause, we can categorize data in different failure classes, which helps to have advanced prediction results.

5.1.3 NTS Engineering Tools

So far, the printing machine structure was analyzed to find a good candidate for applying predictive maintenance. The pump installed in the printing machine is selected for this aim. Furthermore, NTS developed its software tools for controlling and monitoring the printing machine. The following necessary tools from NTS are introduced that will help in the PdM application.

The printing process is monitored and visualized in real-time by a software tool called `NTS Machine Dev Studio` that is running on Windows. It is one of the new industrial products of NTS. In the operating room, there is a soft real-time view of all the parameters that need to be controlled in order to print the product in the desired quality. textitNTS machine dev studio enhances industry productivity by visualizing, controlling various machines, sensors, and controllers in a factory. As illustrated in Figure 1.1, the NTS machine dev studio locates in the control level. This tool is a flexible graphical user interface with these capabilities such as performance analysis, calibration, diagnostics, system testing, and software testing.

To be more precise, the NTS machine dev studio interacts with the other Control Level devices within the factory vertically down with 100 Mbit/s Ethernet Fieldbus. This graphical user interface has many

features: live visual updating, tracing/charting, dashboard, record and playtesting, parameter management, logging, and error view. This software is run on PC windows. One of its goals is to understand the controlling process better and detect any failures by providing live graphs. As shown in Figure 5.9, the

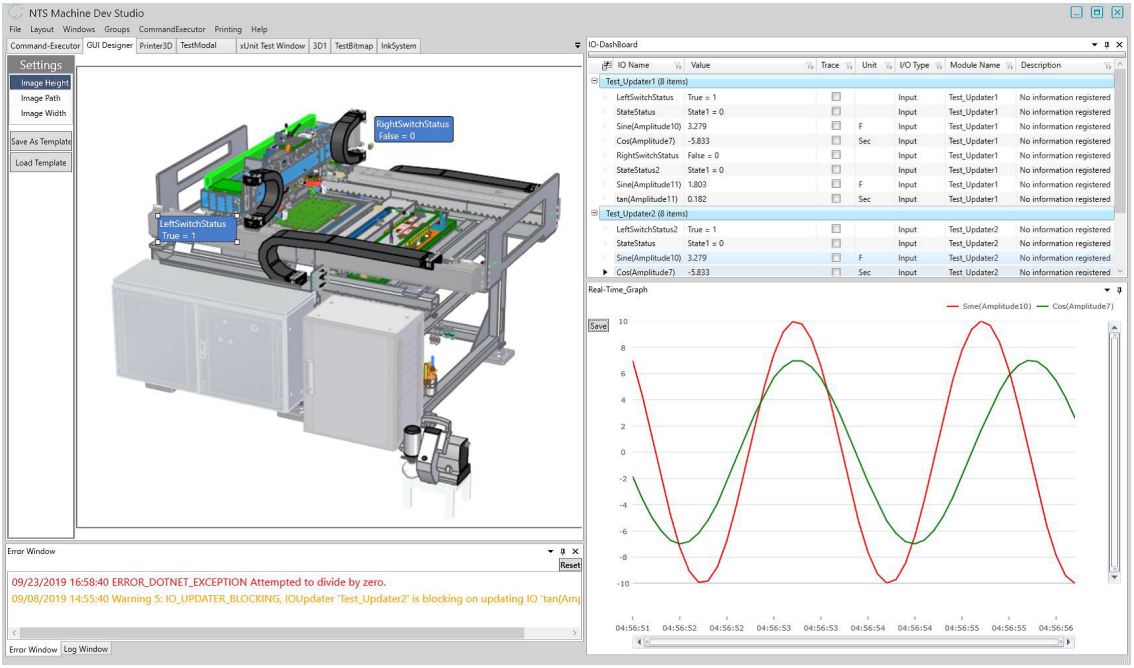

Figure 5.8: NTS graphical user interface

ISU is connected to the PC, and the live sensor data are being collected and monitored. There are many

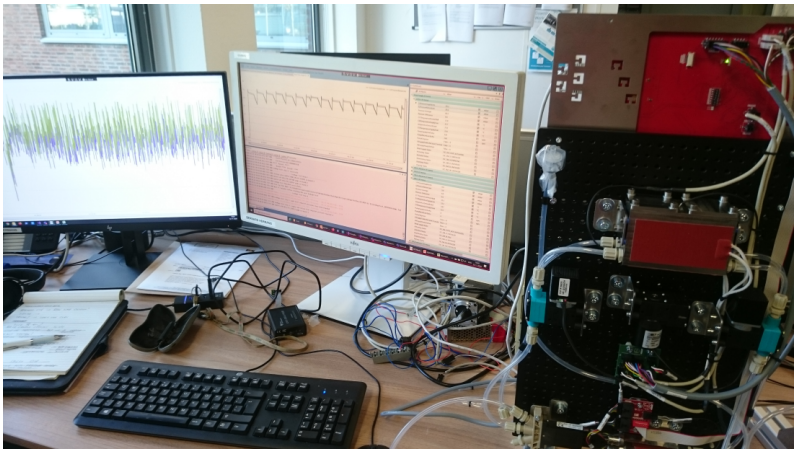

Figure 5.9: Monitoring ink supply unit

parameters related to the controlling process of this unit. Figure 5.10 shows these parameters for each head supply unit in red box.

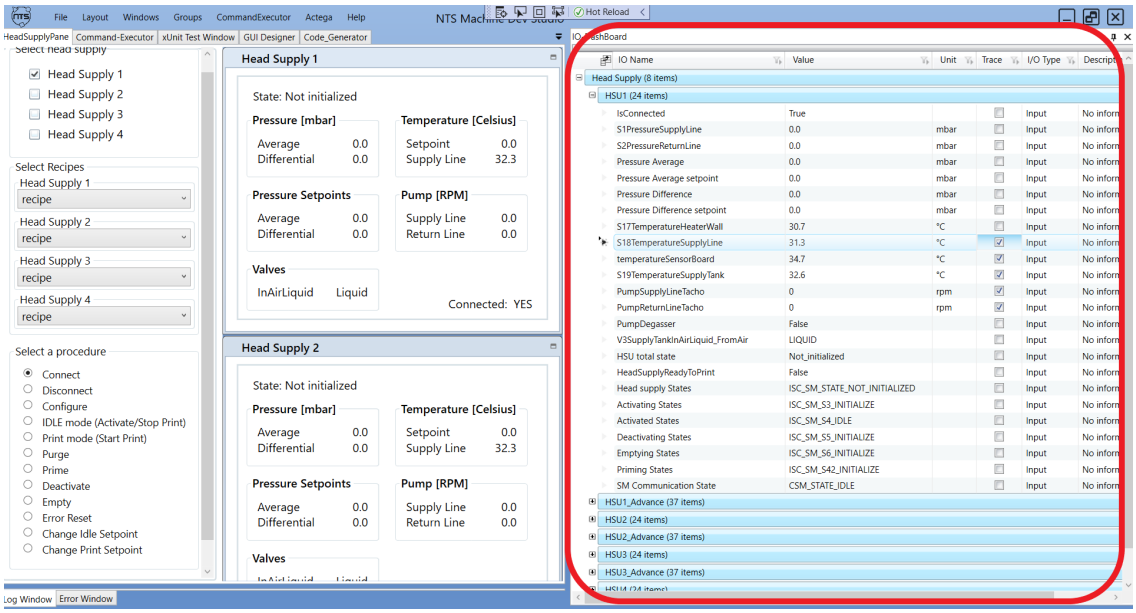

Figure 5.10: NTS software for ink supply unit monitoring

5.1.4 Use Case Development Tools

Based on the hardware/software assessment that we have done so far, we can employ the required tools according to each phase of framework development and implementation. Since the ISU testbed is ready and all the connections and sensors are tested, in the device and control level (as illustrated in Figure 1.1), there is no need for extra HW/SW. The first layer of the framework is the data acquisition, and already data are being transferred from the control level (the DSP board) to a PC with OPC/UA communication protocol. Furthermore, the logs of data are being stored in a soft real-time manner on the PC.

The amount of data being received in one hour from one DSP is 14400 Kbyte (and in the future, there will be 50 DSPs, the data size will be 720 Mbyte/h). Therefore we need to process and analyze data with sufficient speed to avoid the cost of storing useless data for a longer time. Hence, local database capacity also should be calculated in terms of system scalability and reliability. For information, the size of a packet of data from each DSP to the PC in each pooling is 1448 Byte/s.

Networking Tools

For analyzing the communication network in the NTS printing machine, Wireshark [132] tool is employed. With this tool, we can check the bandwidth utilization and latency of the packet delivery to have a better understanding of the flexibility of the size of the data packets and sampling frequency that we can have for our predictive maintenance framework (e.g., if we need data with higher sampling frequency and the system can provide it).

Data Processing Tools

By employing the NTS machine dev studio, monitoring and detecting failure on the system based on a fault threshold mechanism is possible. Although for the data-driven model, we can use this software for collecting data in the first place. For providing a suitable dataset, we need a data log recorder. Therefore, NTS developed another tool called NTS IoT to capture and store data using the MQTT protocol for communication. Now, this communication setup is between a PC and a server that is in NTS. As depicted in Figure 5.11, the IoT application is receiving data from the server.

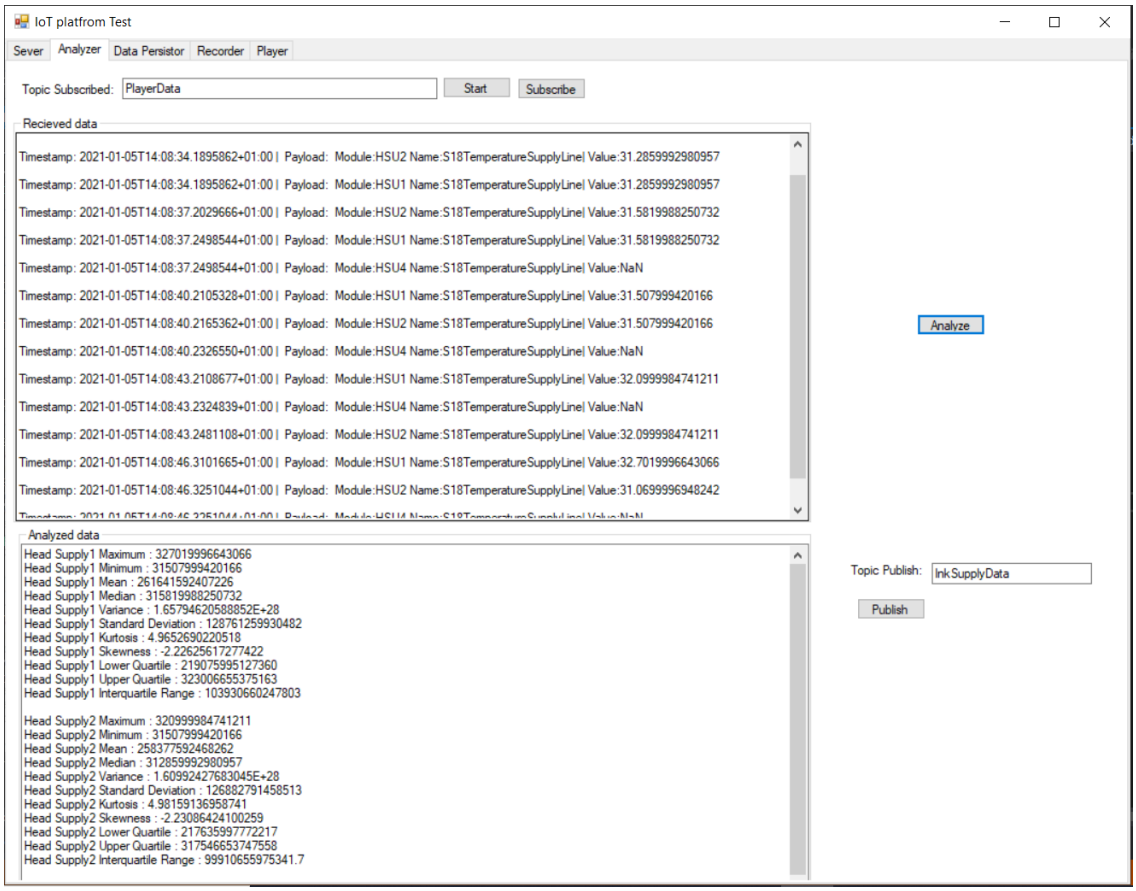

Figure 5.11: IoT platform for PC-Cloud communication

After collecting data, the processing steps should be done. For processing data, there are some popular open-source tools like Apache Spark [125]. The Apache Spark [125] is a fast and general cluster computing system for big data, and it has a unified analytics engine for large scale data processing. Spark and its unique resilient distributed dataset were developed to overcome the limitations of the MapReduce cluster computing paradigm, which forces a particularly linear dataflow structure on the distributed program. Spark runs data science workloads that are up to 100×faster in memory or 10×faster on disk than Hadoop. It has convenient APIs for operating on large datasets and provides high-level libraries, including support for SQL queries, streaming data, machine learning, and graph processing [57]. MapReduce is a programming model and an incorporated implementation for processing big data sets with a parallel algorithm on a single machine with multicore CPUs and a cluster [41].

Modeling Tools

For the machine learning part, Apache Spark MLlib is needed, python, panda package. In addition, the implementation requires the python 2.7 runtime environment and the following python packages: NumPy 1.10.4, scikit-learn 0.17, and pandas 0.17.1.

Data Storage and Visualization Tools

In this case study, we employed MySQL Workbench 8.0.25 [126] database along with Grafana [123] that can be used for results storage and demonstration. Grafana has a powerful processing engine for time series data.

Cloud Provider

For providing cloud infrastructure, now NTS is working with MongoDB [133], which is a general purpose, document-based distribution database built for the cloud era. According to computing power and storage size, we can decide which cloud services we want to utilize.

5.2 Data Acquisition Implementation

In this step, based on the proposed PdM framework, there are five sub-modules inside the data acquisition block, namely Sampling Schema, Data Transfer, Source of Data, Data Storage, Data Visualization. In the following steps, each module is implemented.

5.2.1 Sampling Schema

First, the sampling schema of the system was investigated. After working with the IoT application of the printer machine at NTS, it shows that the sampling frequency is 0.033 Hz (one sample per 30 seconds). It means that the data from the control system is collected every 30 seconds. The collected data is delivered to an MQTT broker that is connected to a local server at NTS.

The sampling frequency is expected to be constant for collecting all kinds of sensory data (i.e., temperature, pressure, speed); however, there was a dynamic frequency algorithm for some of the sensors in the NTS software. For example, the sampling rate changed for the tachometer sensor would reach 10 Hz, and at the same time, pressure and temperature were constant at 0.033Hz.

This sampling frequency behavior (combination of fixed and dynamic style) in multi-sensory methods is problematic for creating a suitable dataset for machine learning applications. In the multi-sensory method, data from multiple sensors should be collected approximately at the same time to be feasible for use in Machine Learning algorithms. One solution is to align the data using techniques such as *Interpolation* [134]. *Interpolation* is a mathematical method that fits a function to the data and uses this function to extrapolate the missing points. The most straightforward type of Interpolation is linear Interpolation, which fills in the mean of data points on either side of a missing data point in place of the missing data [134].

In our case, the problem is solved by changing threshold variables in the dynamic frequency algorithm and fixing the sampling frequency to a constant. Currently, the values are receiving from the NTS IoT applications in good order and simultaneously.

5.2.2 Data Transfer

An OPC UA [135] binary as a data encoding and UA TCP as a transport protocol is utilized for data transferring from DSP to monitoring/controlling NTS tool (namely NTS Machine Dev Studio). Figure 5.12 shows examples of read/write OPC UA messages recorded and displayed with the Wireshark tool. These messages are structured into a header and a message body. The message header contains network information that is interesting for packet filtering, while the message body contains the parameters, which is highlighted in dark blue, shows the binary payload of the selected OPC UA read and write message requests. In the NTS printing machine, the NTS machine dev studio as an OPC UA client [135] requests data from DSP as an OPC UA server every 250 ms. The volume of data is approximately 1Kbyte in each transfer. Therefore, the communication rate is 4Kbyte/s. After analyzing the network and capturing the data packet with running several tests in normal conditions of the NTS printing machine (sending/receiving simulated data), the average delay measured by Wireshark is 16.66 ms. This delay is acceptable from the monitoring tool's perspective; since this graphical user interface is employed, it can provide sufficient condition monitoring information for the operator and maintenance engineers.

Another connection is happening between a server at NTS (storing data that are coming from DSP) and NTS IoT software. NTS IoT tool collects data every 30 seconds, which is mentioned earlier in 5.2.1.

The data transfer method for the printer machine is all wired, and it is using enough bandwidth. Therefore, for this sub-module, everything is suitable to continue with it.

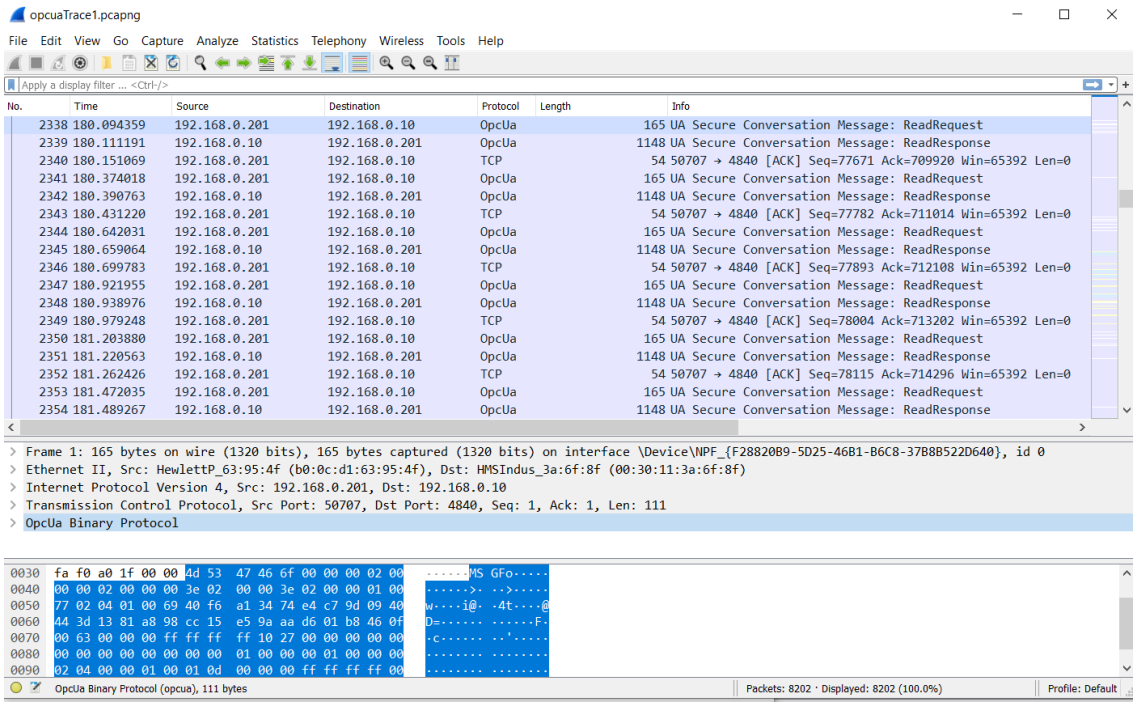

Figure 5.12: Network sniffing results

5.2.3 Source of Data

The next step is to choose between a single-sensory or multi-sensory approach, ensuring that the relevant sensory data is collected. Before selecting the approach, we should investigate the equipment, and whether the relevant data for corresponding equipment exist, then we can decide between single-sensory and multi-sensory approaches. As earlier mentioned, after consulting with NTS and tracking some equipment failures, the decision was made to work with pumps installed in the printer machine's Head Supply Unit (HSU). According to Table 3.3, other similar tasks have been done with the multi-sensory method. Therefore, it is decided to follow the multi-sensory approach. The sensitivity of the parameters on the condition of the pump is analyzed. There are several parameters, such as the pressure of pipes connected to the pumps, the temperature of the liquid inside the pipes, and the pumps' speed. Hence, there are possibilities to work with single sensory and multi-sensory methods. For this implementation, the multi-sensory method has been chosen.

5.2.4 Data Storage

So far, the decision for selecting the sensor data with high probability related to the pump's failure has been made. In addition, the collecting data is happening with a fixed interval for all types of data. The Data Storage module is investigated to find a proper and easily manageable data architecture in this step. For this purpose, we need to look for the data lifecycle in the proposed PdM framework.

With the help of an MQTT broker, the NTS IoT application sends data to a server that is located at NTS. The raw data can be stored in JSON or txt format. The raw data has many irrelevant data points. Therefore, the first step was filtering raw data to capture relevant data points. Filtering unnecessary data helps avoid extra data processing in the following steps and helps manage the data storage better. After filtering raw data, it is stored in CSV format to be suitable for further processing steps.

For storing the data, different methods are available, such as local storage, cloud storage, or a combination of both. In storage strategy, two factors should be considered: scalability and price. Local data storage is cheaper, although storage management will get more complex when the machine works for several months. In addition, with the growth of data, the company would need to buy extra hardware. On the

other hand, the cloud storage approach can help scalability challenges and has greater remote accessibility. For example, if a machine is installed on a customer site, they can send the data to the cloud, and supplier companies can utilize the data to analyze the machine's performance.

There is a concept in big data called *data lake*. Database and data warehouses can only store data that has been structured. On the other hand, a data lake does not respect data like a data warehouse and a database. Instead, it stores all types of data: structured, semi-structured, or unstructured. In our case, for the implementation of the PdM framework, different data types are involved.

Data Lake

In NTS, MongoDB [133] is configured for storing raw data. In sending raw data (before any filtering), MongoDB is helpful as it is suitable for storing unstructured data. As MongoDB is a NoSQL database, it has a more reasonable price in comparison to SQL database. Generally, NoSQL databases are cheaper than SQL ones because, at the same storage level, SQL has more processing overhead than NoSQL databases. Although price-wise, MongoDB is a better choice, it does not help store and manage structured and processed data as it is a NoSQL database.

Since the preprocessed data (kind of semi-structured data) will be generated during the PdM implementation, we need another type of storage. Moreover, after processing the data and generating some results, we need data storage for structured data such as SQL databases. There are two main approaches for storing raw data, preprocessed data and processed data:

- storing data in parquet file format
  - Local storage of parquet files in a shared drive
  - Cloud storage of parquet files (e.g., Azure blob storage)
- storing data in a database
  - SQL database (e.g., SQL Server or MySQL)
  - NoSQL database (e.g., MongoDB)

For each type of data, we can choose between the above items. In the case of raw data, the NoSQL database from MongoDB is a reasonable choice. For preprocessed data, local/cloud storage of parquet files is a speedy and scalable approach. Considering processed data, storing data in a SQL database such as an SQL server is more convenient to analyze the results, and anyone from different engineering fields can work with these structured data types.

It should be mentioned that the most reasonable approach between the above items is storing preprocessed and processed data in Parquet format in the cloud. Apache Parquet [136] is designed to bring efficient columnar storage of data compared to row-based files like CSV. Apache Parquet is created from the ground up with complex nested data structures in mind. Apache Parquet is built to support very efficient compression and encoding schemes [137]. Parquet files are an excellent choice for storing and reading large data files from disk or cloud storage. Using Parquet files with Apache Spark provides an impressive speed improvement compared to employing CSV files with Pandas [**panda**] when reading the content of large files [136].

As shown in Figure 5.13, raw data is collected with NTS IoT software in a local drive, in a JSON format, and then for storing processed data, parquet format is utilized. Storing in parquet format has its constraints. Companies need data scientists to work with this data format. Using SQL database can help engineers from different fields work and analyze this type of file. On the other hand, utilizing an SQL database is costly, and the software license should be purchased. Furthermore, considering adding more sensory data and analysis techniques, there is no exact estimation for the size of the data that the machine will generate in the future. Cloud services can provide required software licenses and also elastic databases. Therefore, the last approach that seems more reasonable is deploying SQL server in Azure and use one of the subscription methods that Azure is suggesting (e.g., pay as you go). In this condition, one does not need a monthly payment for SQL license, and also, if more storage space is needed, it is straightforward in the cloud to change the storage capacity.

At first, an entirely local implementation of the PdM framework is done to realize better the advantages and disadvantages of on-premise implementation of the PdM framework. For this, we can use a combination of SQL database and parquet format files locally.

Since for PdM, a high volume of data needed, one of the primary purposes of this work is to investigate software tools to provide scalability for the PdM procedure. One of the most popular processing engines is Apache Spark [125]. For the implementation of the proposed framework PdM, Apache Spark version 3.1.1 is employed. Apache Spark is a data processing engine to efficiently processes a massive amount of data. If the number of datasets increased, we could use resilient distributed datasets objects of Spark [138].

The raw data currently can be stored in any local drive in both txt and JSON format. The data represents the condition of the HSU of the printer machine. Since the PdM application is implemented for the first time for this machine at NTS, there is not enough data (which should be collected for several months). In this work, the whole end-to-end framework is implemented with a small dataset. As shown in Figure 5.13, the data life cycle starts with sending data from HSU to a local server and then gathered by the NTS IoT application for sending to the Jupyter notebook [139]. After analyzing the data in the Jupyter notebook, the results are sent to the MySQL database. For visualization of the result, Grafana is used. If we use the cloud connection, the data can be sent to the cloud via the MQTT broker.

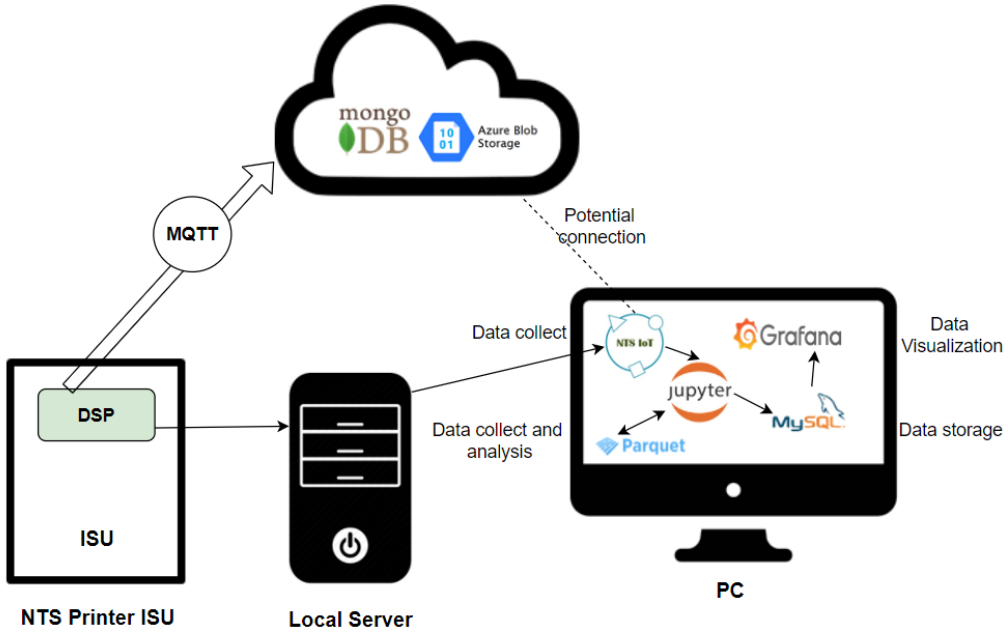

Figure 5.13: Data life cycle in the proposed PdM framework

Splitting Dataset

After filtering raw data and capturing the relevant parameters (to pump condition), we need to divide the dataset into train and test datasets. As this dataset is of time-series format, the data sequence in this splitting process should be considered. Therefore, the splitting is done based on the order of timestamps. It is important to dedicate a suitable portion of the original dataset for training and testing/validation. The dataset should be split considering the existence of both healthy and unhealthy data in the training and test dataset. In our case study, before collecting data, failure scenarios need to be designed for the pump in the printer machine.

Failure Scenarios' Mechanism for Collecting Data

Since for the supervised machine learning application, healthy data (that collected in a normal operational period of the equipment) and erroneous data (that collected in the failure period of the equipment) are

needed, different failure scenarios were designed with the help of NTS experts. There are two main scenarios: a) leaking and b) obstruction, which impact pump operational behavior. For the failure scenario, after starting the machine for several hours, just healthy data were collected, and then in specific moments, we started to induce failure in the pump operation. Thus, a suitable dataset has been made by gathering both healthy and faulty data and recording the moments of failures. It should be mentioned that in this case study, we utilized the data harvested from the obstruction failure scenario. In Figure 5.14, the dataset is depicted, and some healthy and unhealthy samples are pointed by green and red boxes, respectively. Now,

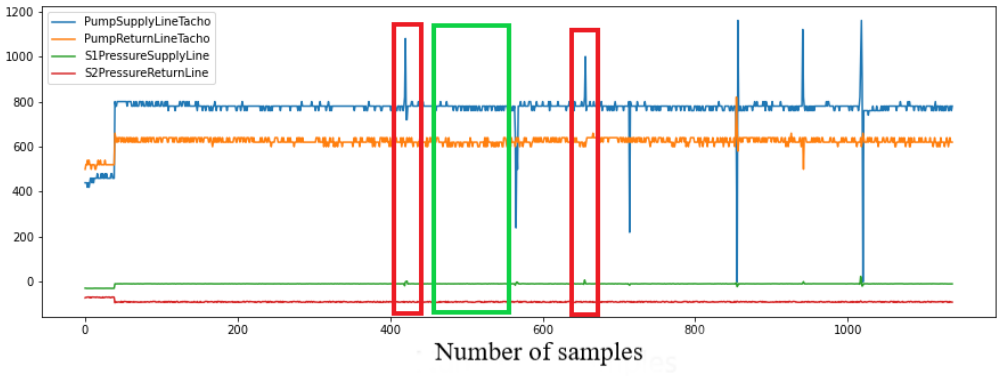

Figure 5.14: Dataset with healthy and unhealthy condition samples. Red boxes represent failure while green box highlights healthy condition

the data is split into train and test datasets. We tried to have enough faulty and healthy data in both train and test datasets based on the failure moments.

After importing the datasets as Spark dataframes, the NULL observations should be removed (i.e., data observations with no values) to have a more clean dataset.

5.2.5 Data Visualization

In this module, we are looking for any signs or anomalies that can help in selecting the proper parameters related to the failure. Several visualization techniques exist to detect abnormalities in the data. As shown in Figure 5.15, train and test datasets present some features that changing over time. Apache Spark does not have a data visualization module. Therefore, to visualize the data, we require changing the Spark dataframes to Pandas dataframe.

Data Type Casting

Machine Learning algorithms usually rely on mathematical operations which require their inputs to be of a numeric type, for example, binary or integer values. Therefore, for both train and test datasets, typecasting of the data to the required types has been done (i.e., timestamp, integer, double types.)

Labeling Methods

The other important step in this time-series dataset is the labeling. We add RUL labels to both train and test datasets in this step. Remaining Useful Life estimates the number of time units (e.g., hours/days/weeks/-months), during which the machine can run in good condition before it fails completely.

The RUL can be estimated by observation of actual failure event (i.e., pump failure) and counting the number of cycles of each pump in descending order to estimate the number of cycles/days each pump can run before a failure occur. For this purpose, we need the average operational life of the pump working in the printing machine. Since the machine is new, we do not have any estimation of the pump's RUL. For

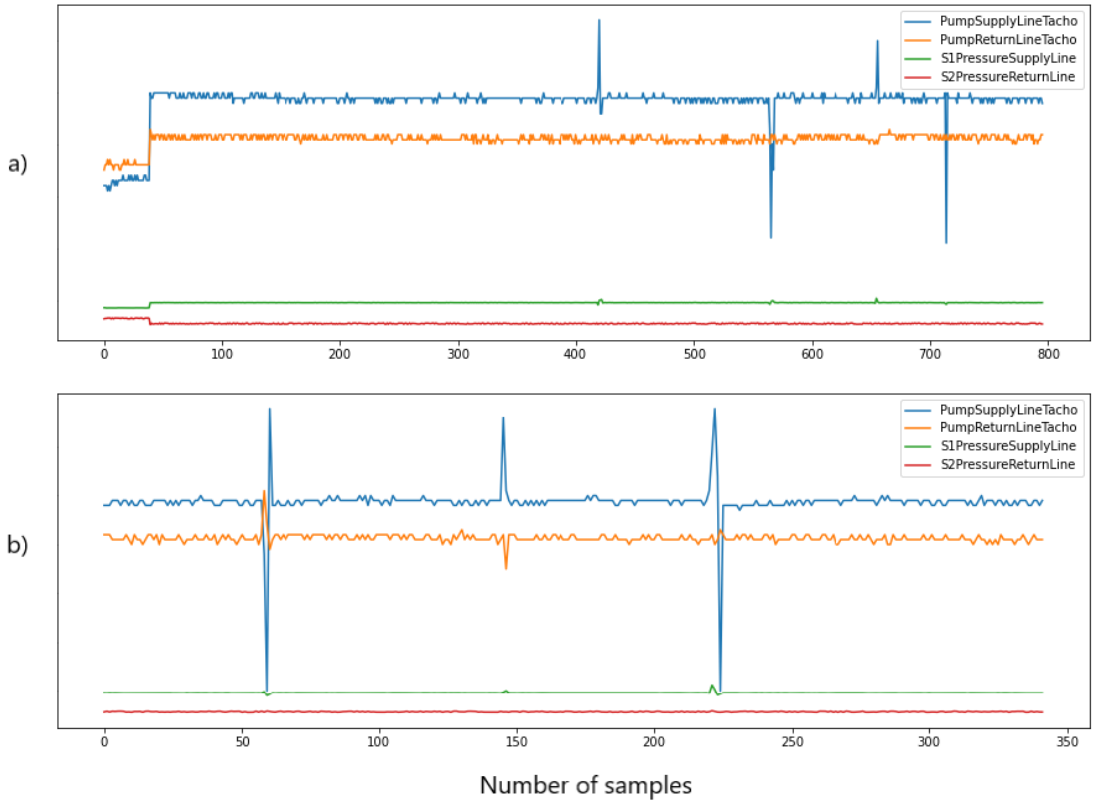

Figure 5.15: Splitting dataset to train (a) and test (b) datasets

now, we just implemented the RUL labeling with a dataset that we have. If there are more reliable data for the pump's RUL in the future, these code blocks can be utilized.

The RUL labeling method is used in the regression analysis. After calculating RUL values, a new column called RUL is added to the Spark dataframe.

Another labeling method is windows labeling. We intentionally made failures for the machine at specific times. Then a table called "Event Table" is made, which consists of start time and end time of failure events. With the help of this table, the train and test datasets are labeled. Hence, by checking the event table and "Timestamp" column of the dataset, one can label the data as unhealthy if they are in the range of failure events.

5.3 Data Preprocessing Implementation

During the data acquisition block implementation, relevant data was harvested with JSON format and stored locally. After applying a filtering process, relevant data were extracted. Filtered data is stored locally in CSV format. In the Data Preprocessing block, several steps are considered for preparing the dataset to deliver it to the ML algorithms. These steps are explained in the following sections.

5.3.1 Data Cleaning

The visualization section shows that some data features have no or very low variance over time. As this class of features is useless in building predictive data models, it is logical to remove those features to reduce data dimensionality. First, we need to define the maximum unaccepted variance threshold. Then it is enough to compute each feature variance and delete the feature if its variance is less than or equal to the

given threshold. In this work, due to the small dataset, if we apply this cleaning method, we lose valuable information. Therefore, we skipped from this step.

Removing Outliers

The key point to getting machine learning to work efficiently is to ensure that the data is utilized for training, which is as clean as possible and has any biases removed from it. Otherwise, machine learning faces outliers that impact the results in the wrong way. There are several statistical methods to identify and remove outliers [140] such as:

- Standard Deviation
- Median Absolute Deviation
- Interquartile Deviation
- Z-Score

Due to the lack of data and working with industrial applications (any outlier can be a sign of machine misbehavior), any data elimination can cause deterioration in our case study's result. Hence, we keep all the data.

Removing Noise

In reality, there is always noise. Whenever a value is measured, some error will be presented by its capture, transmission, or other reasons. The measured values can be introduced as:

$$Measured\_value = True\_value + noise$$

It is required to extract the true value but also the noise. Several algorithms like signal processing and filtering algorithms help remove noise from a signal and getting as close to the truth as possible. To achieve this goal, a moving average algorithm to the data is applied. The Spark dataframe is partitioned utilizing a window module, and the corresponded rolling average feature of each numerical data feature in that dataframe is calculated. Therefore, the irregular data is removed. In Figures 5.16 and 5.17, the features before and after applying noise reduction are represented.

As expected, and it can be seen in Figures 5.16 and 5.17, the results are smoother graphs that can help in optimizing the results of machine learning algorithms.

Filling Missing Value

The real data mostly has a lot of missing values. Some reasons can cause missing values, for example, data corruption or failure to record data. Since many machine learning algorithms can not deal with missing values, the treatment of missing data is critical during the preprocessing phase of the dataset. There are several ways to handle missing values in the dataset, such as:

- Deleting rows with missing values
- Impute missing values for continuous variable
- Impute missing values for categorical variable
- Other Imputation Methods
- Using algorithms that support missing values
- Prediction of missing values

Missing values can be treated by eliminating the rows or columns that contain null values. If columns have more than half of the rows as null, then the entire column can be dropped. The rows which have one or more column values as null can also be dropped. For creating a robust and reliable model, we need to train a model without any null value. Although, employing this technique in some cases, for example, the portion of missing values, is unreasonable compared to the whole dataset. It results in the poor performance of machine learning due to a loss of information.

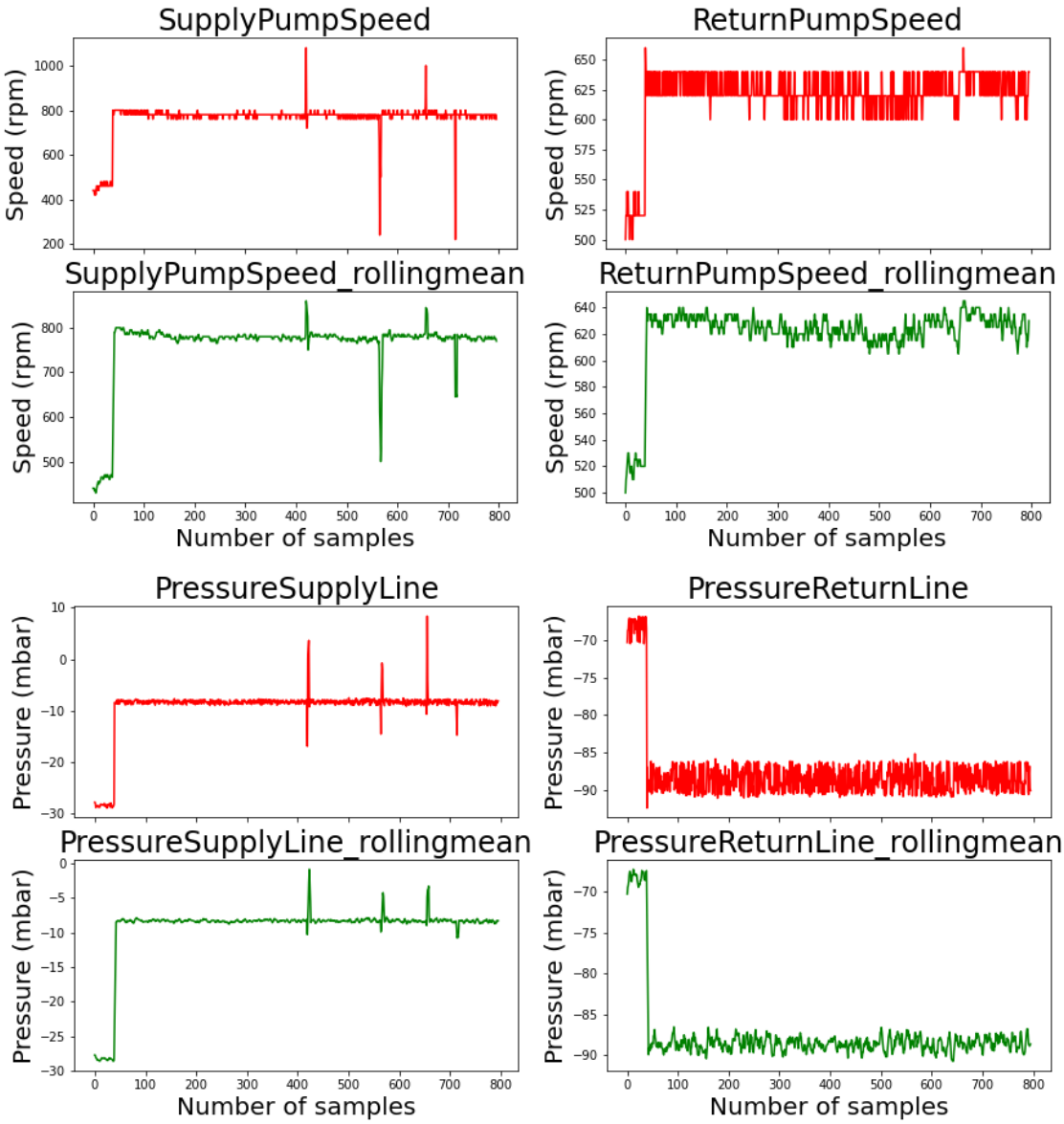

Figure 5.16: Noise reduction for train dataset. Red graphs shows signal before noise reduction and greens are showing it after noise reduction

Another approach is to impute missing values with mean/median. In this technique, columns in the dataset with continuous numeric values can be substituted with the median, mean, or mode of remaining values in the column. Employing this method helps to avoid the loss of data in comparison with the earlier method.

The next method is the Imputation method for categorical columns. When the type of missing values is categorial, for instance, string or numerical, then the missing values can be substituted with the most frequent category. This method also helps the problem of data loss that results from the deletion of rows or columns.

Other imputation methods [141] may be more suitable to assign missing values, depending on the data type. In the case of the time-series dataset, there is a popular method called Interpolation. The Interpolation of the variable can be done before and after a timestamp for a missing value.

Another popular method is the prediction of missing values. In the earlier approaches, for handling

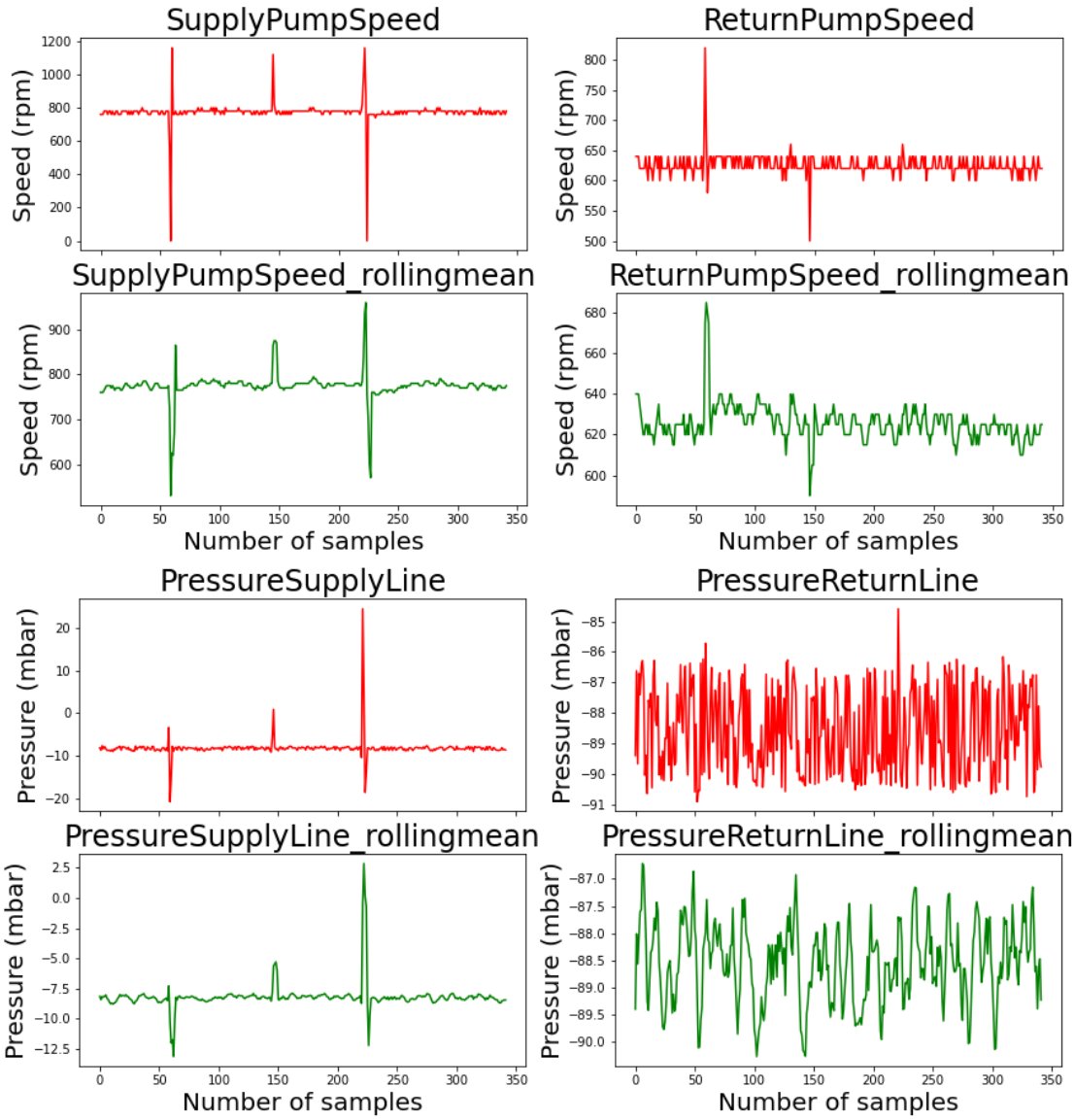

Figure 5.17: Noise reduction for test dataset. Red graphs shows signal before noise reduction and greens are showing it after noise reduction

missing values, we do not benefit from the association of the variable containing the missing value and other variables. Utilizing the features which do not contain nulls can be used to predict missing values. The regression or classification algorithms can be employed for the forecasting of missing values [142].

Every dataset contains missing values that need to be treated effectively to build a robust model. Several popular methods to handle missing values have been presented. There are no strict rules to employ a particular way, depending on how and what the data is about various methods on different features. Therefore, we can conclude that knowing the dataset is important, giving insight into how to preprocess the data and treat missing values.

To tackle missing values, removing null values by dropping the rows with null values was implemented. After applying this method to the corresponding dataset, the number of rows before and after removing null values were equal. Thus, in this dataset, there were no null values to be removed.

5.3.2 Feature Engineering

By applying a moving average algorithm to the Removing Noise Section 5.3.1, mean values are generated for each signal and stored in separate columns as new extracted features (time-domain features). In many scenarios, raw data values do not have meaningful information for the application of machine learning. Hence, time- or frequency-domain data are needed to be extracted. Furthermore, all these extracted values and raw values need to be scaled to avoid any data bias.

Normalizing Data

One of the steps in data preprocessing is feature normalization. The goal of normalization is to re-scale each feature to be in the [0,1] domain. Hence, feature normalization unifies the scale of all data features, which helps ML algorithms to generate accurate models. For example, applying the normalization function can generate normalized features out of a set of numerical data features. Since there are raw data and extracted features in the dataset, normalization was performed for both raw data and generated features. The following Figures 5.18 and 5.19 show train and test data features after normalizing them in [0,1] domain. The rolling average features were normalized.

Standardization

For feature scaling, we can use normalization or standardization techniques. In this procedure, features are re-scaled such that each feature value is centered around the mean with a unit standard deviation. In predictive modeling, normalized features, standardized features, or a collection of both sets can be used to achieve better performance. Therefore, in this step, a list of numerical features is standardized in the Spark dataframe. The following Figures 5.20 and 5.21 show standardized data features generated for the train dataset. The rolling average features were standardized (i.e., low-noise data features).

5.3.3 Dimension Reduction

In previous steps, many new features out of existing ones were generated. However, it is possible to generate more features in terms of the time- or frequency-domain. In this case study, it was decided to continue with the number of features that have been extracted so far and not to generate more.

Until now, the data dimensionality has significantly expanded. In order to go further into predictive data modeling, the data dimensionality needs to be reduced. Reducing data dimensions enables us to visualize the data more efficiently, test different parametric settings of machine learning algorithms to optimize the predictive solutions, and make much use of memory and storage utilities. Now, the question is what features can be removed and lead to a dimension reduction without losing useful information. Considering useful information in machine learning, we need to keep as much as diversity of the data. There are several methods for dimension reduction, such as Heatmaps, t-SNE plots, multi-dimensional scaling. One of the most popular ones is PCA. Each Principle Component (PC) is a specific combination of input variables. Since linear models such as linear regression need as many independent features, PCA can provide this quality. All the PCs are independent of each other. Now, the PCs can be computed, and based on the diversity threshold that is needed, it is possible to keep the PCs and discard the rest of them. This threshold can be calculated based on feature variances. After calculating the variance for each PC, they are represented in a cumulative variance graph.

As PCA is a widespread technique among reviewed literature in Chapter 3 related to predictive maintenance application, we also decided to utilize this technique as a dimension reduction approach. After applying the PCA algorithm to the Spark dataframe, the required number of PC features was obtained. The following Figures 5.22 and 5.23 describe the accumulated data variance obtained by the first four PCs in the train and test datasets. As presented in Figures 5.22 and 5.23, utilizing the PCA algorithm helped to reduce nearly all data variance in the first 4 PC features. It means that efficiently reducing data dimensionality from 17 features to 4 features without losing variance. Visualization of PC features presented in Figures 5.24 and 5.25. As Figures 5.24 and 5.25 show that the generated PCs have different scales, re-scaling those features using normalization and standardization procedures should be done.

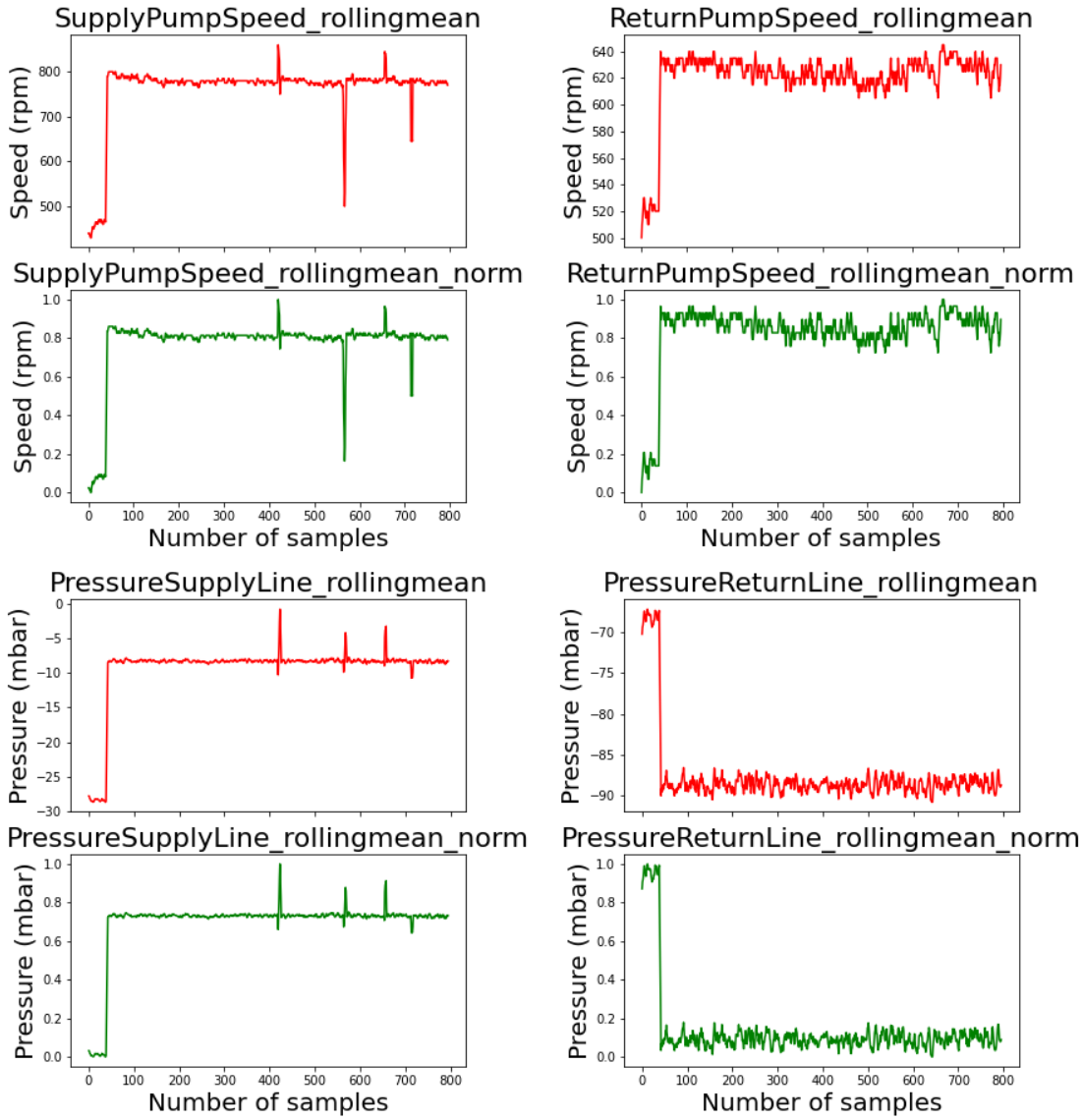

Figure 5.18: Normalization on rolling average features of train dataset. Red graphs show the signals before normalization and greens show them after normalization

Feature Selection

Many features were generated, even in the Dimension Reduction section, PCs were produced. The most effective step for training a machine learning algorithm is delivering relevant features to it. For example, the aim is to find the features that have more impact on the machine's failure. Hence, we should conduct a feature importance algorithm in this step to find the most relevant features corresponding to the target output. We can use a combination of PCA and other techniques to get better results.

When there is a dataset with many features, we can benefit from data importance techniques. It is tempting to think that a large number of features will help a model make better predictions but, that is incorrect. Trying to train a model on a set of features with no or very little correlation gives inaccurate results. When dealing with multi-dimensional datasets, it is important to filter out non-correlated features. Instead, it is better to use fewer highly correlated features to train a model.

Datasets with more features or higher dimensions are a recent problem. These days, data collection

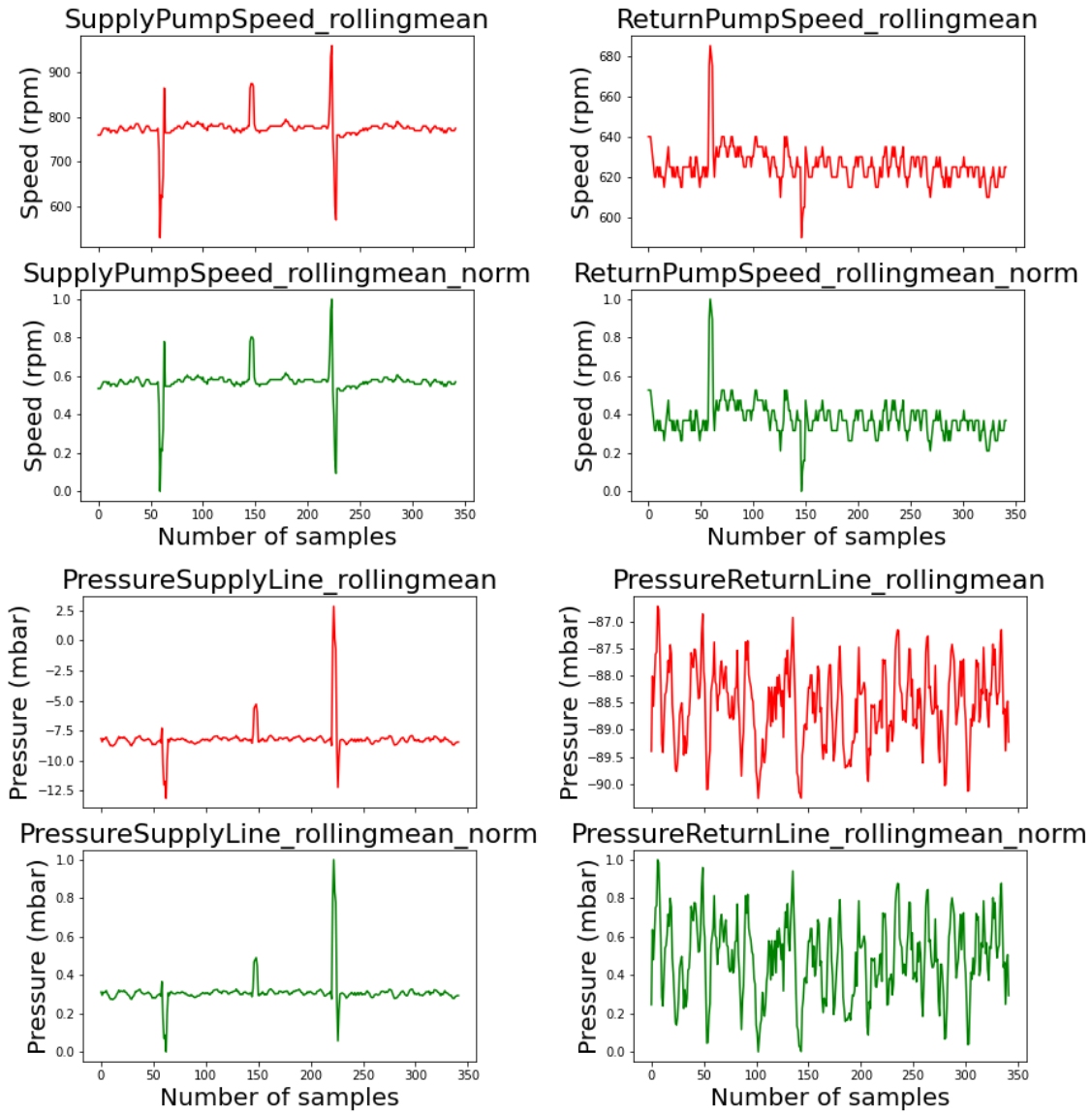

Figure 5.19: Normalization on rolling average features of test dataset. Red graphs show the signals before normalization and greens show them after normalization

and storage have never been easier. Usually, many datasets have features with similar information. It acts as noise in the system and increases complexity. Some features also have very little variance. If the output has a lot of variances, then a feature with low variance will not improve a model.

By employing feature selection methods for machine learning, we can benefit from: a) Reducing the chance of overfitting. b) Enhancing the algorithm running speed by reducing the I/O, CPU, and RAM load. It needs the production system to create and utilize the model by reducing the number of actions needed to read and preprocess data and perform data analytics techniques. c) Increasing the model's interpretability by revealing the most informative items that drive the model's results.

To realize the importance of each feature in a dataset, a technique known as PCC [143] was used in this work. This technique compares which features correlate with the output (the labeled column (RUL and ClassLable)) in the considered dataset). The PCA and PCC are generally used for linear variable selection. PCC has been widely utilized for variable selection due to its simplicity and as it helps to identify the

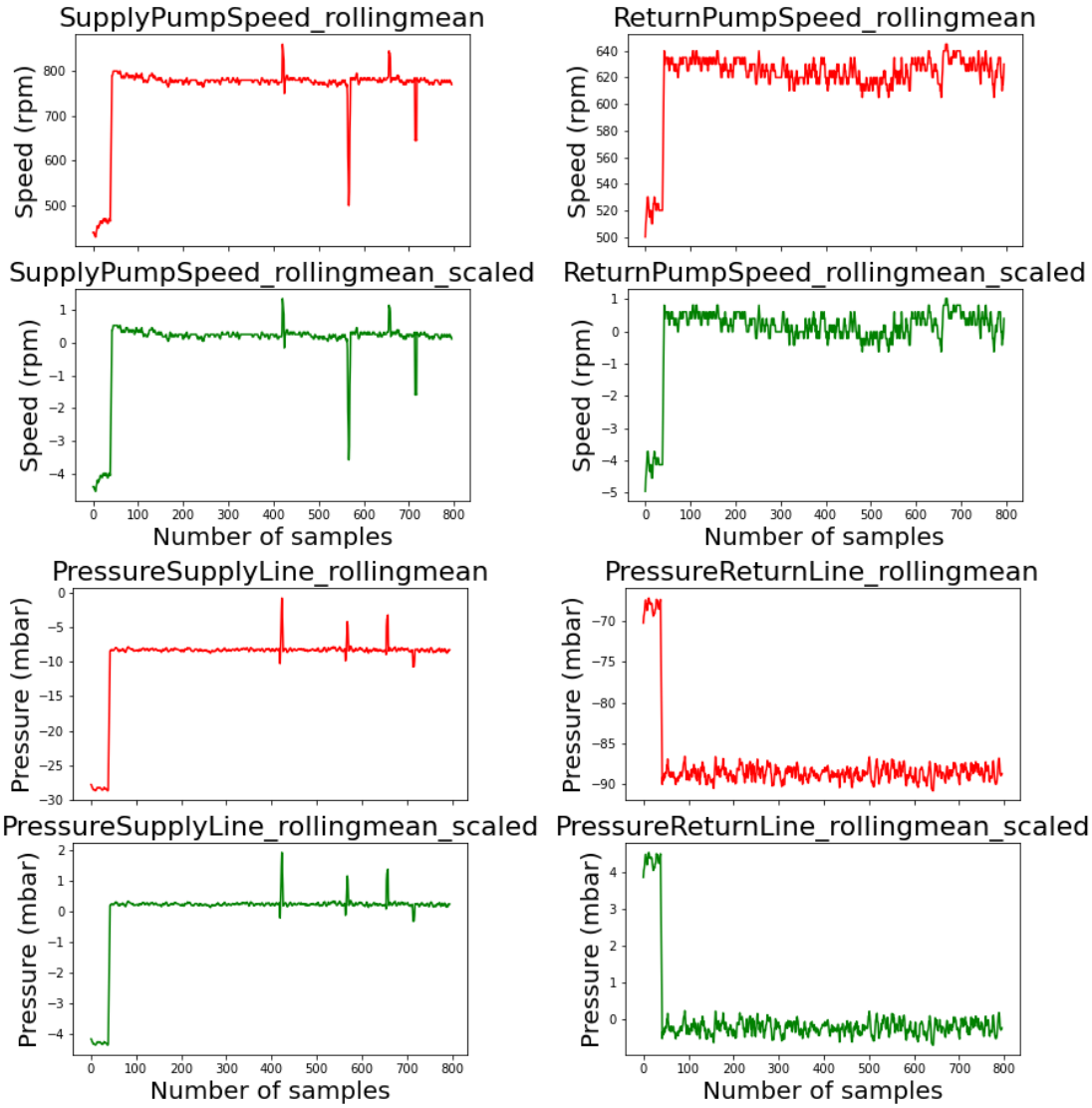

Figure 5.20: Standardization on rolling average features of train dataset. Red graphs show the signals before standardization and greens show them after standardization

degree of correlation between independent and target variables. However, there is a big difference between PCA and PCC that PCA has been used for recognizing variables that have high variances affecting the target variable [143]. Therefore, it is assumed that combining these two techniques will help improve the machine learning results.

Pearson's Correlation Coefficient helps to find out the relationship between two quantities. It estimates the strength of correlation between two variables. The Pearson's Correlation Coefficient value can be between -1 to +1. 1 indicates that they are highly correlated, and 0 indicates no correlation. -1 indicates that there is a negative association. It has an inverse proportion.

It should be considered that if there is a large dataset and the result shows a small coefficient, i.e., 0.3, then it is not necessarily a bad result. The dataset might have a large statistically notable association. Likewise, note that correlation may not mean causation. Just because two variables are associated, it does not mean that one directly caused the other.

To obtain the proper subset of data features, the PCC technique is employed. As PCC measures the correlation between target variables and independent variables, first, we identify target variables. There

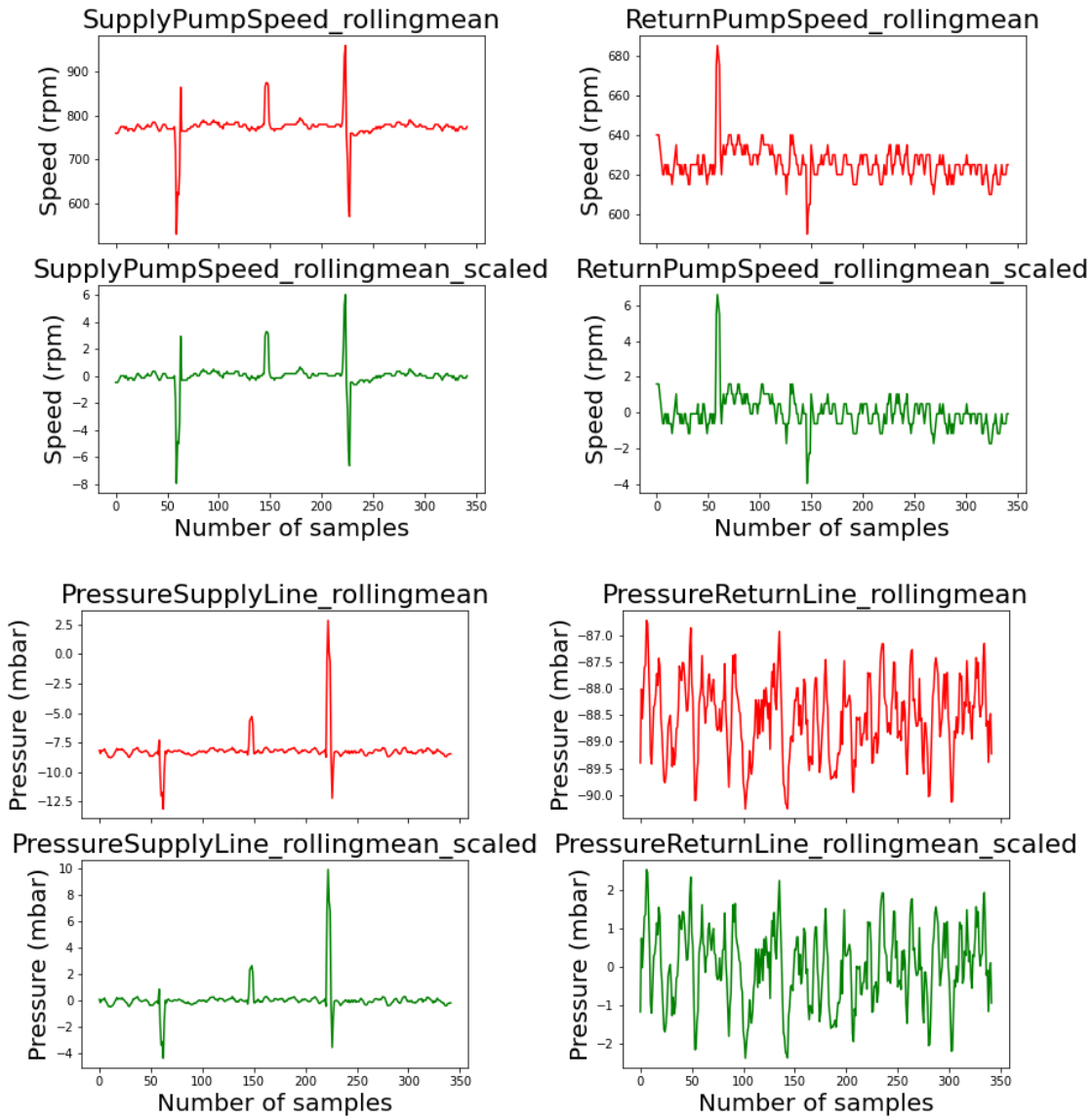

Figure 5.21: Standardization on rolling average features of test dataset. Red graphs show the signals before standardization and greens show them after standardization

are two different target variables, one of them is RUL (the pump's Remaining Useful Life), and the other one is ClassLabel (healthy and unhealthy labeling values). RUL is employed for regression analysis and Classlabel for binary classifiers.

Two vectors were built after obtaining the best-correlated features to the output variable (RUL/ClassLabel). One of these vectors will be used as the train data vector in all regression algorithms (that its feature correlated with RUL) and another vector for classifiers (that its feature correlated to ClassLabel).

By completing this step, the second building block that is data preprocessing, is finished. Now the data is ready for feeding to the next BB that is the Predictive Analytics block.

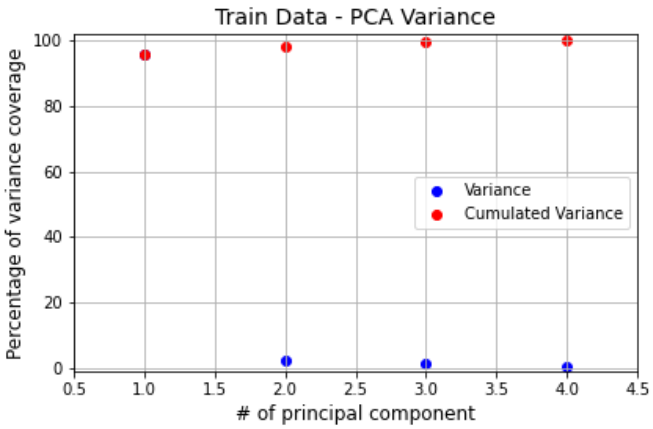

Figure 5.22: Data variance obtained by the first four PCs in train dataset

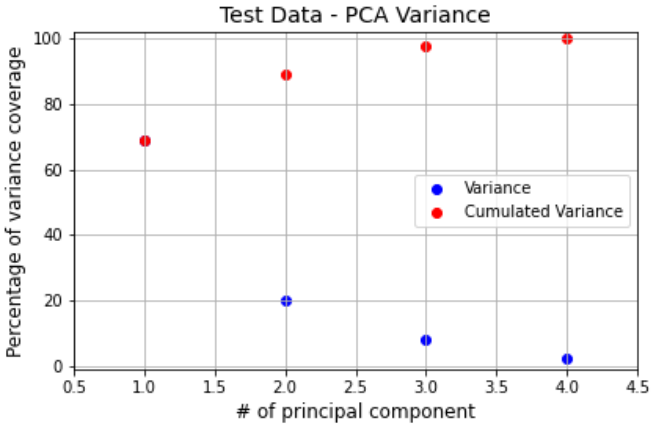

Figure 5.23: Data variance obtained by the first 4 PCs in test dataset

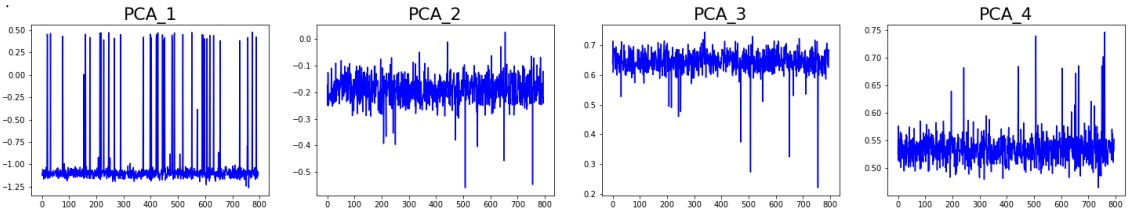

Figure 5.24: PCs visualization of train dataset

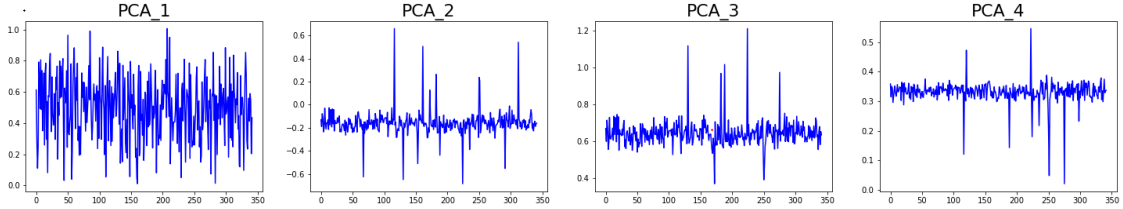

Figure 5.25: PCs visualization of test dataset

5.4 Predictive Analytics Implementation

Several machine learning algorithms are employed in this block to predict the pump's performance (if the pump is working properly or if it needs maintenance). This block consists of three main machine learning models: a) Supervised Learning, b) Unsupervised Learning c) Semi-supervised Learning. Moreover, several techniques will be discussed for the validation of these algorithms. In this work, it was considered to implement supervised and semi-supervised learning.

After obtaining the positive and negative correlated features to the target variable (RUL/ClassLabel) from the previous building block, two training vectors to feed MLs were built. One of these vectors will be used as the training vector in all regressor algorithms (related to RUL) and another one used for classifier (related to ClassLabel) algorithms.

5.4.1 Supervised Learning

Supervised learning algorithms are trained by utilizing labeled data. In addition, independent feature data is given to the model along with the target value. In the following steps, we will discuss building models and validate them.

Model Building

In the model building module of supervised learning, there are two categories, which are **regressor** and **classifier**. Based on the popularity of the machine learning algorithm, we decided to implement several regressor algorithms such as linear regression, general linear regression, and decision tree. For classifier, these algorithms are chosen: logistic regression, decision tree, and random forest.

Regressors

To start with regressors, some parameters need to be initialized for each of the algorithms. In the LR algorithm, there are some parameters that one can set, such as the maximum iterations of LR, Lambda, and elastic net [144]. The next step is to fit the LR model to the train data vector. The regression model is built by feeding the train data vector to the LR instance. It should be mentioned that the training vector consists of selected features (by PCC) and the target variable (which is the RUL values). The third step is to build a test data vector compatible with the training vector. Therefore, the same train vector as a test vector was used. After executing the mentioned steps and delivering the test vector to the trained model, the prediction result can be achieved.

Since we have built the first regression model and applied it to the test dataset, now the model can be evaluated. Apache Spark has helpful functions that calculate evaluation metrics of regression models. To employ them, it is needed to initialize an evaluation object with the predicted label and the labels we defined for the considered test dataset. Now, by initializing this evaluation instance, some evaluation metrics can be calculated. These metrics are employed to evaluate the regressors' model performance: R Squared (R^2), MSE, RMSE, and MAE. In addition, it is aimed to compare and analyze these results in the next building block, which is the result evaluation block. Another regression model known as Generalized Linear Regression (GLR) was built by following the same steps as done for Linear Regression.

A significant aspect of training a machine learning model is to evaluate whether the model is overfitting or underfitting the data. Overfitting commonly occurs when a model tries to fit all the data points, capturing noises in the process, leading to inaccurate model development. Underfitting is a scenario where a data model cannot capture the relationship between the input and output variables accurately, generating a high error rate on both the training set and unseen data. There is a parameter called *Lambda* that can be used for regulating the model accuracy. By choosing the Lambda (also called the regularization rate) value, the purpose is to discover the correct balance between training-data fit and simplicity: If the chosen lambda value is too high, our model will be simple, but we run the risk of under-fitting our data. As a result, the model will not learn enough about the training data to make valuable predictions. For implementing Generalized Linear Regression (GLR), different Lambda Values were utilized to investigate the impact of Lambda for better accuracy and a minor error in the next building block.

The last regression model in this section is the algorithm. Here, two parameters of the decision tree, namely, the max number of bins (that enables more feature partitioning) and max depth (which is the length of the longest path from the root to a leaf), are set. There is no theoretical calculation of the best depth and bins of a decision tree. Therefore, several depths and bins were chosen to apply a trial and error procedure. Consequently, based on the evaluation metrics, it is possible to reach the optimal ones. For evaluating the Decision Tree model, R-Squared, MSE, RMSE, and MAE metrics are calculated; this is presented in the next block.

Classifiers

In this part, the goal is to classify the pump's performance into two classes: normal class (which is assigned with 0) and faulty class (which is assigned with 1). To achieve this goal, three binary classification models were built, namely: Logistic Regression, Decision Tree, and Random Forest.

Cross Validation

In the context of classification algorithms, Cross-Validation (CV) is a technique to avoid model overfitting. In order to apply cross-validation in Apache Spark, firstly, it is needed to build a parameter grid object. Parameter grids enable a given classifier to try out different parameter settings and optimize the accuracy for each combination—a parameter grid loops over a list of regression and elastic search parameters of logistic regression. The purpose of grid-search is to find the optimal hyperparameters of a model, resulting in the most accurate predictions. An illustration of cross-validation technique presented in Figure 5.26.

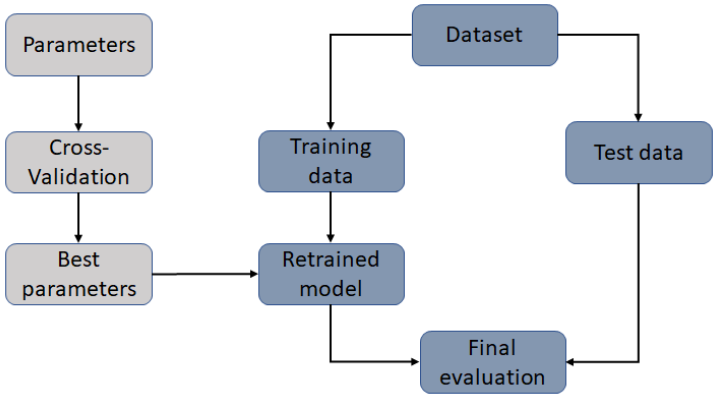

Figure 5.26: Illustration of cross validation

One of the popular techniques in the ML models consists of randomly picking samples out of the available data and split them into train and test datasets. In the case of time series also, cross-validation can be very helpful. However, randomly selecting time series samples and assigning them to the test set or the training dataset does not seem logical. Ignoring the sequence of data causes facing the problem of forecasting values in the past by using values from the future. In simple words, the aim is to avoid future-looking while training train the model. There is a temporal dependency between observations which must be preserved during testing.

Cross-validating of a time-series model can be done by employing a rolling base cross-validation technique. First, a small subset of data is selected for training; the second step is predicting this to the next data points, finally measuring the accuracy for the predicted data points. The same predicted data are then added as part of the next training set, and subsequent data points are predicted.

Here, Figure 5.27 is an image of this technique: In k-fold cross-validation, first, the dataset is split into several folds, then the model is trained on all folds except one, then testing the model remaining folds. These steps need to be repeated until the model is tested on each of the folds, and the final metrics will be the average of scores obtained in every fold. This prevents overfitting and evaluates model performance in

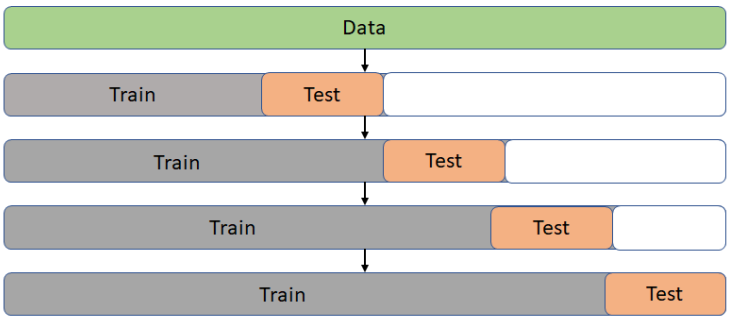

Figure 5.27: K-fold cross validation for time series data

a more robust way than a simple train-test split.

Therefore, for evaluating the performance of the models, first, the CrossValidator function from PySpark is used for k=3 folds.

The number of folds is generally defined by the number of instances included in the dataset. For example, in 10-fold cross-validation with only ten instances, there would only be one instance in the testing set. This instance does not correctly represent the variation of the underlying distribution. For selecting k, we must ensure that the training set and testing set are drawn from the same distribution. Finding the right value for k is not an exact science because it is hard to estimate how well the fold represents the overall dataset. However, if the dataset size increases dramatically, like if we have over 200,000 instances, it can be seen that 20-fold cross-validation would lead to folds of 20,000 instances; this should be sufficient to test the model reliably.

In short, we can say the number of folds depends on the data size. In addition, because we are using time series data, cross-validation cannot benefit from the shuffling of the data, such that the folds do not contain inherent bias. Hence, we have to go for a lower number of folds to avoid bias.

Additionally, we should also consider the computational costs for the different values. High K means more folds, thus higher computational time and vice versa. Subsequently, one needs to find an optimal spot between those by doing a hyper tuning analysis. Furthermore, if the dataset size is small, using k-fold cross-validation would not make sense.

Based on our observation and knowing that the corresponding dataset is small, we decided to use about 30% of the dataset for testing, and consequently, we used 3-fold cross-validation. After choosing k, which is equal to 3, we created a CrossValidator object and passed the model, parameter grid, and evaluation instances to it. For all three classifiers, we run 3-fold cross-validation to find the optimum result. The evaluation metrics, in this case, are accuracy, macro/micro recall, macro/micro precision, and F1-score, which will be discussed in the next building block.

For implementing the two other classifiers, which are Decision Tree and Random Forest, the same steps of Linear Regression are followed. These algorithms' performance was analyzed along with all the classifiers and regressors in the next building block.

5.4.2 Unsupervised Learning

This section has two modules: model building and model validation modules. For unsupervised learning in industrial cases when there is no labeled data, Auto Encoder (AE) as an unsupervised learning algorithm is a good choice. Moreover, another technique called Peak detection seems applicable in our case study.

Peak Detection

In this method, a signal is chosen with the most relevant parameters to specific equipment failure. In our case study, after discussion with NTS, we realized the pump's speed and pressure could show more meaningful information related to the pump's failure. Therefore, the data received from the pump's tachometer is collected and fed to the peak detection algorithm. Peaks and valleys are detected. The second step is

to count the number of peaks or valleys in a time window. Depending on the sampling frequency, the window size should be chosen. Currently, the sampling frequency is low (0.033 Hz), so the window size should be bigger than 1 minute. After applying this window size and counting the number of peaks during the window size, the result shows some changes in the number of peaks before failure happens. If, for example, we have an assumption that when the system is in its normal operation condition, the number of peaks should be equal to N in the window size of M minutes. Then by implementing an algorithm based on the number of peaks, we can catch anomalies in the machine.

In this work, Peak detection is implemented by connecting MySQL [126] database and Grafana [123] software. Grafana provides features to detect changes in time series data. When writing queries in Grafana, we caught the moments that the number of peaks exceeded 200 in 3 minutes (that was one of the failure symptoms). Figure 5.28 shows a database with more than 500,000 samples used for the peak detection method. It should be mentioned that because this dataset is collected (customer side, which from 5 ISUs for five months)at some random time, we can not use it for other machine learning algorithms such as classifiers employed in this work. This dataset just has been used for peak detection.

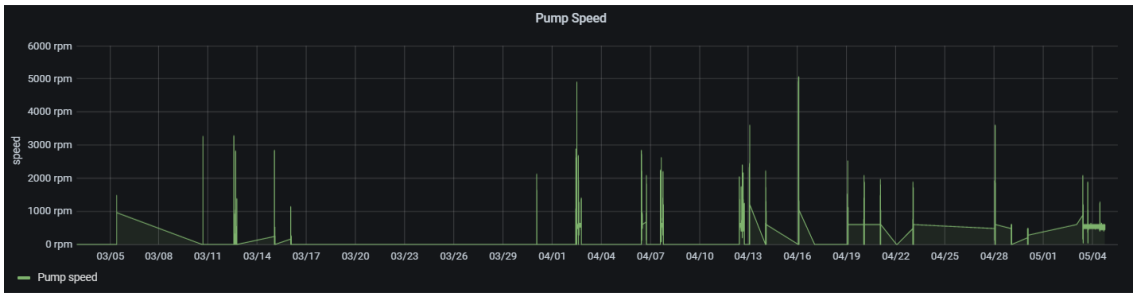

Figure 5.28: A big dataset collected for several days

In this step, we used knowledge-based methods to validate the model as we discussed with the NTS expert about the pump's characteristics and constraints. By tracking the pump's speed closely from normal operational time to the moment of failure, a fluctuation is observed in the speed signal. On approaching the failure moments, the period of the fluctuation increases, and in the end, it leads to the failure of the pump. Based on its datasheet, this type of pump is designed for continuous operation and impulse operation has a harmful effect on the pump's useful life, we can count the number of peaks in a specific period to catch earlier pump failure symptoms.

5.4.3 Semi-Supervised Learning

For the semi-supervised method, there are two modules: Model Building and Model Validation. First, we try to implement a semi-supervised model and then evaluate it with some evaluation metrics. A semi-supervised method is employed that is known as the One-Class Classifier (OCC). It is a domain of machine learning that presents techniques for anomaly and outlier detection. The OCC can be helpful for imbalanced classification datasets where there are none or very few examples of the minority class. It is also effective for datasets where there is no coherent structure to separate the classes that could be learned by, for example, a supervised algorithm. OCC is fit on a training dataset that only has examples from the normal class. Once the model is trained, it is used to classify new examples as either normal or not-normal, i.e., outliers or anomalies.

Since Apache Spark MLlib does not support this type of classifier, we employed another library. The scikit-learn library [145] supports common one-class classification algorithms aimed for outlier or anomaly detection, such as Isolation Forest, One-Class SVM, Local Outlier Factor, and Elliptic Envelope.

In this work, One-Class SVM has been utilized. The important characteristic of one-class SVM that makes it attractive for our application is that it categorizes new data as similar or different from the training set. Based on this assumption, if feeding it with a totally healthy class dataset (as the training dataset), it will categorize the test dataset according to whether they are similar to train data. It assigns 1 for similar

data and -1 for the dissimilar of the dataset. If it labeled the data as 1, it means the machine condition is healthy, and if it labeled it as -1, it means the system is in unhealthy condition.

As we do not have enough labeled data from the failure events, we need a labeled dataset in a way that is useful in one-class SVM (the labeled values are used to model evaluation, not for training the model). First, we considered an operational period of the machine that we are sure there are no failure events there (or minority of classes are unhealthy and the majority are healthy). Then, we labeled that whole period as healthy (i.e., labeled as 0). Therefore, if we feed the machine learning algorithm with this almost healthy dataset, it will consider it a healthy behavior. If any other event happens for the machine except this defined healthy behavior, it will consider it an anomaly. We intentionally made a failure for the machine to test this method at specific times. Then we created a table called "Event Table," which consists of start time and end time of failure events. With the help of this table, we labeled the test dataset. Hence, by checking the event table and "Timestamp" column of the dataset, it can label the data as unhealthy if they are in the period of the event. We used these label values for validating the performance of the one-class classifier.

It should be mentioned that the training dataset, as illustrated earlier in Figure 5.15, consists of failure events too, but SVM considered those values as outliers, and it will discard them as a minority class. Consequently, One-Class SVM recognizes the majority class and determines it as a positive class (in our case, normal behavior).

5.5 Result Evaluation Block Implementation

So far, the machine learning models were implemented. Now in this block, we can analyze and compare the results of different models. The result evaluation block consists of several modules. First, we evaluate the results with dedicated evaluation metrics, and then we choose the models that should be deployed for the production line. Finally, we try to improve the models' results.

In this step, we want to examine the results obtained based on the proposed PdM framework. For this aim, we designed several different scenarios to investigate the impact of each block of the framework. Six different scenarios were defined for supervised and semi-supervised learning. Table 5.1 shows these designed scenarios. For example, in scenario 1, we see the result when we consider that all of the blocks and techniques are implemented, such as scaling data, extracting time-domain feature, dimension reduction with PCA, and feature selection with Pearson Correlation Coefficient. In another scenario, for example, scenario 3, the PCA is discarded to see the influence of PCA in the final prediction results.

Table 5.1: Comparison scenarios based on PdM blocks

Scenario	Train-Test(%)	Scaled raw data	Time Feature	Number of features	PCA	PCC
1	70-30	YES	YES	10	YES	YES
2	70-30	YES	NO	10	YES	YES
3	70-30	NO	YES	10	NO	YES
4	70-30	YES	NO	4	NO	YES
5	70-30	YES	NO	4	NO	NO
6	70-30	YES	NO	3	YES	NO

In the following sections, first, the regressor results are presented and then the classifiers. After that, we will continue with peak detection results, and finally, one-class SVM result will be discussed in this section. Note that based on the defined scenarios, we will see the result in the following sections.

5.5.1 Supervised and Semi-Supervised Model Evaluation Metrics

For supervised, semi-supervised, and unsupervised methods, there are different evaluation metrics. Since in the supervised methods, regressors are utilized for calculating the remaining useful life of the equipment (for example, the operational time of the machine will finish after 400 cycles), evaluation metrics are needed to estimate how close predictions are to the outcomes. The mean squared error (MSE) estimates the average of the squares of the errors, which indicates the difference between the estimator and the estimated. The

mean absolute error (MAE) is a quantity adopted to estimate how close predictions are to the target values. By applying metrics such as R^2, MSE, RMSE, and MAE, we are trying to find the closest value to the actual value and find a best-trained model.

Other metrics such as F1-Score, recall, precision, accuracy are employed to evaluate models trained by classifiers. For example, these metrics show how many predicted labels are correct and how many are faulty. In this concept, we used a confusion matrix that is commonly employed for summarizing the performance of a classification algorithm.

The range of the metric that is acceptable should be decided beforehand based on our application. In general, the RMSE values between 0.2 and 0.5 show that the model can relatively predict the data accurately. In addition, an Adjusted R-squared of more than 0.75 is an outstanding value for showing accuracy. In some cases, an Adjusted R-squared of 0.4 or more is acceptable as well. MSE measures the sum of squared distances between our target variable and predicted values. Although these values seem acceptable, considering the company's industrial environment and cost of failure, we need higher accuracy numbers. For example, using a confusion matrix that delivers TN, TP, FN, and FP values and assigning weight to each of these values can better estimate the performance of the ML models. For example, FP for a production line can be more costly than True Negative in some cases. Therefore, the thresholds of the evaluation metrics should be decided based on the production line and company requirements. In this work, we did not implement the threshold investigation.

Regressor Results Evaluation

In this dissertation, we implement several regressors to predict the remaining useful life of the pump. By utilizing the proposed PdM framework based on defined scenarios, Figure 5.29 shows the obtained results.

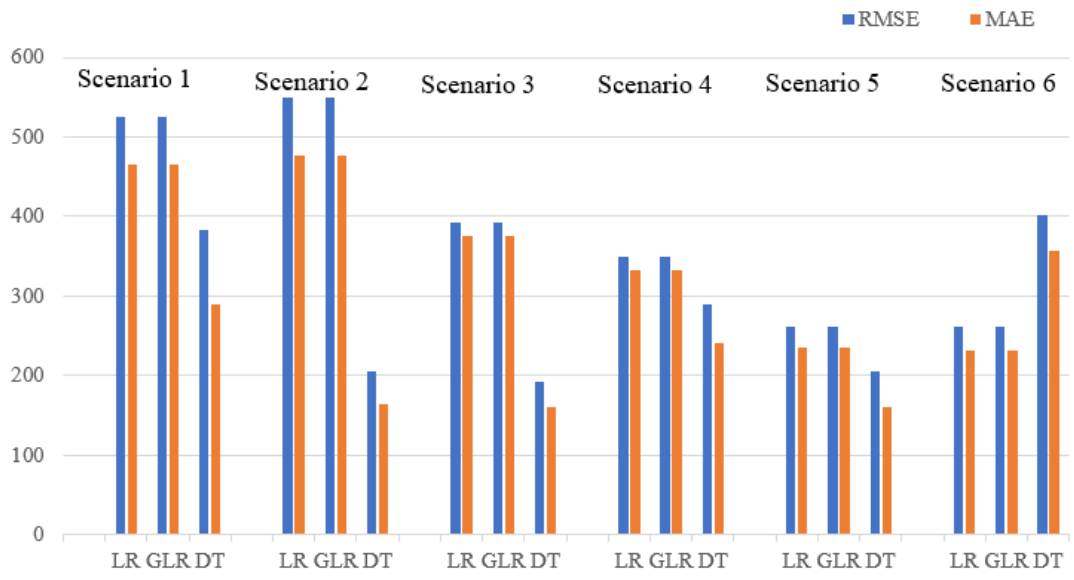

Figure 5.29: Comparing scenarios results for regressors implementation based on RMSE and MAE

The important thing about the regressors' results is the high error values. Due to the type of data that we are using for detecting the remaining useful life of the equipment, it can not give us a correct estimation. For RUL estimation, the collected data should be related to the age of the system. In this project's scope, having this data was not feasible as observing the aging of the machine requires a longer timespan. Thus, the dataset we are using can not deliver meaningful information about RUL to the ML algorithms. For regressor algorithms, we did not expect to obtain a good result. Nevertheless, the purpose is to implement and compare the impact of each technique on the final result. We make a comparison based on the employed techniques.

In scenario 5, as can be seen in Figure 5.29, almost for all three regressors, the RMSE and MAE values are less compared to other scenarios; however, scenario 1 has the worst result among the existing scenarios. Moreover, it is evident that among these regressors, the Decision Tree has the best result in almost all of them except scenario 6. Comparing scenarios 1 and 2 shows that the DT algorithm works very well when we do not extract the time-domain feature. The expectation that extracting time-domain features is helpful is not correct here, as we have a small dataset, and this approach can remove essential data.

From scenario 1 to scenario 3, there is a dramatic decrease for LR and GLR. Also, there is a slight decrease for DT, which means that employing PCA increases error for scenario 1. Moreover, discarding PCA in scenario 3 improves the obtained results. Even scenario 4 to 5 shows a noticeable decrease which is the result of discarding PCC.

These different scenarios show that using raw data and extracted features in combination with PCA and PCC should carefully be selected to have better results. Generally, PCA is suitable for recognizing the influence of high variance input variables on the target variables. However, comparing scenarios 5 and 6 shows an increase in error values by employing the PCA technique since our dataset has very low variances.

Classifiers Results Evaluation

We implement three classifiers to predict the condition of the pump (healthy or unhealthy) based on a multi-sensory dataset. Following the steps of the proposed framework, Figure 5.30 depicts the obtained results.

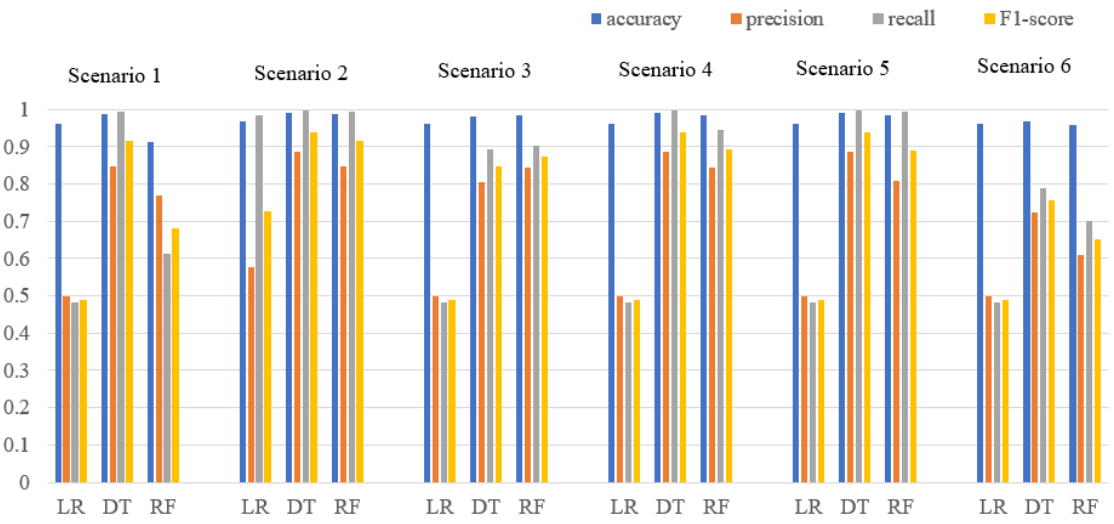

Figure 5.30: Comparing multiple scenarios for classifiers based on accuracy, precision, recall, and F1-score

We notice that the logistic regression model has the lowest performance in all the scenarios since this model requires independent variables and a large sample size to perform well. Furthermore, we can observe that scenario 2 has the best result for all the models due to employing PCA and PCC. Additionally, we can notice the impact of the extracted time-domain feature by comparing scenario 1 and scenario 2. Eliminating this feature significantly improves the outcome since we are losing valuable data by extracting the time-domain feature. It is due to the fact that our sampling frequency, i.e., 0.033HZ, is relatively low. Therefore, we do not have enough data to extract time-domain features. Comparing scenarios 2 and 4 shows that discarding the PCA results in lower performance in the logistic regression model. Likewise, comparing scenarios 4 and 6 shows a decrease in results that happened because of removing PCC.

Overall, we can conclude that the extraction of the time-domain feature has a negative impact on the performance of all the classifier models. Additionally, PCA improves the logistic regression model by

injecting more features. Finally, for all the scenarios, the combination of PCC and PCA techniques delivers the best results.

One-class SVM Results Evaluation

In this study, we implement one type of unary classification called one-class SVM, which Figure 5.31 represents the achieved results for the six scenarios. Comparing scenarios 1 and 2 shows that discarding the

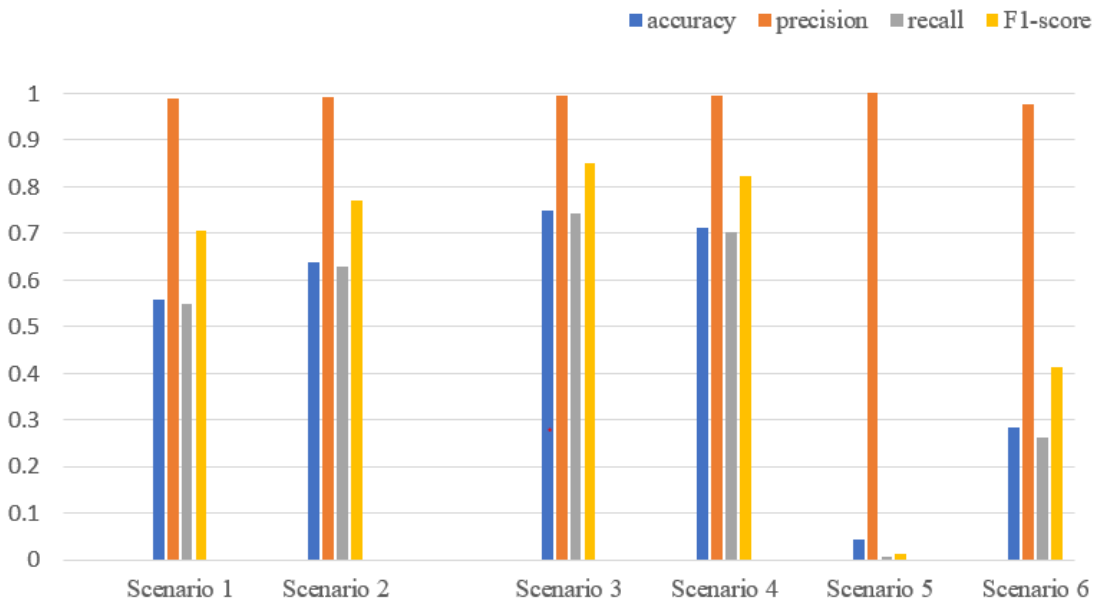

Figure 5.31: Comparing multiple scenarios for one-class support vector machine based on accuracy, precision, recall, and F1-score

time-feature extraction improves the outcome. Additionally, not using the scaled raw data along with time-feature extraction heightens the performance, as shown in scenario 3. Furthermore, considering scenarios 4 and 5, we can realize the critical role of the PCC. As removing it in scenario 5, decreases the accuracy, recall, and F1-score significantly. According to the obtained results, scenario 5 has the lowest performance, which does not utilize PCC and PCA. By enabling the PCA in scenario 6, we can observe a noticeable improvement that highlights PCA's impact.

Peak Detection Results

Based on the peak detection approach explained in Section 5.4.2, we analyzed our dataset, stored in MySQL, to detect changes in the number of peaks in the defined time window. To this end, we employed the Grafana processing engine for detecting the anomaly in the number of peaks. Additionally, it allows efficient monitoring in real-time, as well as creating dashboards for data visualization. We collected the speed of the pump for several days as displayed in Figure 5.32.

By zooming in, as in Figure 5.33, we can notice some fluctuations in the speed signal indicating an impulse operation behavior of the pump. The duration of these fluctuations varies, which can cause failure in case of an extended period, e.g., 30 minutes.

To detect this kind of anomaly, we defined an alert query in Grafana, which counts the number of peaks over a 3-minute time frame. As depicted in Figure 5.34, when the number of peaks exceeds 200 values, an email alert is sent.

5.5.2 Model Deployment

As described in Section 5.5, we have used several models: regressor, classifiers, one-class SVM, and peak detection in 6 scenarios. In this step, we should decide the most reliable one for the deployment in production. To this end, for regressor models, we consider the error value metrics, i.e., RMSE, MAE. Moreover, classifier and one-class SVM is evaluated based on accuracy, precision, recall, and F1-score metrics. Finally, since peak detection is a knowledge-based evaluation, the final decision is up to the experts.

According to the analysis presented in Section 5.5, scenario 5 for the regressor models, scenario 2 for the classifiers, and scenario 3 for one-class SVM have the most reliable results. Next, we optimized the employed model in the corresponding scenario to achieve better performance. Consequently, as shown in Figure 5.35, we optimized the GLR regressor for scenario 5 by increasing the *Lambda*. As a result, we observe a noticeable improvement in RMSE and MAE values for Lambda in the range of 0 to 100. However, increasing the Lambda by more than 100 results in saturated error values.

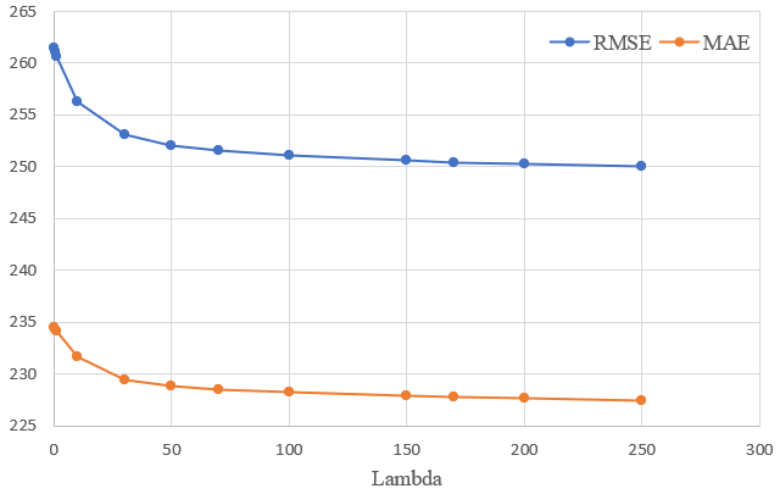

Figure 5.35: GLR regressor optimization with change of *lambda*

Furthermore, as presented in Figure 5.36, the DT regressor is optimized for scenario 5 by changing the depth and bin of the tree. Increasing the depth and bin to a specific limit obtained by try and error can improve the error values. While growing after that limit can harm the performance. Additionally, to optimize the one-class SVM for scenario 3, we decrease the *nu* parameter from 0.6 to 0.01. As depicted in Figure 5.37, nu value reduction has a significant impact on all the measured metrics. To further visualize the significant improvement in the accuracy, Figure 5.38 shows the impact of the optimization on one-class

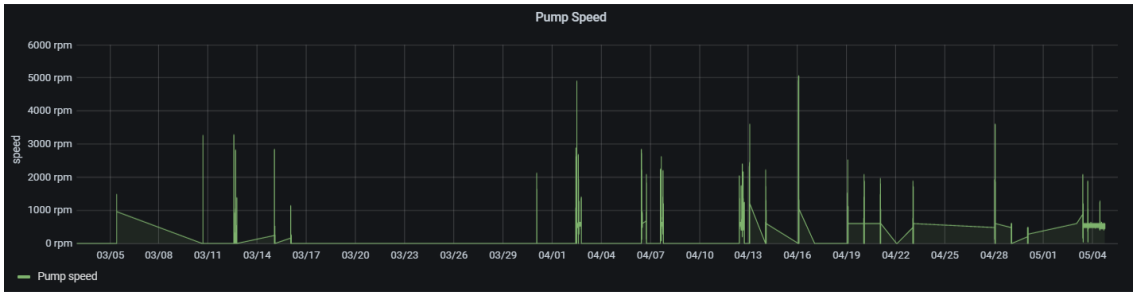

Figure 5.32: Big sample of data that collected for several days

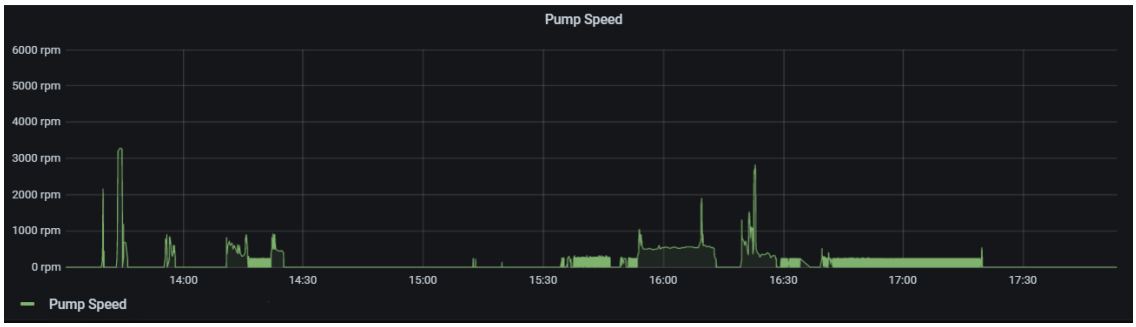

Figure 5.33: Moments represent several fluctuations in pump speed

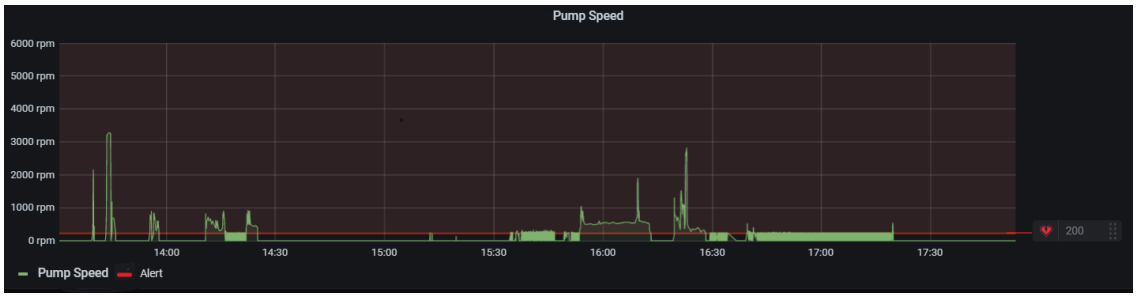

Figure 5.34: Setting alert for capturing fluctuation behavior in the pump's speed

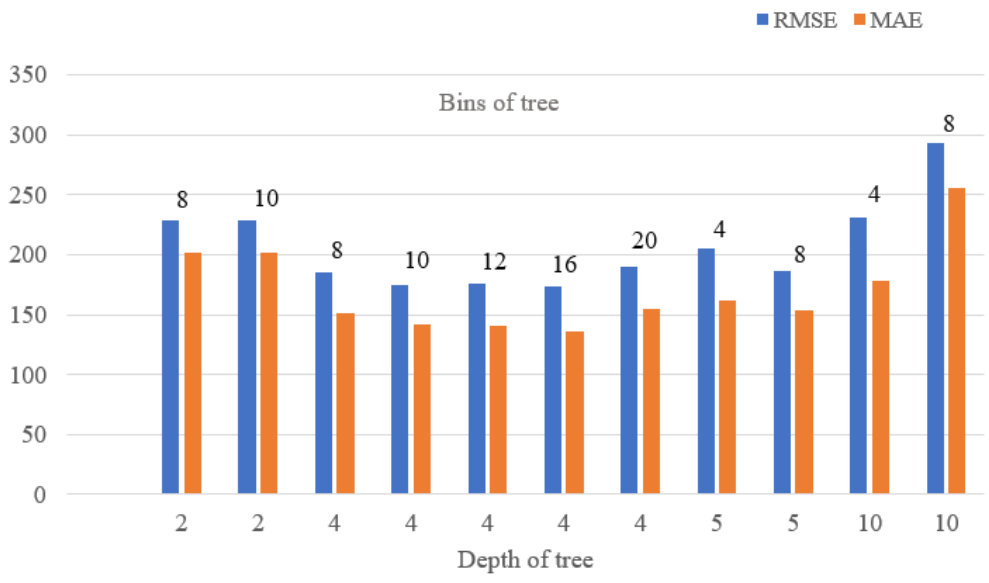

Figure 5.36: Decision tree optimization with change of depth and bins of the tree

SVM. Figure 5.38 shows the test dataset. Comparing the results before and after optimization, respectively shown in Figures 5.39 and 5.40, we realize remarkable precise predicted results.

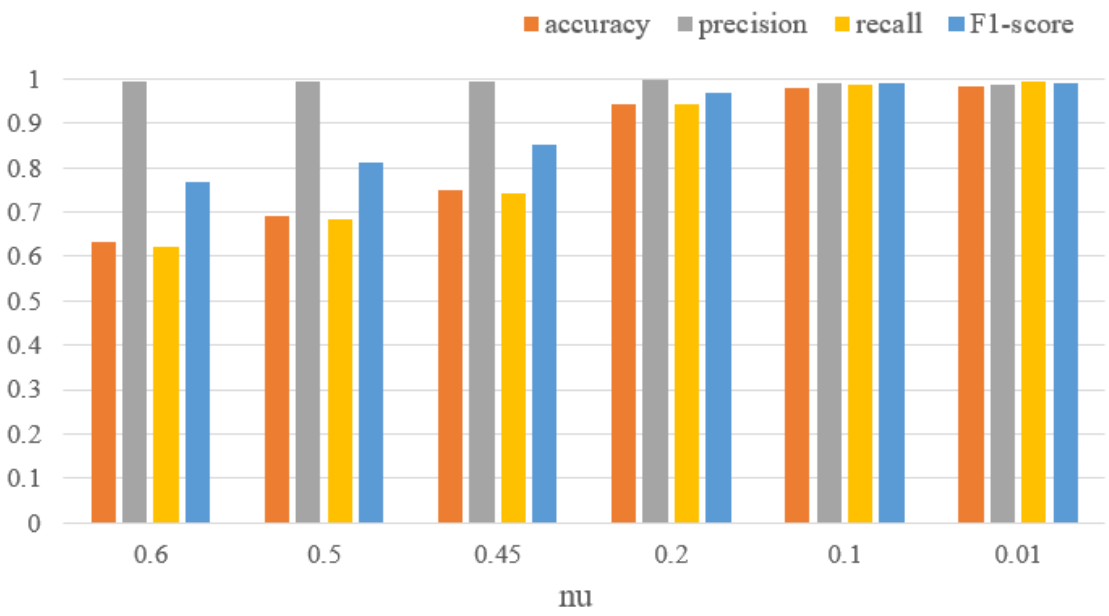

Figure 5.37: Optimization for one-class SVM with applying different *nu* values

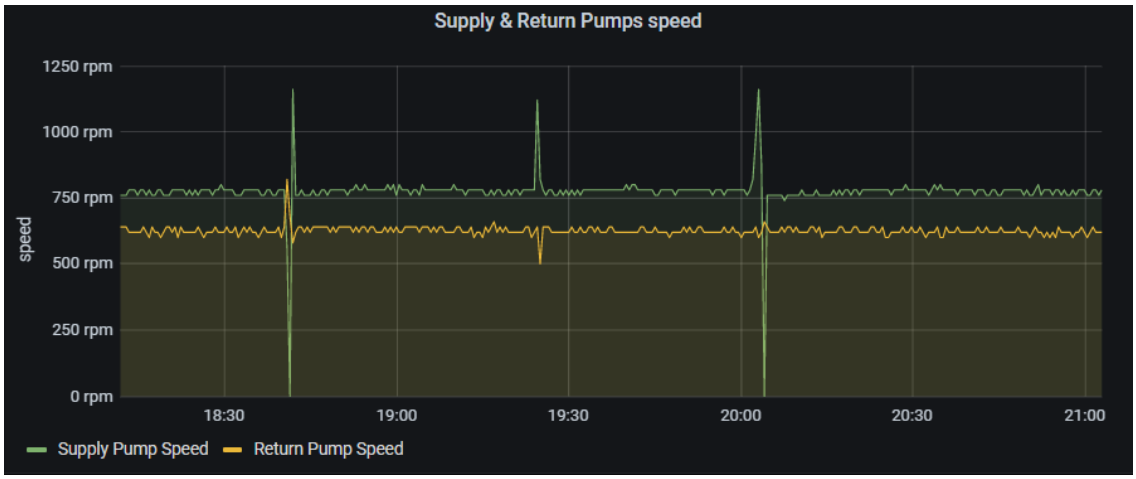

Figure 5.38: Test dataset

Overall, according to the above comparisons, regressor models, even after optimization, are not ready to be deployed in production. On the other hand, classifiers show promising results except for logistic regression. Moreover, one-class SVM performs adequately. Finally, peak detection evaluation is highly dependent on expert decisions. Worth noting, the deployment of the model is out of the scope of this work. However, the chosen model can be deployed either on the cloud or locally.

5.5.3 Model Improvement

According to the final stage in the result evaluation layer of our framework, we should improve the deployed model. To this end, the experts are required to define a threshold for accuracy and error levels. If the model evaluation results are below the threshold values, a message is sent to the upper layer, i.e., decision making, and the maintenance team is notified by an alarm.

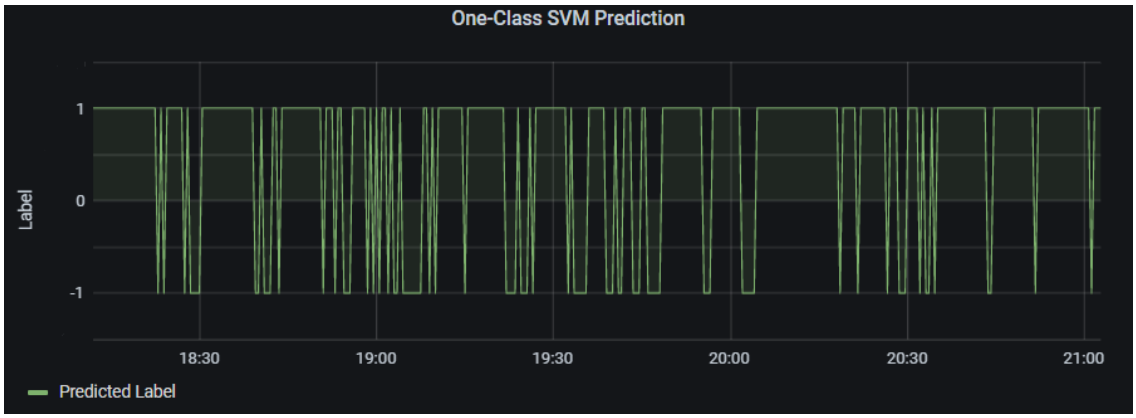

Figure 5.39: One-class SVM prediction before optimization (nu=0.45)

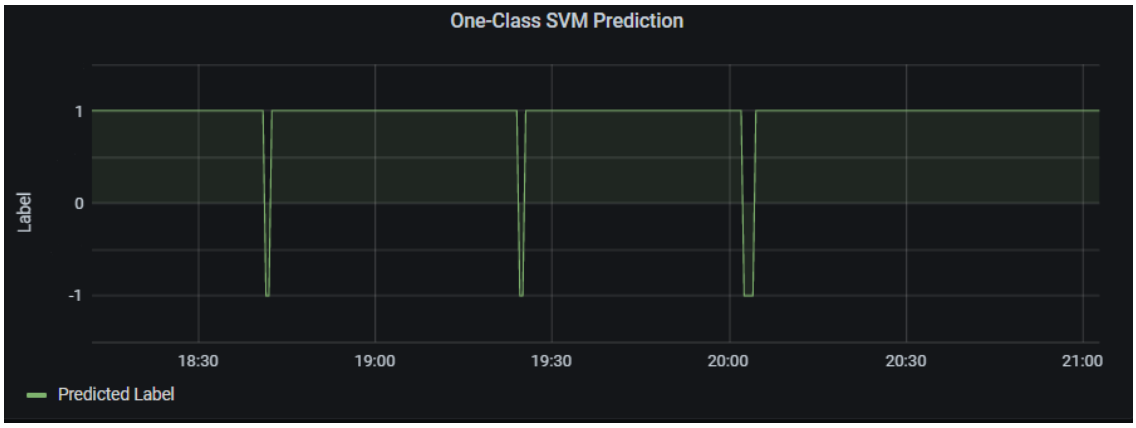

Figure 5.40: One-class SVM prediction after optimization (nu=0.01)

5.6 Decision Making Implementation

To implement the last layer, we use the obtained results from the result evaluation layer to make certain decisions and act accordingly. The Decision Making layer consists of two modules, namely Setting Alarm System and Maintenance Strategy.

5.6.1 Setting Alarm System

For our alarm system implementation, we set up a notification channel between Grafana and an SMTP server, i.e., Gmail, to warn when a deviation happens. In our case, we configured Gmail as the SMTP server. After establishing the connection to the SMTP server and setting the threshold on the Grafana, as shown in Figure 5.41, we can receive alert emails in case a change happens in the predicted results. Figure 5.42 displays a sample alert email that can be sent to multiple emails at the same time.

Alternatively, we can use Power Bi [124] for visualization and alarm systems instead of Grafana, which is available either on the cloud or on-premise. Although it provides a user-friendly GUI and a powerful processing engine, the on-premise open source version does not support an alert mechanism. Hence, we could not utilize it in our study.

5.6.2 Maintenance Strategy

In the Maintenance Strategy module, expert decision makers implement the necessary strategies for maintenance. The result of this module is used to react to the physical world to achieve a particular purpose, such

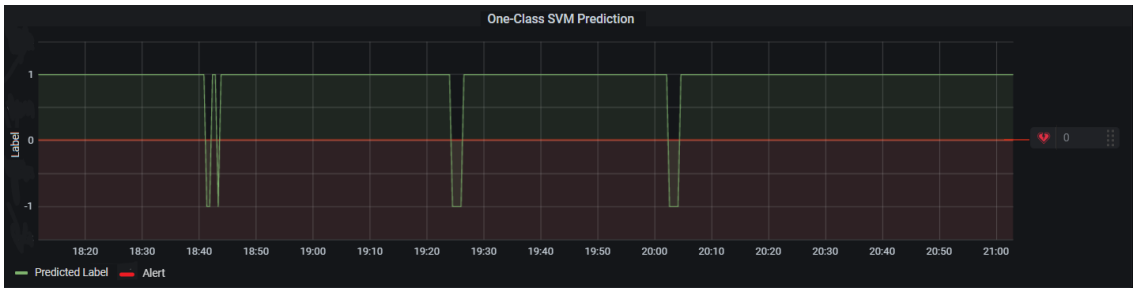

Figure 5.41: Setting alert threshold for predicted results

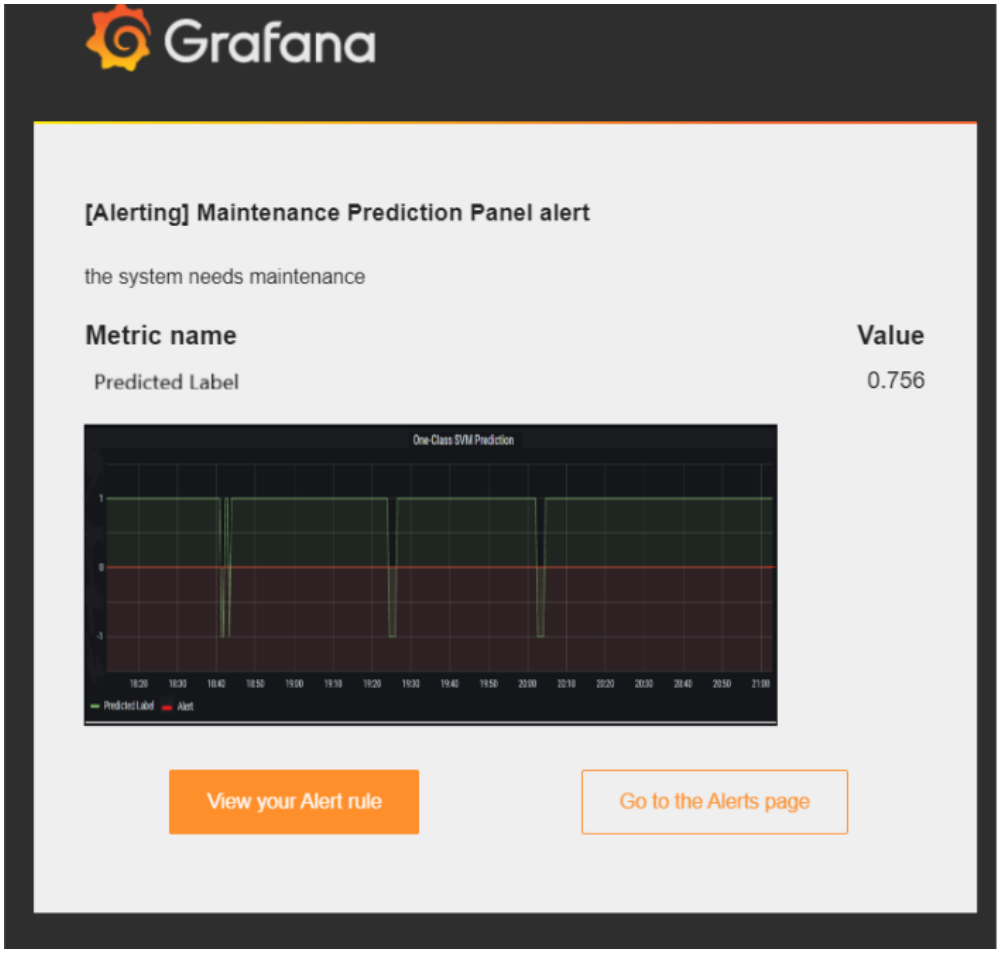

Figure 5.42: Receiving maintenance alert in a Gmail

as minimizing the cost of maintenance, realizing zero-defect manufacturing, and reducing the breakdown. Additionally, based on the outcome of the Setting Alarm System module, we can include the function of error correction, compensation, and feedback control to continue to run the equipment and process in a healthy condition.

There are different techniques to correct and compensate for the errors. Nonetheless, in our study, after receiving the alert email indicating a critical situation of the pump, NTS experts, based on the symptoms of failures, should decide about the maintenance strategy, e.g., modifying control parameters to resolve the failure. However, we could not implement this module as it depends on the control engineer's decision which was not available.

Chapter 6

Conclusion and Future Work

This chapter presents the conclusions of the project. The first part focuses on providing answers to the research questions. In the second part, we propose some suggestions for possible future work to continue this project.

6.1 Conclusion

This dissertation set out to propose a predictive maintenance framework. The primary motivation behind this work was to act against upcoming maintenance work proactively to lower the cost. To perfectly be utilized in IoT-enabled smart companies and factories, we designed our proposed framework based on the data-driven model and employed data analytics algorithms.

Most of the existing work in the domain of predictive maintenance has significantly focused on developing various anomaly detection techniques through simulations. However, in this study, we have aimed at designing an end-to-end PdM framework. Therefore, we have further evaluated our proposed framework using real-world manufacturing data. The obtained results have demonstrated the effectiveness of our proposed framework.

In order to answer the first research question of this work 1.3, we have conducted an SLR on 27 existing predictive maintenance research papers. Furthermore, we have classified the algorithms and techniques addressed in these studies according to the multiple phases of an end-to-end predictive maintenance project. In summary, the results of our SLR has shown that:

- 48% of the studies discussed how the data could be acquired for a predictive maintenance project,
- 11% of the studies presented some solutions for cleaning the data sets,
- 37% of the studies suggested some feature engineering techniques to prepare the data for some algorithms,
- 15% of the studies proposed new algorithms to predict the upcoming maintenance based on the past data, and finally
- 22% of the studies provided business decision making dashboards and guidelines for the companies.

After performing our SLR study, we have addressed the second research question 1.3. The primary purpose of this question is to suggest a general predictive maintenance framework that is highly applicable per different use cases. Consequently, We have designed an end-to-end predictive maintenance framework that covers all the phases of a predictive maintenance project discovered in the previous research question. The proposed predictive maintenance framework consists of 5 layers:

- *Data Acquisition*, which addresses sampling schema, Data transfer, source of data and data storage, and Data visualization
- *Data Preprocessing*, which contains data cleaning, data enrichment and correlation, feature engineering, and dimension reduction techniques

- *Predictive Analytics*, which is divided into three groups: supervised learning, Semi-supervised learning, and unsupervised learning
- *Result Evaluation*, which supports the Supervised and semi-supervised model evaluation metrics, Unsupervised Evaluation metrics, Model deployment, and Model improvement
- *Decision Making*, which provides setting alarm system and maintenance strategy

Finally, to answer the third research question (i.e., to evaluate our proposed framework), we have implemented a case study at NTS Group. In this case study, we have illustrated a scenario in which a pump installed in a printing machine was out of work after a couple of days. In particular, we employed the NTS testbed and conducted several experiments to assess the impact of each block of the proposed framework. Then, considering the impact of each block in the result, we decided on the strategy for selecting the techniques.

In this case study, we have employed several supervised and unsupervised learning techniques. Particularly, three regressors were utilized: *Linear Regression*, *Generalized Linear Regression* and *Decision Tree*. Due to the lack of some available information, such as the historical *Remaining Useful Life* of the pump, we did not expect to obtain this much precise results from our analysis. To overcome this limitation, we have made some assumptions in our preprocessing steps. The results of our analysis show that PCA has a negative impact and, eventually, leads to more errored values. Interestingly, in most of the experiments, the Decision Tree shows the best results in terms of *Root Squared Error* and *Mean Absolute Error*. We have also employed three classifiers: *Logistic Regression*, *Decision Tree* and *Random Forest*. For the test data set, the classifiers performed quite well and obtained significant accuracy of 95%, 99%, and 98%, respectively, for LR, DT and RF. As expected, LR had the lowest performance among all the other classifiers in all the experiments. Moreover, involving the PCC technique for classifiers increased the recall by around 5% and at the same time decreased precision by about 4% for Random Forest. As mentioned before, we have also utilized the One-Class SVM, which is a semi-supervised learning algorithm. Applying the optimized version of this classifier resulted in 99% for all of our evaluation metrics. These different algorithms and techniques demonstrate that using raw data, extracted features combined with PCA and PCC should be selected carefully to achieve better results, not deteriorating the result. Finally, we used *Peak Detection* as an unsupervised method to detect whether the number of peaks in the speed signal of the pump exceeds a predefined threshold and implement an alert system to warn the experts accordingly.

We stored all the results of ML algorithms in a MySQL database connected to the Grafana processing engine to visualize further and demonstrate the results. Besides these ML algorithms, the preprocessing steps such as scaling, feature extraction, feature selection, dimension reduction, and feature selection were analyzed to achieve better accuracy and fewer error values. Accordingly, in almost all test scenarios, time-domain feature extraction caused less accuracy and higher error as extracting this feature in low sampling frequency resulted in discarding useful information. Generally, in training ML algorithms, the size of the data matters a lot. On the other hand, when we work with time-domain features, sampling frequency plays a crucial role in the results. Consequently, we can argue that the size of the data and higher sampling frequency can highly impact the algorithm's quality. Nevertheless, another important technique for alerting results is feature importance. We have shown that PCC is more valuable than PCA in the case of low variance data. Based on the results, employing PCA alone can deteriorate the results. However, in combination with PCC, it can alter the result negligibly in our dataset.

By summarizing, our research brings the following scientific achievements:

- A complete framework for failure prediction and analysis in emerging Industry 4.0 settings.
- A truly test over a real-life case study represented by a printing machine.

Please note that this case study had multiple limitations, mainly related to the duration of the experiment and the small size of the collected data. In addition to this, our research also brings the following managerial achievements:

- Failure prediction and analysis are a critical process of every organization that falls in the broader Industry 4.0 setting; therefore, a common standardization action should be necessary (e.g., method-

ologies, processes, software interfaces, etc.);
- In Industry 4.0 settings, the complex failure prediction and analysis process is not only a job for data scientists but a multidisciplinary team should be formed to include expertise from every domain (e.g., the mechanical and electrical area);
- Predictive maintenance is one of the critical future assets of next generation Industry 4.0 big data-driven organizations; hence it should play a more significant role within stakeholders in future years.

6.2 Future Work

Currently, it should be noted that the framework has been applied to one use-case study so far. As future work, we plan to test this framework on other machines and more use-cases to verify the results and findings from this work. Furthermore, considering other machine learning models such as artificial neural networks and other feature engineering methods, there is a significant opportunity to improve the predicted results.

Another aspect of future work on the framework is employing cloud analytics services. Having utilized these types of services, the framework can be way more scalable when the data size is bigger and, thus, more resources are required. Therefore, an exciting direction for future work is to run a cost comparison between the on-premise realization of the framework versus on-cloud implementation.

Additionally, a work can be devoted to deriving key performance indicators by combining the essential features discovered, deploying the model in a production environment, deriving a failure probability from the multi-failure prediction output, and clustering failure modes to understand better the failure probability root cause of the various faults.

Considering the limitations of the case study, we finally suggest NTS group collect the data for at least one year to improve the training dataset. Further work will also focus on implementing the framework for all 4 HSU of the printing machines to test the scaling limits over multiple HSUs.

Bibliography

[1] Michele Compare, Piero Baraldi and Enrico Zio. “Challenges to IoT-Enabled Predictive Maintenance for Industry 4.0”. In: *IEEE Internet of Things Journal* 7.5 (2019), pp. 4585–4597.

[2] Mario Hermann, Tobias Pentek and Boris Otto. “Design principles for industrie 4.0 scenarios”. In: *2016 49th Hawaii international conference on system sciences (HICSS)*. IEEE. 2016, pp. 3928–3937.

[3] Rainer Drath and Alexander Horch. “Industrie 4.0: Hit or hype? [industry forum]”. In: *IEEE industrial electronics magazine* 8.2 (2014), pp. 56–58.

[4] Heiner Lasi et al. “Industry 4.0”. In: *Business & information systems engineering* 6.4 (2014), pp. 239–242.

[5] Ray Y Zhong et al. “Intelligent manufacturing in the context of industry 4.0: a review”. In: *Engineering* 3.5 (2017), pp. 616–630.

[6] Weiting Zhang, Dong Yang and Hongchao Wang. “Data-driven methods for predictive maintenance of industrial equipment: A survey”. In: *IEEE Systems Journal* 13.3 (2019), pp. 2213–2227.

[7] Rui Zhao et al. “Deep learning and its applications to machine health monitoring”. In: *Mechanical Systems and Signal Processing* 115 (2019), pp. 213–237.

[8] Zhe Li, Yi Wang and Ke-Sheng Wang. “Intelligent predictive maintenance for fault diagnosis and prognosis in machine centers: Industry 4.0 scenario”. In: *Advances in Manufacturing* 5.4 (2017), pp. 377–387.

[9] Erim Sezer et al. “An industry 4.0-enabled low cost predictive maintenance approach for smes”. In: *2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*. IEEE. 2018, pp. 1–8.

[10] Eric Levrat, Benoit Iung and Adolfo Crespo Marquez. “E-maintenance: review and conceptual framework”. In: *Production Planning & Control* 19.4 (2008), pp. 408–429.

[11] Detlef Zuehlke. “SmartFactory—Towards a factory-of-things”. In: *Annual reviews in control* 34.1 (2010), pp. 129–138.

[12] Jay Lee, Behrad Bagheri and Hung-An Kao. “A cyber-physical systems architecture for industry 4.0-based manufacturing systems”. In: *Manufacturing letters* 3 (2015), pp. 18–23.

[13] Thomas Bangemann et al. “PROTEUS—Creating distributed maintenance systems through an integration platform”. In: *Computers in industry* 57.6 (2006), pp. 539–551.

[14] Behrad Bagheri et al. “Cyber-physical systems architecture for self-aware machines in industry 4.0 environment”. In: *IFAC-PapersOnLine* 48.3 (2015), pp. 1622–1627.

[15] Yoji Yamato, Hiroki Kumazaki and Yoshifumi Fukumoto. “Proposal of lambda architecture adoption for real time predictive maintenance”. In: *2016 fourth international symposium on computing and networking (CANDAR)*. IEEE. 2016, pp. 713–715.

[16] Bernard Schmidt and Lihui Wang. “Cloud-enhanced predictive maintenance”. In: *The International Journal of Advanced Manufacturing Technology* 99.1-4 (2018), pp. 5–13.

[17] Lin Zhang et al. “Cloud manufacturing: a new manufacturing paradigm”. In: *Enterprise Information Systems* 8.2 (2014), pp. 167–187.

[18] Sara Landset et al. “A survey of open source tools for machine learning with big data in the Hadoop ecosystem”. In: *Journal of Big Data* 2.1 (2015), p. 24.

[19] Gian Antonio Susto et al. “An adaptive machine learning decision system for flexible predictive maintenance”. In: *2014 IEEE International Conference on Automation Science and Engineering (CASE)*. IEEE. 2014, pp. 806–811.

[20] Andrew KS Jardine, Daming Lin and Dragan Banjevic. “A review on machinery diagnostics and prognostics implementing condition-based maintenance”. In: *Mechanical systems and signal processing* 20.7 (2006), pp. 1483–1510.

[21] Hamid Reza Golmakani. “Optimal age-based inspection scheme for condition-based maintenance using A* search algorithm”. In: *International journal of production research* 50.23 (2012), pp. 7068–7080.

[22] Qiushi Zhu, Hao Peng and Geert-Jan van Houtum. “A condition-based maintenance policy for multi-component systems with a high maintenance setup cost”. In: *Or Spectrum* 37.4 (2015), pp. 1007–1035.

[23] Dothang Truong. “How cloud computing enhances competitive advantages: A research model for small businesses”. In: *The Business Review, Cambridge* 15.1 (2010), pp. 59–65.

[24] Zhe Li, Kesheng Wang and Yafei He. “Industry 4.0-potentials for predictive maintenance”. In: *Advances in Economics, Business and Management Research* (2016).

[25] Geert Waeyenbergh and Liliane Pintelon. “A framework for maintenance concept development”. In: *International journal of production economics* 77.3 (2002), pp. 299–313.

[26] Jim Daily and Jeff Peterson. “Predictive maintenance: How big data analysis can improve maintenance”. In: *Supply Chain Integration Challenges in Commercial Aerospace*. Springer, 2017, pp. 267–278.

[27] Katerina Lepenioti et al. “Prescriptive analytics: Literature review and research challenges”. In: *International Journal of Information Management* 50 (2020), pp. 57–70.

[28] Jian Qin, Ying Liu and Roger Grosvenor. “A categorical framework of manufacturing for industry 4.0 and beyond”. In: *Procedia cirp* 52 (2016), pp. 173–178.

[29] Jay Lee. “Industry 4.0 in big data environment”. In: *German Harting Magazine* 1.1 (2013), pp. 8–10.

[30] DIN Spec. “91345: Referenzarchitekturmodell Industrie 4.0 (RAMI4. 0)”. In: *Deutsches Institut für Normung (DIN) eV* (2016).

[31] Amy JC Trappey et al. “A review of essential standards and patent landscapes for the Internet of Things: A key enabler for Industry 4.0”. In: *Advanced Engineering Informatics* 33 (2017), pp. 208–229.

[32] Andrew KS Jardine, Daming Lin and Dragan Banjevic. “A review on machinery diagnostics and prognostics implementing condition-based maintenance”. In: *Mechanical systems and signal processing* 20.7 (2006), pp. 1483–1510.

[33] JZ Sikorska, Melinda Hodkiewicz and Lin Ma. “Prognostic modelling options for remaining useful life estimation by industry”. In: *Mechanical systems and signal processing* 25.5 (2011), pp. 1803–1836.

[34] Earl Cox. “Fuzzy fundamentals”. In: *IEEE spectrum* 29.10 (1992), pp. 58–61.

[35] Amulya K Garga et al. “Hybrid reasoning for prognostic learning in CBM systems”. In: *2001 IEEE Aerospace Conference Proceedings (Cat. No. 01TH8542)*. Vol. 6. IEEE. 2001, pp. 2957–2969.

[36] Xiao-Sheng Si et al. “A Wiener-process-based degradation model with a recursive filter algorithm for remaining useful life estimation”. In: *Mechanical Systems and Signal Processing* 35.1-2 (2013), pp. 219–237.

[37] Ming Dong and David He. "Hidden semi-Markov model-based methodology for multi-sensor equipment health diagnosis and prognosis". In: *European Journal of Operational Research* 178.3 (2007), pp. 858–878.

[38] Bhaskar Saha, Kai Goebel and Jon Christophersen. "Comparison of prognostic algorithms for estimating remaining useful life of batteries". In: *Transactions of the Institute of Measurement and Control* 31.3-4 (2009), pp. 293–308.

[39] Marcos E Orchard and George J Vachtsevanos. "A particle-filtering approach for on-line fault diagnosis and failure prognosis". In: *Transactions of the Institute of Measurement and Control* 31.3-4 (2009), pp. 221–246.

[40] Michael J Roemer and Gregory J Kacprzynski. "Advanced diagnostics and prognostics for gas turbine engine risk assessment". In: *2000 IEEE Aerospace Conference. Proceedings (Cat. No. 00TH8484)*. Vol. 6. IEEE. 2000, pp. 345–353.

[41] Dazhong Wu et al. "Cloud-based machine learning for predictive analytics: Tool wear prediction in milling". In: *2016 IEEE International Conference on Big Data (Big Data)*. IEEE. 2016, pp. 2062–2069.

[42] Zhiwei Gao, Carlo Cecati and Steven X Ding. "A survey of fault diagnosis and fault-tolerant techniques—Part I: Fault diagnosis with model-based and signal-based approaches". In: *IEEE Transactions on Industrial Electronics* 62.6 (2015), pp. 3757–3767.

[43] M El Hachemi Benbouzid. "A review of induction motors signature analysis as a medium for faults detection". In: *IEEE transactions on industrial electronics* 47.5 (2000), pp. 984–993.

[44] Subhasis Nandi, Hamid A Toliyat and Xiaodong Li. "Condition monitoring and fault diagnosis of electrical motors—A review". In: *IEEE transactions on energy conversion* 20.4 (2005), pp. 719–729.

[45] Gojko M Joksimović et al. "Stator-current spectrum signature of healthy cage rotor induction machines". In: *IEEE Transactions on Industrial Electronics* 60.9 (2012), pp. 4025–4033.

[46] Xiang Gong and Wei Qiao. "Bearing fault diagnosis for direct-drive wind turbines via current-demodulated signals". In: *IEEE Transactions on Industrial Electronics* 60.8 (2013), pp. 3419–3428.

[47] Zhipeng Feng, Ming Liang and Fulei Chu. "Recent advances in time–frequency analysis methods for machinery fault diagnosis: A review with application examples". In: *Mechanical Systems and Signal Processing* 38.1 (2013), pp. 165–205.

[48] Jason JA Costello, Graeme M West and Stephen DJ McArthur. "Machine learning model for event-based prognostics in gas circulator condition monitoring". In: *IEEE Transactions on Reliability* 66.4 (2017), pp. 1048–1057.

[49] Yuchen Jiang and Shen Yin. "Recursive total principle component regression based fault detection and its application to vehicular cyber-physical systems". In: *IEEE Transactions on Industrial Informatics* 14.4 (2017), pp. 1415–1423.

[50] Deyong You, Xiangdong Gao and Seiji Katayama. "WPD-PCA-based laser welding process monitoring and defects diagnosis by using FNN and SVM". In: *IEEE Transactions on Industrial Electronics* 62.1 (2014), pp. 628–636.

[51] KK McKee et al. "A review of machinery diagnostics and prognostics implemented on a centrifugal pump". In: *Engineering asset management 2011*. Springer, 2014, pp. 593–614.

[52] Jay Lee, Hung-An Kao, Shanhu Yang et al. "Service innovation and smart analytics for industry 4.0 and big data environment". In: *Procedia Cirp* 16.1 (2014), pp. 3–8.

[53] Mikel Canizo et al. "Real-time predictive maintenance for wind turbines using Big Data frameworks". In: *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*. IEEE. 2017, pp. 70–77.

[54] Jay Lee et al. "Intelligent prognostics tools and e-maintenance". In: *Computers in industry* 57.6 (2006), pp. 476–489.

[55] Dragan Djurdjanovic, Jay Lee and Jun Ni. "Watchdog Agent—an infotronics-based prognostics approach for product performance degradation assessment and prediction". In: *Advanced Engineering Informatics* 17.3-4 (2003), pp. 109–125.

[56] Mari Cruz Garcia, Miguel A Sanz-Bobi and Javier Del Pico. "SIMAP: Intelligent System for Predictive Maintenance: Application to the health condition monitoring of a windturbine gearbox". In: *Computers in Industry* 57.6 (2006), pp. 552–568.

[57] Wenjin Yu et al. "A global manufacturing big data ecosystem for fault detection in predictive maintenance". In: *IEEE Transactions on Industrial Informatics* 16.1 (2019), pp. 183–192.

[58] Jinjiang Wang et al. "A new paradigm of cloud-based predictive maintenance for intelligent manufacturing". In: *Journal of Intelligent Manufacturing* 28.5 (2017), pp. 1125–1137.

[59] Nathan Marz and James Warren. *Big Data: Principles and best practices of scalable real-time data systems*. New York; Manning Publications Co., 2015.

[60] Barbara Kitchenham. "Procedures for performing systematic reviews". In: *Keele, UK, Keele University* 33.2004 (2004), pp. 1–26.

[61] Barbara A Kitchenham, Emilia Mendes and Guilherme H Travassos. "Cross versus within-company cost estimation studies: A systematic review". In: *IEEE Transactions on Software Engineering* 33.5 (2007), pp. 316–329.

[62] *ACM digital library*. `https://dl.acm.org/`. Accessed: 2021-01-14.

[63] *IEEE Xplore*. `https://ieeexplore.ieee.org/`. Accessed: 2021-01-10.

[64] *ScienceDirect*. `https://www.sciencedirect.com/`. Accessed: 2021-01-15.

[65] *Multidisciplinary Digital Publishing Institute (MDPI)*. `http://www.mdpi.com/`. Accessed: 2021-01-12.

[66] *Springer*. `https://link.springer.com/`. Accessed: 2021-01-12.

[67] Yassine Bouabdallaoui et al. "Predictive Maintenance in Building Facilities: A Machine Learning-Based Approach". In: *Sensors* 21.4 (2021), p. 1044.

[68] Jose-Raul Ruiz-Sarmiento et al. "A predictive model for the maintenance of industrial machinery in the context of industry 4.0". In: *Engineering Applications of Artificial Intelligence* 87 (2020), p. 103289.

[69] Jack CP Cheng et al. "Data-driven predictive maintenance planning framework for MEP components based on BIM and IoT using machine learning algorithms". In: *Automation in Construction* 112 (2020), p. 103087.

[70] Wenjin Yu et al. "A global manufacturing big data ecosystem for fault detection in predictive maintenance". In: *IEEE Transactions on Industrial Informatics* 16.1 (2019), pp. 183–192.

[71] Matteo Calabrese et al. "SOPHIA: An event-based IoT and machine learning architecture for predictive maintenance in industry 4.0". In: *Information* 11.4 (2020), p. 202.

[72] Qinhua Hu et al. "A new online approach for classification of pumps vibration patterns based on intelligent IoT system". In: *Measurement* 151 (2020), p. 107138.

[73] Simone Panicucci et al. "A cloud-to-edge approach to support predictive analytics in robotics industry". In: *Electronics* 9.3 (2020), p. 492.

[74] Fotis Foukalas. "Cognitive IoT platform for fog computing industrial applications". In: *Computers & Electrical Engineering* 87 (2020), p. 106770.

[75] Ke Xu et al. "Advanced Data Collection and Analysis in Data-Driven Manufacturing Process". In: *Chinese Journal of Mechanical Engineering* 33 (2020), pp. 1–21.

[76] Tanvir Alam Shifat and Jang-Wook Hur. "EEMD assisted supervised learning for the fault diagnosis of BLDC motor using vibration signal". In: *Journal of Mechanical Science and Technology* 34.10 (2020), pp. 3981–3990.

[77] Adebena Oluwasegun and Jae-Cheon Jung. "The application of machine learning for the prognostics and health management of control element drive system". In: *Nuclear Engineering and Technology* 52.10 (2020), pp. 2262–2273.

[78] Rocco Langone, Alfredo Cuzzocrea and Nikolaos Skantzos. "Interpretable Anomaly Prediction: Predicting anomalous behavior in industry 4.0 settings via regularized logistic regression tools". In: *Data & Knowledge Engineering* 130 (2020), p. 101850.

[79] Farzam Farbiz, Yuan Miaolong and Zhou Yu. "A Cognitive Analytics based Approach for Machine Health Monitoring, Anomaly Detection, and Predictive Maintenance". In: *2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA)*. IEEE. 2020, pp. 1104–1109.

[80] Giovanni Gravito de Carvalho Chrysostomo et al. "A Framework for Big Data Analytical Process and Mapping—BAProM: Description of an Application in an Industrial Environment". In: *Energies* 13.22 (2020), p. 6014.

[81] Salvatore Cavalieri and Marco Giuseppe Salafia. "A Model for Predictive Maintenance Based on Asset Administration Shell". In: *Sensors* 20.21 (2020), p. 6028.

[82] Vincent Ciancio et al. "Towards prediction of machine failures: overview and first attempt on specific automotive industry application". In: *IFAC-PapersOnLine* 53.3 (2020), pp. 289–294.

[83] Emiliano Traini et al. "Machine learning framework for predictive maintenance in milling". In: *IFAC-PapersOnLine* 52.13 (2019), pp. 177–182.

[84] Riccardo Pinto and Tania Cerquitelli. "Robot fault detection and remaining life estimation for predictive maintenance". In: *Procedia Computer Science* 151 (2019), pp. 709–716.

[85] Stefano Proto et al. "PREMISES, a scalable data-driven service to predict alarms in slowly-degrading multi-cycle industrial processes". In: *2019 IEEE International Congress on Big Data (BigDataCongress)*. IEEE. 2019, pp. 139–143.

[86] Pauline Ong et al. "Efficient gear fault feature selection based on moth-flame optimisation in discrete wavelet packet analysis domain". In: *Journal of the Brazilian Society of Mechanical Sciences and Engineering* 41.6 (2019), pp. 1–14.

[87] Sai Van Cuong and Prof Maxim Shcherbakov. "PdM: A predictive maintenance modeling tool implemented as R-package and web-application". In: *Proceedings of the Tenth International Symposium on Information and Communication Technology*. 2019, pp. 433–440.

[88] Ricardo Silva Peres et al. "IDARTS–Towards intelligent data analysis and real-time supervision for industry 4.0". In: *Computers in industry* 101 (2018), pp. 138–146.

[89] Insun Shin et al. "A framework for prognostics and health management applications toward smart manufacturing systems". In: *International Journal of Precision Engineering and Manufacturing-Green Technology* 5.4 (2018), pp. 535–554.

[90] Farzana Kabir, Brandon Foggo and Nanpeng Yu. "Data driven predictive maintenance of distribution transformers". In: *2018 China International Conference on Electricity Distribution (CICED)*. IEEE. 2018, pp. 312–316.

[91] Farzan Majdani, Andrei Petrovski and Daniel Doolan. "Evolving ANN-based sensors for a context-aware cyber physical system of an offshore gas turbine". In: *Evolving systems* 9.2 (2018), pp. 119–133.

[92] Zhe Li, Yi Wang and Ke-Sheng Wang. "Intelligent predictive maintenance for fault diagnosis and prognosis in machine centers: Industry 4.0 scenario". In: *Advances in Manufacturing* 5.4 (2017), pp. 377–387.

[93] Behrad Bagheri et al. "Cyber-physical systems architecture for self-aware machines in industry 4.0 environment". In: *IFAC-PapersOnLine* 48.3 (2015), pp. 1622–1627.

[94] Bernhard Mueller. "Additive manufacturing technologies–Rapid prototyping to direct digital manufacturing". In: *Assembly Automation* (2012).

[95] Dong-Yeul Song et al. “A new approach to cutting state monitoring in end-mill machining”. In: *International Journal of Machine Tools and Manufacture* 45.7-8 (2005), pp. 909–921.

[96] G Campatelli and A Scippa. “Prediction of milling cutting force coefficients for Aluminum 6082-T4”. In: *Procedia Cirp* 1 (2012), pp. 563–568.

[97] Yalcin M Ertekin, Yongjin Kwon and Tzu-Liang Bill Tseng. “Identification of common sensory features for the control of CNC milling operations under varying cutting conditions”. In: *International Journal of Machine Tools and Manufacture* 43.9 (2003), pp. 897–904.

[98] Y Altintas, M Eynian and H Onozuka. “Identification of dynamic cutting force coefficients and chatter stability with process damping”. In: *CIRP annals* 57.1 (2008), pp. 371–374.

[99] OB Abouelatta and J Madl. “Surface roughness prediction based on cutting parameters and tool vibrations in turning operations”. In: *Journal of materials processing technology* 118.1-3 (2001), pp. 269–277.

[100] A Verl et al. “Sensorless automated condition monitoring for the control of the predictive maintenance of machine tools”. In: *CIRP annals* 58.1 (2009), pp. 375–378.

[101] Roberto Teti et al. “Advanced monitoring of machining operations”. In: *CIRP annals* 59.2 (2010), pp. 717–739.

[102] Panling Huang et al. “Vibration analysis in milling titanium alloy based on signal processing of cutting force”. In: *The International Journal of Advanced Manufacturing Technology* 64.5-8 (2013), pp. 613–621.

[103] Rui Zhao et al. “Machine health monitoring using local feature-based gated recurrent unit networks”. In: *IEEE Transactions on Industrial Electronics* 65.2 (2017), pp. 1539–1548.

[104] Yanxue Wang, Zexian Wei and Jianwei Yang. “Feature trend extraction and adaptive density peaks search for intelligent fault diagnosis of machines”. In: *IEEE Transactions on Industrial Informatics* 15.1 (2018), pp. 105–115.

[105] Ron Kohavi and George H John. “Wrappers for feature subset selection”. In: *Artificial intelligence* 97.1-2 (1997), pp. 273–324.

[106] Zarita Zainuddin, Kee Huong Lai and Pauline Ong. “An enhanced harmony search based algorithm for feature selection: Applications in epileptic seizure detection and prediction”. In: *Computers & Electrical Engineering* 53 (2016), pp. 143–162.

[107] Jerome H Friedman. “Greedy function approximation: a gradient boosting machine”. In: *Annals of statistics* (2001), pp. 1189–1232.

[108] Weichao Luo et al. “A hybrid predictive maintenance approach for CNC machine tool driven by Digital Twin”. In: *Robotics and Computer-Integrated Manufacturing* 65 (2020), p. 101974.

[109] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[110] Jesper E Van Engelen and Holger H Hoos. “A survey on semi-supervised learning”. In: *Machine Learning* 109.2 (2020), pp. 373–440.

[111] Thurston Sexton et al. “Hybrid datafication of maintenance logs from ai-assisted human tags”. In: *2017 ieee international conference on big data (big data)*. IEEE. 2017, pp. 1769–1777.

[112] Gian Antonio Susto et al. “Prediction of integral type failures in semiconductor manufacturing through classification methods”. In: *2013 IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA)*. IEEE. 2013, pp. 1–4.

[113] Leo Breiman. “Random forests”. In: *Machine learning* 45.1 (2001), pp. 5–32.

[114] Krantiraditya Dhalmahapatra et al. “Decision support system for safety improvement: An approach using multiple correspondence analysis, t-SNE algorithm and K-means clustering”. In: *Computers & Industrial Engineering* 128 (2019), pp. 277–289.

[115] Christos Boutsidis et al. “Randomized dimensionality reduction for k-means clustering”. In: *IEEE Transactions on Information Theory* 61.2 (2014), pp. 1045–1062.

[116] Hashem M Hashemian. "State-of-the-art predictive maintenance techniques". In: *IEEE Transactions on Instrumentation and measurement* 60.1 (2010), pp. 226–236.

[117] Sailendu Biswal and GR Sabareesh. "Design and development of a wind turbine test rig for condition monitoring studies". In: *2015 International Conference on Industrial Instrumentation and Control (ICIC)*. IEEE. 2015, pp. 891–896.

[118] Symone Gomes Soares and Rui Araújo. "An on-line weighted ensemble of regressor models to handle concept drifts". In: *Engineering Applications of Artificial Intelligence* 37 (2015), pp. 392–406.

[119] Jong-Ho Shin, Hong-Bae Jun and Jae-Gon Kim. "Dynamic control of intelligent parking guidance using neural network predictive control". In: *Computers & Industrial Engineering* 120 (2018), pp. 15–30.

[120] Nesime Tatbul et al. "Precision and recall for time series". In: *arXiv preprint arXiv:1803.03639* (2018).

[121] Tianfeng Chai and Roland R Draxler. "Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature". In: *Geoscientific model development* 7.3 (2014), pp. 1247–1250.

[122] Julio-Omar Palacio-Niño and Fernando Berzal. "Evaluation metrics for unsupervised learning algorithms". In: *arXiv preprint arXiv:1905.05667* (2019).

[123] *Grafana*. https://grafana.com/. Accessed: 2021-03-04.

[124] *Power BI*. https://powerbi.microsoft.com/en-us/. Accessed: 2021-03-04.

[125] *Apache Spark™ - Unified Analytics Engine for Big Data*. https://spark.apache.org/. Accessed: 2021-01-16.

[126] *MySQL*. https://www.mysql.com/products/workbench/. Accessed: 2021-03-04.

[127] *Open platform communication foundation.opc-ua 12-parts specifications*. https://opcfoundation.org/developer-tools/specifications-unified-architecture. Accessed: 2020-12-27.

[128] *Pump Manual Documentation*. https://manualzz.com/doc/19881138/diaphragm-liquid-pump-nf-1.25-operating-and-installation. Accessed: 2020-11-15.

[129] Kristoffer McKee et al. "A review of major centrifugal pump failure modes with application to the water supply and sewerage industries". In: *ICOMS Asset Management Conference Proceedings*. Asset Management Council. 2011.

[130] Michael Volk. *Pump characteristics and applications*. CRC Press, 2013.

[131] Jason Mais. "Spectrum analysis: The key features of analyzing spectra". In: *SKF USA, Inc* (2002).

[132] *Wireshark, network protocol analyzer*. https://www.wireshark.org/. Accessed: 2020-11-05.

[133] *mongoDB database*. https://www.mongodb.com/2. Accessed: 2020-12-14.

[134] MN Noor et al. *Filling missing data using interpolation methods: Study on the effect of fitting distribution*. Vol. 594. Trans Tech Publ, 2014.

[135] Stefan-Helmut Leitner and Wolfgang Mahnke. "OPC UA–service-oriented architecture for industrial applications". In: *ABB Corporate Research Center* 48 (2006), pp. 61–66.

[136] *Parquet format file*. https://parquet.apache.org/. Accessed: 2021-05-01.

[137] Todor Ivanov and Matteo Pergolesi. "The impact of columnar file formats on SQL-on-hadoop engine performance: A study on ORC and Parquet". In: *Concurrency and Computation: Practice and Experience* 32.5 (2020), e5523.

[138] *Resilient Distributed Dataset (RDD)*. https://databricks.com/glossary/what-is-rdd. Accessed: 2021-02-05.

[139] *Jupyter notebook*. https://jupyter.org/. Accessed: 2021-01-10.

[140] Christophe Leys et al. "Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median". In: *Journal of experimental social psychology* 49.4 (2013), pp. 764–766.

[141] A Rogier T Donders et al. "A gentle introduction to imputation of missing values". In: *Journal of clinical epidemiology* 59.10 (2006), pp. 1087–1091.

[142] Kamakshi Lakshminarayan, Steven A Harp and Tariq Samad. "Imputation of missing data in industrial databases". In: *Applied intelligence* 11.3 (1999), pp. 259–275.

[143] CD Jayaweera and N Aziz. "Reliability of Principal Component Analysis and Pearson Correlation Coefficient, for Application in Artificial Neural Network Model Development, for Water Treatment Plants". In: *IOP Conference Series: Materials Science and Engineering*. Vol. 458. 1. IOP Publishing. 2018, p. 012076.

[144] *MLlib Documentation*. `https://spark.apache.org/docs/latest/ml-classification-regression.html`. Accessed: 2021-03-01.

[145] *scikit-learn Lib*. `https://scikit-learn.org/stable/`. Accessed: 2021-03-04.