Design and investigation of an IIOT-enable device for remote monitoring and predictive maintenance of hydraulic hammers

Published version
Politecnico di Bari
DOI:10.60576/poliba/iris/heidarpour-farhad_phd2023

Politecnico di Bari

LIBERATORIA PER L'ARCHIVIAZIONE DELLA TESI DI DOTTORATO

Al Magnifico Rettore
del Politecnico di Bari

Il sottoscritto HEIDARPOUR FARHAD nato a CHALOOS (IRN) il 29/09/1990

residente a BARI (BA) in via GIAN GIUSEPPE CARULLI, 74 e-mail farhad.heidarpour@poliba.it

iscritto al 3° anno di Corso di Dottorato di Ricerca in INGEGNERIA MECCANICA E GESTIONALE ciclo 35°

ed essendo stato ammesso a sostenere l'esame finale con la prevista discussione della tesi dal titolo:

DESIGN AND INVESTIGATION OF AN IIOT-ENABLED DEVICE FOR REMOTE MONITORING AND PREDICTIVE MAINTENANCE OF HYDRAULIC HAMMERS.

DICHIARA

1) di essere consapevole che, ai sensi del D.P.R. n. 445 del 28.12.2000, le dichiarazioni mendaci, la falsità negli atti e l'uso di atti falsi sono puniti ai sensi del codice penale e delle Leggi speciali in materia, e che nel caso ricorressero dette ipotesi, decade fin dall'inizio e senza necessità di nessuna formalità dai benefici conseguenti al provvedimento emanato sulla base di tali dichiarazioni;
2) di essere iscritto al Corso di Dottorato di ricerca a in Ingegneria Meccanica e Gestionale ciclo 35°, corso attivato ai sensi del "*Regolamento dei Corsi di Dottorato di ricerca del Politecnico di Bari*", emanato con D.R. n.286 del 01.07.2013;
3) di essere pienamente a conoscenza delle disposizioni contenute nel predetto Regolamento in merito alla procedura di deposito, pubblicazione e autoarchiviazione della tesi di dottorato nell'Archivio Istituzionale ad accesso aperto alla letteratura scientifica;
4) di essere consapevole che attraverso l'autoarchiviazione delle tesi nell'Archivio Istituzionale ad accesso aperto alla letteratura scientifica del Politecnico di Bari (IRIS-POLIBA), l'Ateneo archivierà e renderà consultabile in rete (nel rispetto della Policy di Ateneo di cui al D.R. 642 del 13.11.2015) il testo completo della tesi di dottorato, fatta salva la possibilità di sottoscrizione di apposite licenze per le relative condizioni di utilizzo (di cui al sito http://www.creativecommons.it/Licenze), e fatte salve, altresì, le eventuali esigenze di "embargo", legate a strette considerazioni sulla tutelabilità e sfruttamento industriale/commerciale dei contenuti della tesi, da rappresentarsi mediante compilazione e sottoscrizione del modulo in calce (Richiesta di embargo);
5) che la tesi da depositare in IRIS-POLIBA, in formato digitale (PDF/A) sarà del tutto identica a quelle **consegnate**/inviate/da inviarsi ai componenti della commissione per l'esame finale e a qualsiasi altra copia depositata presso gli Uffici del Politecnico di Bari in forma cartacea o digitale, ovvero a quella da discutere in sede di esame finale, a quella da depositare, a cura dell'Ateneo, presso le Biblioteche Nazionali Centrali di Roma e Firenze e presso tutti gli Uffici competenti per legge al momento del deposito stesso, e che di conseguenza va esclusa qualsiasi responsabilità del Politecnico di Bari per quanto riguarda eventuali errori, imprecisioni o omissioni nei contenuti della tesi;
6) che il contenuto e l'organizzazione della tesi è opera originale realizzata dal sottoscritto e non compromette in alcun modo i diritti di terzi, ivi compresi quelli relativi alla sicurezza dei dati personali; che pertanto il Politecnico di Bari ed i suoi funzionari sono in ogni caso esenti da responsabilità di qualsivoglia natura: civile, amministrativa e penale e saranno dal sottoscritto tenuti indenni da qualsiasi richiesta o rivendicazione da parte di terzi;
7) che il contenuto della tesi non infrange in alcun modo il diritto d'Autore né gli obblighi connessi alla salvaguardia di diritti morali od economici di altri autori o di altri aventi diritto, sia per testi, immagini, foto, tabelle, o altre parti di cui la tesi è composta.

Luogo e data Bari, 03/03/2023 Firma

Il/La sottoscritto, con l'autoarchiviazione della propria tesi di dottorato nell'Archivio Istituzionale ad accesso aperto del Politecnico di Bari (POLIBA-IRIS), pur mantenendo su di essa tutti i diritti d'autore, morali ed economici, ai sensi della normativa vigente (Legge 633/1941 e ss.mm.ii.),

CONCEDE

- al Politecnico di Bari il permesso di trasferire l'opera su qualsiasi supporto e di convertirla in qualsiasi formato al fine di una corretta conservazione nel tempo. Il Politecnico di Bari garantisce che non verrà effettuata alcuna modifica al contenuto e alla struttura dell'opera.
- al Politecnico di Bari la possibilità di riprodurre l'opera in più di una copia per fini di sicurezza, back-up e conservazione.

Luogo e data Bari, 03/03/2023 Firma

Politecnico di Bari

RICHIESTA DI EMBARGO

Sottoscrivere solo nel caso in cui si intenda auto-archiviare la tesi di dottorato nell'Archivio Istituzionale ad accesso aperto alla letteratura scientifica POLIBA-IRIS (https://iris.poliba.it)) non in modalità "Accesso Aperto", per motivi di segretezza e/o di proprietà dei risultati e/o informazioni sensibili o sussistano motivi di segretezza e/o di proprietà dei risultati e informazioni di Enti esterni o Aziende private che hanno partecipato alla realizzazione della ricerca.

Il sottoscritto HEIDARPOUR FARHAD nato a CHALOOS (IRN)

Il 29/09/1990 residente BARI (BA) alla via GIAN GIUSEPPE CARULLI, 74 indirizzo e-mail farhad.heidarpour@poliba.it

iscritto/a al corso di dottorato di ricerca INGEGNERIA MECCANICA E GESTIONALE ciclo35°

Autore della tesi di dottorato dal titolo: DESIGN AND INVESTIGATION OF AN IIOT-ENABLED DEVICE FOR REMOTE MONITORING AND PREDICTIVE MAINTENANCE OF HYDRAULIC HAMMERS. (Hammer4.0)

e ammesso a sostenere l'esame finale:

NON AUTORIZZA

Il Politecnico di Bari a pubblicare nell'Archivio Istituzionale di Ateneo ad accesso aperto il testo completo della tesi depositata per un periodo comunque non superiore a 12 (dodici) mesi decorrenti dalla data di esame finale.

Specificare la motivazione *(apporre una crocetta sulla motivazione):*

☐ Brevetto
(*indicare nel campo libero la data della domanda di deposito*)__________________________________

☒ Segreto industriale, se è stato firmato un accordo di non divulgazione.
(indicare nel campo libero gli estremi dell'accordo) Pon Industriale

☐ Segreto d'ufficio a tutela di progetti ____________________________________

☐ Motivi di priorità nella ricerca (previo accordo con terze parti) ________________________

☐ Motivi editoriali ____

☐ Altro (*specificare*) ____

Saranno comunque consultabili ad accesso aperto i dati bibliografici e l'abstract.

Il sottoscritto dottorando dichiara in virtù di quanto sopra che si rende opportuno procrastinare la pubblicazione della tesi attraverso l'Archivio Istituzionale ad accesso aperto (POLIBA -IRIS) e di impostare la data di embargo, in fase di deposito della tesi di dottorato in formato elettronico (pdf/A) per un periodo *di embargo* non superiore a 12 (dodici) mesi decorrenti dalla data di esame finale.

Il sottoscritto dottorando dichiara di essere a conoscenza che a scadenza della data di embargo su riportata, la tesi verrà pubblicata attraverso l'Archivio Istituzionale ad accesso aperto alla letteratura scientifica del Politecnico di Bari.

Luogo e data Bari, 03/03/2023

Firma Dottorando

Firma Relatore ____________________________

Department of Mechanics, Mathematics and Management
MECHANICAL AND MANAGEMENT ENGINEERING
Ph.D. Program
SSD: ING-IND/15–Design methods
for Industrial Engineering

Final Dissertation

DESIGN AND INVESTIGATION OF AN IIOT-ENABLED DEVICE FOR REMOTE MONITORING AND PREDICTIVE MAINTENANCE OF HYDRAULIC HAMMERS

---

by

Heidarpour Farhad

Supervisor:

Prof. Uva Antonio Emmanuele

coordinator of Ph.D. Program:

Prof. Demelio Giuseppe P.

Course n°35, 04/12/2019-03/03/2023

Department of Mechanics, Mathematics and Management
MECHANICAL AND MANAGEMENT ENGINEERING
Ph.D. Program
SSD: ING-IND/15–Design methods
for Industrial Engineering

Final Dissertation

DESIGN AND INVESTIGATION OF AN IIOT-ENABLED DEVICE FOR REMOTE MONITORING AND PREDICTIVE MAINTENANCE OF HYDRAULIC HAMMERS

---

by

Heidarpour Farhad

Referees:

Prof. Serackis Artūras

Prof. Gerbino Salvatore

Supervisor:

Prof. Uva Antonio Emmanuele

Coordinator of Ph.D Program:

Prof. Demelio Giuseppe P.

Course n°35, 04/12/2019-03/03/2023

Abstract

Hydraulic hammers, also known as breakers and peckers, are utilized in a wide variety of applications to demolish a structure and break rocks into smaller sizes. These tools and equipment are extremely sensitive and operate in harsh environments. As a result, there is a widespread requirement for remote control and monitoring of equipment and machines. In addition, given the technological advances in sensors, data transmission, and data collection via the Internet of Things, as well as the high demand for data analytics and the importance of maintenance in the fourth industrial revolution, artificial intelligence is being used as a powerful tool.

Thus, remote monitoring of industrial equipment such as hydraulic hammers has become a critical feature of Industry 4.0 and Internet of Things technologies. Data collection has also recently received a lot of attention to improve machines' ability to make future decisions based on the collected data and increase efficiency. However, a major challenge is to ensure the lifetime of equipment and machines and reduce the time and cost of maintenance, which directly affects the cost and competitiveness of the product. Therefore, machine learning, deep learning, and predictive maintenance models have become important. The first part of this study (INDECONNECT® project) involves presenting the design and development of an Internet of Things device, specifically a data logger, that aims to enhance the performance of hydraulic breakers through remote monitoring.

The device is equipped with sensors for data collection, analysis, and management. By designing the platform and strategically placing sensors, the device is expected to obtain vast amounts of data (Big Data) regarding various

aspects of the hydraulic hammer such as vibration, machine operation time, oil pressure, temperature, and oil flow, based on the operation conditions and type of material used. Analyzing the large amount of data collected by the Data logger directly from the hydraulic hammer during its operation can provide valuable information for adjusting process planning, implementing predictive maintenance, and establishing standard technical information for different modes of the Hydraulic hammer.

Secondly, the project seeks to predict maintenance operations by utilizing artificial intelligence tools, particularly machine learning and deep learning methods, based on a dataset of various components at different time periods. In this study, we utilized machine learning and deep learning algorithms to predict machine and component failures for two different time periods - one day and seven days in the future. Nevertheless, as maintenance prediction datasets tend to be unbalanced, we employed two approaches in this study - a weighted average coefficient and a two-step method - to predict the probability of long-term failures. The two-step method is a novel technique that significantly reduces the data set imbalance and enhances the performance of neural network algorithms. The outcomes indicate that the Convolutional Neural Network is the most effective in predicting the likelihood of machines and components failing.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Objective and Research Question

The production and manufacturing of goods and tools date back to the earliest days of human history. However, with the advent of the first industrial revolution (Industry 1.0) in the 18th century, a new factory system emerged, powered by steam, leading to large-scale industries. During the second industrial revolution (Industry 2.0) in the 19th century, electricity was utilized to develop innovative mechanisms and processes that increased production speed and facilitated mass production. The third industrial revolution (Industry 3.0) saw the widespread use of IT technology, automated production systems, and robots in various industries. The fourth industrial revolution (Industry 4.0), an evolution of the third industrial revolution, incorporates communication and intelligent information technologies, connecting computers and various automations via the Internet and data transmission networks.

The Internet of Things (IoT) and the Industrial Internet of Things (IIoT) are two fundamental concepts and technologies that have become increasingly prevalent in the fourth industrial revolution. The IoT and IIoT are utilized in various industries to monitor, analyze, and make devices smarter, setting up a platform that manages a range of smart devices. With the implementation of IoT in devices, they can communicate with each other seamlessly and without human intervention. This interconnectedness enables devices to share data and work together in ways that optimize productivity, efficiency, and safety. Since IoT-based applications do not require human intervention, they are used in many industries [1].

In recent years, industrial equipment monitoring and data collection by sensors, data loggers, and operators have attracted much attention. This has improved the ability of machines to make intelligent decisions based on the collected data and has led manufacturers to adopt product service systems (PSS) and the Internet of Things (IoT) [2, 3]. Today, technology and the Internet of Things provide the ability to remotely monitor in the harsh environments and predict equipment maintenance using intelligent algorithms and machine learning [4]. Despite the advancements in Industry 4.0, there still exists a need for standardization and modularity in the field. The distributed computing of the massive amounts of data generated by sensors and interactions with industrial machines also poses a significant challenge [5]. In addition to predicting failures, a major challenge is performing reliable and error-free maintenance operations and validating fully functional equipment in a continuous and timely manner. To this end, significant efforts have been made to develop and test real-time technical service systems and software for mobile apps to avoid undesirable faults and problems [6]. In [7], innovative forms of virtual reality (AR) are combined to improve not only maintenance but also training of PSS types. In [8], an approach to machine tool management is presented that combines sensors, a timing module, and engineers to control the department store in real-time.

Nowadays, advances in transportation technology also lead to global competition among companies worldwide to produce a product with high quality and a low price. However, maintenance of production equipment and machinery reduces productivity (machine downtime and machine calibration) and directly affects the cost of the final production of the product [9, 10]. About one-third of equipment maintenance expenditures in the United States are unnecessary and only drive-up costs [11]. Thus, maintenance directly affects human resources and material consumption and is a major concern of the fourth industrial revolution [12].

Therefore, artificial intelligence tools, particularly machine learning and deep learning, due to their great potential in creating automatic models for Big Data analysis, allow us to reduce repair and maintenance costs, maximize components' working lives, reduce machine downtime, increase performance and operational safety, and improve decision-making capabilities regarding the ideal timing and actions for machine maintenance [13, 14, 15, 16, 17].

1.2 Contribution

The aim of this work is to propose and develop, as a first step, an innovative device (data logger) to improve monitoring, safety and performance. It is attached to the hydraulic breakers and reports detailed information about the operation in real time or near real time to the customer and the technical department. The data logger provides real-time data on operating hours, maintenance intervals, oil pressure, flow, temperature, vibration, and equipment location. The second step is to analyze a set of data obtained from various sources available online [18]. First, 10 of the most popular machine learning algorithms used in various predictive maintenance (PdM) works were applied. Machine learning algorithms used:

- Random Forest Classifier (RFC);
- eXtreme Gradient Boosting Classifier (XGB Classifier);
- Logistic Regression (LR);
- Extra Trees Classifier;
- Bagging Classifier;
- Support Vector Classifier (SVC);
- Linear Support Vector Classifier (Linear SVC);
- Stacking Classifier;
- Adaptive Boosting Classifier (AdaBoost);
- Decision Tree Classifier.

Second, an innovative Deep Learning method was applied to the dataset. In this work, two types of Deep Learning algorithms are used:

- Convolutional Neural Networks (CNN)
- Long Short-Term Memory networks (LSTM)

The results of the Machine Learning algorithms and Deep Learning models were compared and discussed over the next 24 hours. Considering that the maintenance and repair of industrial equipment is a time-consuming process and, in many cases, requires the replacement of parts and the purchase and supply of spare parts and industrial equipment may be located in remote areas (e.g., road construction machinery, hydraulic breakers, etc.) [19]. Therefore, we improved PdM models with traditional learning machines and Deep Learning algorithms for up to 7 days and compared the results.

1.3 Part Outline

This thesis is composed by four chapters. The first and current Chapter 1 provides an introduction about the reference context. The following Chapter 2 describes the state of the art of Industry 4.0, its challenges and technologies, focusing on remote monitoring, data acquisition, design requirements and standards, sensors and microprocessors, sensor placement, cybersecurity and data transmission. Chapter 3 presents artificial intelligence algorithms and datasets, the challenges of maintaining predictive datasets, data preparation, data analysis, data visualization, and the study of the structure of machine learning and deep learning algorithms. To better explain the results for each component and machine, short-term (24-hour) and long-term (7-day) maintenance predictions are presented. The studies and contributions reported in this thesis are always compared with the results presented in various articles on this data set. Chapter 4 serves as a concluding chapter, summarizing the research findings and suggesting possible areas for future research.

Chapter 2

IIoT-enabled device for remote monitoring

2.1 State of the Art

This chapter describes the background and literature on an IIoT device (data logger), challenges, and Industry 4.0 technologies with a focus on remote monitoring of hydraulic hammers.

2.1.1 Introduction

The fourth industrial revolution has affected all aspects of industry by changing attitudes toward design, requirements, production and manufacturing, and even delivery and payment. Unlike previous revolutions, the fourth industrial revolution was predictable [5]. Therefore, companies, institutes, and researchers were given the opportunity to actively shape the future. The concept of the fourth industrial revolution was first introduced in Germany as a leading industrial and technology sector to create a fully integrated industry in 2011 [20]. In contrast, Italy presented a national plan for Industry 4.0 about four years later. One of the most important factors of the fourth industrial revolution is the change in technology and production processes, which has created new employment and investment opportunities. In Figure 2-1, you can see nine different areas of the fourth industrial revolution, which can be divided into four main parts: Interconnectivity, Automation, Machine Learning, and Real-Time Data.

The fourth industrial revolution is characterized by the central role played by Internet of Things (IoT) and Industrial Internet of Things (IIoT) devices. These devices facilitate communication and interaction between objects and things through sensors and smartphones, with the ultimate goal of achieving the Internet of Services (IoS). By providing companies with the ability to offer their products over the Internet, IoS enables them to offer services and enhance their overall value proposition [21]. The Industrial Internet of Things uses a range of sensors, actuators, processors, GSM modules, etc. to improve the manufacturing process. By using smart machines, the data that traditional industrial machines have been generating for years in the industry can be analyzed in real-time or near real-time. The philosophy of IIoT is that smart machines and industrial equipment are not only better than humans at storing and collecting data in an integrated way but are also better at analyzing and transmitting important information that aids in improving technical business decisions.

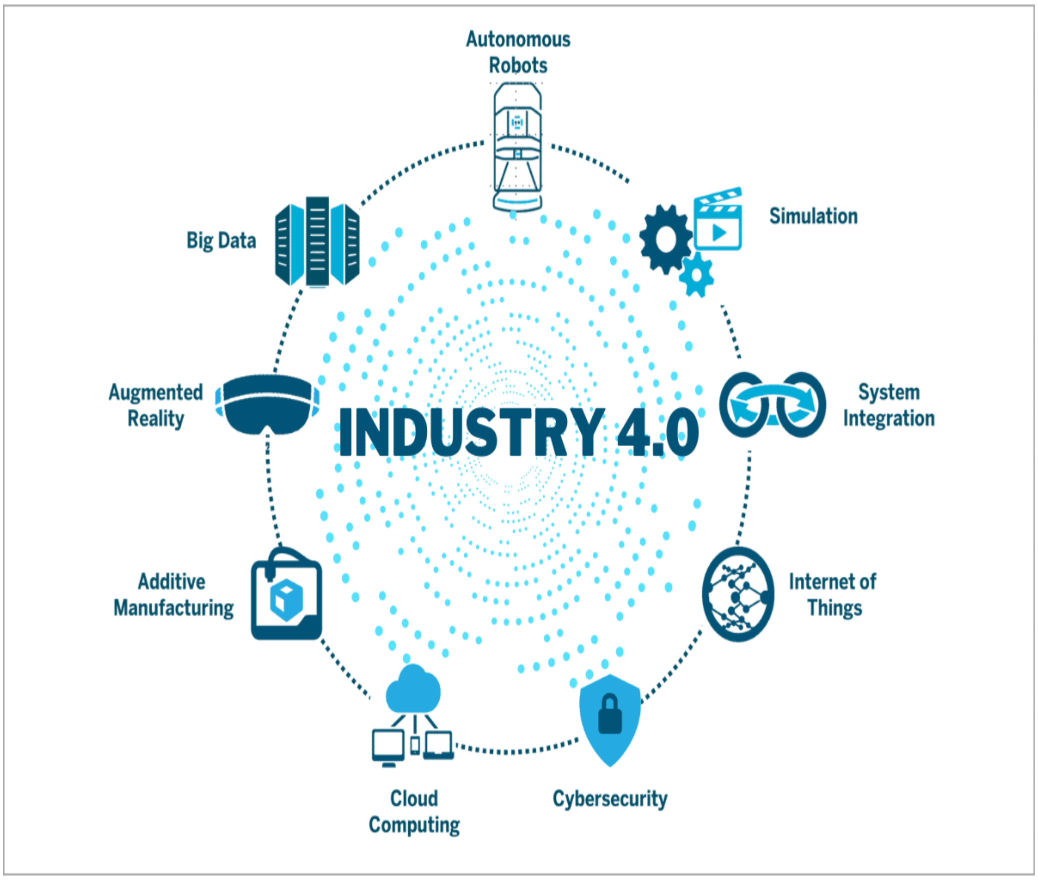


Fig 2- 1. The Industry 4.0 enabling technologies [22].

The utilization of connected sensors and actuators can assist companies in detecting inefficiencies and issues early on, resulting in time and cost savings and supporting business intelligence initiatives. The Industrial Internet of Things (IIoT) is particularly promising for enhancing quality control, implementing sustainable and eco-friendly practices, ensuring supply chain traceability, and improving overall manufacturing supply chain efficiency. Moreover, the IIoT plays a crucial role in various industrial processes, such as predictive maintenance (PdM), improved field service, energy management, and asset tracking.

Generally, the shift in society's demand for cheaper and more technological products and services is a powerful stimulus for improving technologies. The development of connected devices and tooling environments, combined with available data collection and analysis techniques has led manufacturers to incorporate Product Service Systems (PSS) and the Internet of Things (IoT) [23, 24]. Figure 2-2 shows the image and position of the Indeconnect data logger.

Fig 2- 2. Image and position of the INDECONNECT data logger on a hydraulic hammer INDECO HP 600 FS.

This chapter describes a monitoring device (IIoT) that can monitor both real-time and non-real-time measurements. The data logger is mounted to machine tools, which perform the necessary preprocessing to transmit these measurements to a cloud server by sensors. To increase the efficiency of product development, the hydraulic machinery and tooling industry must have reliable and accurate monitoring systems that can be accessed quickly and accurately [25]. For fluid power systems, and especially for oil-hydraulic percussion units, it is crucial to know the temperature, pressure, oil flow, and vibration of the system under different operating conditions [26].

This project will provide an innovative device (data logger) to improve monitoring, safety, and performance. It is attached to the hydraulic hammers and reports detailed information about the operation in real-time or close to real-time to the customer and the engineering department. It will provide real-time data on operating hours, maintenance intervals, pressure, flow and temperature of the oil, vibrations, and the location of the equipment.

2.1.2 Challenges

As described earlier, the core and main goal of the fourth industrial revolution is machine intelligence, data acquisition, data analysis, and object-to-object and human-to-object interaction to decentralize production and personalize products and services. This project (INDECONNECT) is about making hydraulic hammers intelligent. One of the upcoming challenges is to review the performance and structure of hydraulic hammers in order to identify the sensitive and vulnerable points of these machines. It should be recalled that hydraulic breakers are widely used in dam construction, tunneling, mining and road construction.

The study of the sales data of the company INDECO has shown that these devices are used all over the world from Australia to Mexico and from Northern Europe to South Africa. This issue has presented us with many challenges, such as the different types of rock and soil, ambient temperature and humidity, which directly affect the performance of hydraulic breakers. For example, the ambient temperature affects the density of the oil and thus the pressure and flow rate. The studies carried out in this project on hydraulic hammers are based on the ambient temperature of the city of Bari in August 2021, with an average temperature of 29 degrees Celsius and humidity of 87%, based on the results of the meteorological terminal of Bari Karol Wojtyla airport station. Moreover, the results are considered with a safety factor of 20% to determine the requirements.

After reviewing [27, 28, 29, 30, 31], it is impossible to verify the performance of hydro percussion devices without performing a certain number of tests on samples. In fact, for any research work on hydraulic hammers, we need to conduct experimental tests to verify the data and performance results of the set. In [32, 33], it was reported that the first challenge and the reason for the incomplete description of the conditions of laboratory tests is the lack of accurate knowledge of the production conditions of hydraulic hammer components, possible leakage and drag forces.

In this project, the INDECO HP 3000 FS (Figure 2-3) model was studied to determine design requirements such as ambient and oil operating temperature, oil pressure, vibration, and oil flow. Based on these design requirements, sensors, microprocessors, batteries, communication modules, and device box standards were investigated. It should be noted that all design requirements and claims were evaluated at most operating pressures.

Fig 2- 3. INDECO HP 3000 FS .

The final challenge is data transmission over the Internet and cloud storage. As mentioned earlier, hydraulic breakers are typically deployed in harsh environments far from residential areas, so the quality of Internet coverage in these areas is weak or, in many cases, there is no Internet access at all. Internet standards also vary from country to country. In Mexico, for example, there is only 2g and 3g coverage in urban and non-urban areas, while in European countries most areas have 3G and 4g Internet coverage. In General, A remote monitoring device for a hydraulic hammer should have the following design requirements:

- **Real-time monitoring:** The device should be designed to provide real-time monitoring of key operating parameters such as hydraulic pressure, temperature, impact frequency, and energy output.

- **Wireless connectivity:** The device should be designed to connect wirelessly to a monitoring system, allowing operators to view performance data in real-time from a remote location.
- **Robustness:** The device should be designed to withstand the harsh operating conditions of a hydraulic hammer, including vibration, shock loads, and exposure to dust and debris.
- **Power efficiency:** The device should be designed to consume minimal power, ensuring long battery life and reducing the need for frequent recharging.
- **Data storage:** The device should be designed to store data locally, allowing for offline analysis and troubleshooting.
- **Compatibility:** The device should be designed to be compatible with a wide range of hydraulic hammers, regardless of manufacturer or model.
- **User interface:** The device should be designed with a user-friendly interface that provides easy access to data and allows operators to configure settings and alerts.

Overall, a remote monitoring device for a hydraulic hammer should provide operators with the information they need to optimize performance, reduce downtime, and increase productivity, while minimizing the need for onsite inspections and maintenance.

2.2 Hydraulic Hammer Structure

A series of experimental measurements will be performed to evaluate how the sensors' data can predict the hydraulic hammer's behavior. The hydraulic hammer that will be used for this work is an INDECO HP series. (See Fig 2-1.) This hydraulic hammer is of medium and big size and all its components except the back-head, which is made of ductile cast iron, are made of steel. This hydraulic hammer has a maximum power between 160 and 200 bars and the oil flow rate is between 1340-600 l/min, depending on operating conditions. The main components of the breaker are shown in Figure 2-4.

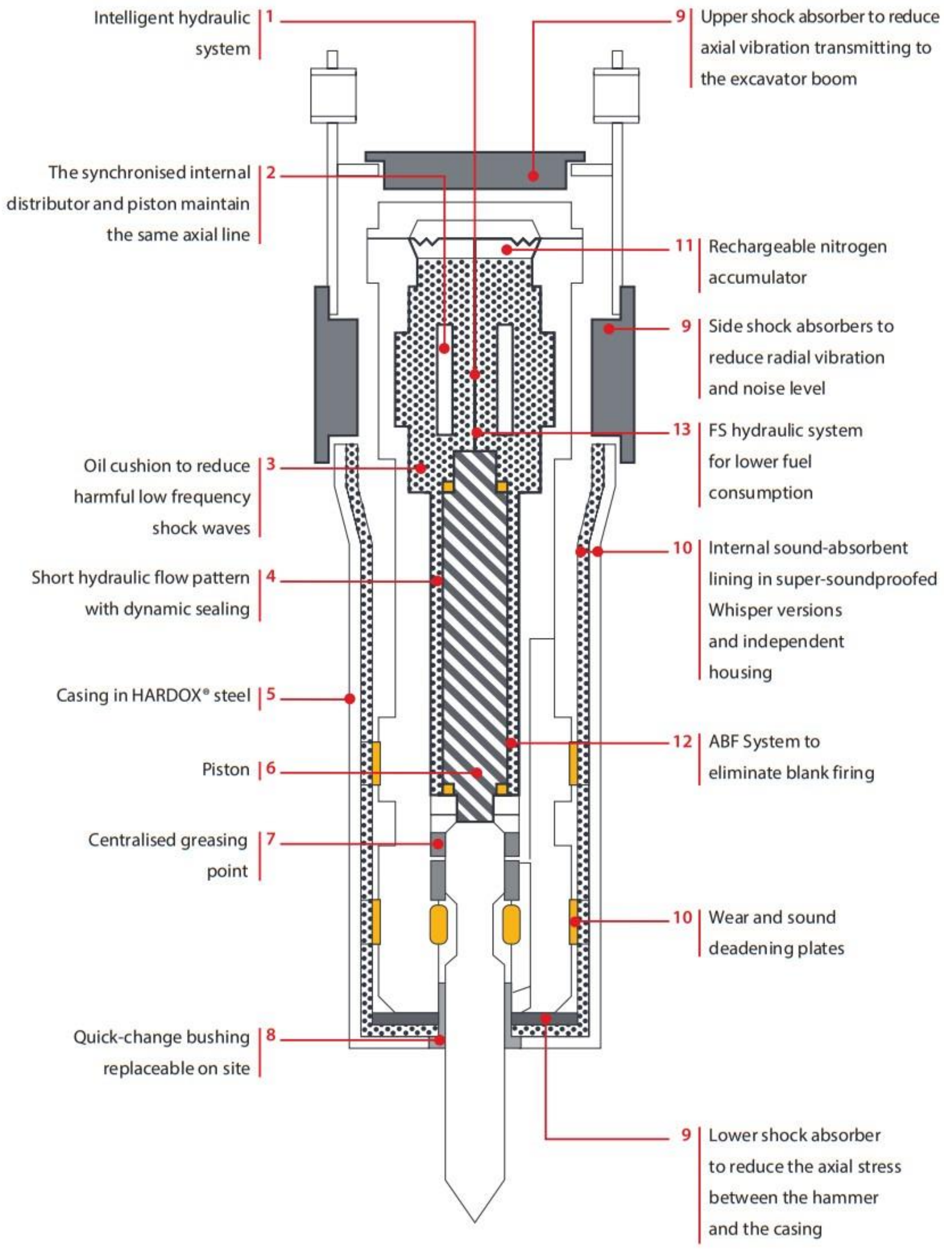


Fig 2- 4. The hydraulic hammer is used in the experiments, showing a sectioned view at a center plane.

Indeco hydraulic hammers are an outstanding expression of Italian high-tech and construction quality applied to demolition. In-depth research into hydraulic systems, materials, heat treatment and accessories have enabled Indeco to establish a reputation on markets throughout the world for product excellence. With its many different models, divided into large, medium and small and available in various versions, Indeco has the widest range of hammers available anywhere in the world. This provides end-users with a huge choice, ensuring that they can find the ideal hammer/excavator match. All Indeco hammers have a special intelligent hydraulic system |1|, enabling them to automatically vary the energy and frequency of the blows according to the hardness of the material being demolished.

This optimizes the hydraulic pressure delivered by the machine, thus improving productivity and enhancing the overall performance. Exclusive features such as the synchronized internal distributor |2| aligned with the piston, the oil cushions |3| for vibration dampening and the short hydraulic flow pattern |4| make it possible to completely do away with seals in the distribution area, a decisive factor in extending the working life of the hammer and significantly reducing downtimes. The use of special low-alloy steels, exclusively manufactured according to Indeco's own formula greatly lengthen the average working life of the major hammer components. The housing |5| is made out of extra-strength HARDOX® steel wear plates, which eliminate buckling.

The piston |6| is divided into two parts, for greater impact energy and lower operating costs. The centralized greasing system |7| enables the sliding parts to remain lubricated even when the hammer is operating horizontally, thus considerably reducing wear and tear on components and extending product lifetime. The “quick change” interchangeable bushing |8| is available in various materials for different jobs; it is inserted into the lower tool bushing where the tool moves, and reduces maintenance times and costs, by cutting out the long machine downtimes needed to replace the traditional fixed bushing.

All carriers which mount Indeco hammers benefit from the Indeco dual shock-absorption system |9|: an internal hydraulic one and a mechanical one, located outside the body, which substantially reduce the vibrations transmitted to the excavator. The excavator boom is also subject to lower stress levels, as Indeco hammers are considerably lighter under working conditions than rival makes in the same class. Alongside the standard versions there is also a super-soundproofed Whisper version, whose body is lined internally with sound-absorbent material |10| and an “anti-rumble” paint, which – combined with a few modifications to the bushing – enable noise emission levels to be considerably reduced. By lowering pressure peaks, the rechargeable hydraulic/nitrogen accumulator |11| also reduces stress in the excavator hydraulic circuit, keeps the gas charge and energy per blow constant, and reduces maintenance and operating costs.

The ABF (Anti Blank Firing) system |12|, installed as standard on all of the mediumand large-range Indeco hammers, cuts out blank fire by eliminating any down pressure from the hammer whenever the tool is not resting firmly on the surface to be demolished. This increases the service life of all components subject to wear and tear, as well as reducing stress to the hammer body and excavator arm. As well as being efficient and reliable, Indeco hydraulic hammers are now proving to be even more environmentally-friendly and low on fuel consumption. With a now even more efficient hydraulic system |13|, the HP series has now also become FS (Fuel Saving).

The crucial component of a hydraulic hammer is the piston, which is driven by the hydraulic system. The hydraulic system generates high-pressure oil flow that is directed to the back head of the hammer, where it pushes the piston forward, creating a powerful impact force. Other important components of a hydraulic hammer include the front head, which houses the chisel or working tool, and the housing, which contains the hydraulic system and supports the hammer during operation. The front head and the chisel are designed to withstand the high impact forces and vibrations generated during operation.

The hydraulic system itself is also a crucial component, consisting of a hydraulic pump, control valves, and hoses that direct the flow of oil to the back head of the hammer. The design and efficiency of the hydraulic system play a significant role in the performance and effectiveness of the hydraulic hammer.

Overall, the successful operation of a hydraulic hammer depends on the effective integration and interaction of all these components, each of which plays a critical role in delivering the required impact force and achieving the desired results.

2.3 Technical and Design Requirements

The technical and design requirements serve as a crucial foundation for the development and implementation of a product or system. They provide a comprehensive outline of what needs to be constructed and the manner in which it should be constructed.

Technical requirements specify the technical aspects of the product or system. These include things like software or hardware components, programming languages, data structures, security requirements, and performance criteria. Technical requirements ensure that the product or system is reliable, efficient, and safe.

Design requirements, on the other hand, specify how the product or system should look and function from the user's perspective. This includes elements such as the user interface, navigation, and user experience. Design requirements ensure that the product or system is user-friendly, intuitive, and aesthetically pleasing.

Technical and design requirements are crucial for ensuring that the end product or system satisfies the needs and expectations of users. The absence of such requirements may result in misunderstandings, delays, and increased expenses for the development team. Furthermore, these requirements play a significant role in making sure that the product or system can be easily scaled, maintained, and adjusted to accommodate evolving user needs and advancements in technology.

Therefore, A test rig is an essential tool for evaluating the performance and functionality of products and systems, including hydraulic hammers. The importance of a test rig can be summarized as follows:

- **Performance testing:** A test rig allows for accurate and repeatable testing of the hydraulic hammer's performance, including impact frequency, energy output, and hydraulic pressure. This helps to ensure that the hammer meets design specifications and customer requirements.
- **Quality control:** A test rig is used to verify the quality of the hydraulic hammer and identify any potential defects or weaknesses in the design or manufacturing process. This helps to ensure that the product is safe and reliable.
- **Design optimization:** By using a test rig, manufacturers can evaluate the impact of design changes on the performance of the hydraulic hammer. This helps to optimize the design and improve performance, while reducing the time and cost associated with physical prototyping.
- **Cost savings:** A test rig can help to identify potential problems early in the design or manufacturing process, reducing the risk of costly recalls or warranty claims.
- **Customer satisfaction:** A test rig helps to ensure that the hydraulic hammer meets customer requirements and is reliable and safe to use. This can lead to increased customer satisfaction and loyalty.

Overall, a test rig is an essential tool for manufacturers of hydraulic hammers, helping to ensure quality, optimize design, reduce costs, and improve customer satisfaction.

The first stage of the study involves examining several parameters of the hydraulic hammer, including its vibration, oil flow, oil temperature, and oil pressure in the cylinder, during idle, standard, and high operations. The experiments are carried out using an in-house test rig located in the Research and Development laboratory of Indeco Ind. SpA company. Due to Indco's corporate policy, it is not possible to depict the in-house test stand and measuring equipment. But Figure 5-2 shows the experimental measurement setup schematically.

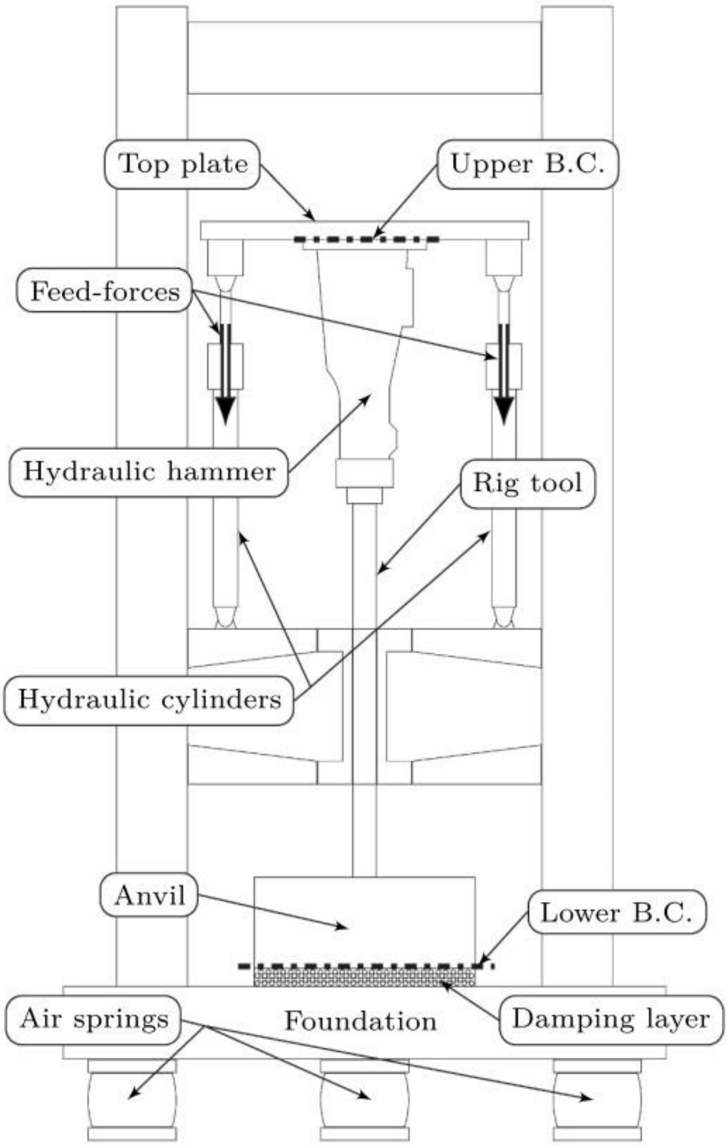


Fig2- 5. Schematics of the experimental measurement set-up.

To ensure the accuracy of the results, the hydraulic hammer is used in a laboratory environment, and a steel test rig is used to mount the hammer. To minimize any errors that may arise from the interaction between the hammer and the ground, a vibration-damping system is employed to isolate the hammer from the ground. The hydraulic hammer is mounted on the steel block using a rigging tool known as an anvil. In engineering and construction, it is common to use damping layers to reduce the transmission of stress waves and vibrations. When an object is subjected to a sudden impact or vibration, stress waves travel through the object, causing it to vibrate and potentially leading to damage or failure.

In the case of an anvil, a damping layer can be placed between the anvil and its resting surface to absorb the stress waves and reduce the amount of vibration transmitted to the surface below. This can help to reduce noise, prevent damage to the surface, and improve the overall stability and longevity of the anvil.

Common materials used as damping layers include rubber, cork, and other resilient materials that can absorb the shock and dissipate the energy of the stress wave. The thickness and composition of the damping layer will depend on the specific application and the level of vibration that needs to be dampened. In this study, a damping layer thickness of 12 mm is used for hydraulic hammers with 200 bar power. This thickness provided adequate protection to the testing surface and absorb enough energy to reduce the amount of vibration transmitted.

In order to study the various parameters of hydraulic hammers and select appropriate sensors, it is important to minimize ambient vibrations that could interfere with the measurements. In this framework, a layered damping approach is used along with supporting steel balls that takes into account the dimensions and weight of both the hammer and the test rig.

A force of 24 kN is applied to the top plate of the frame via the hydraulic system to ensure that the hydraulic hammer remains in a constant position during impacts. This helps to reduce the possibility of measurement deviations caused by the movement of the hammer during operation.

To measure pressure and oil flow during operation, high pressure and oil flow sensors are mounted directly on the structure. Placing the sensors in this manner helps to maintain operating conditions with the least possible error. Since the highest pressure in hydraulic breakers occurs in *a cavity in the rear of the head*, the sensors are placed in this location to obtain the most accurate pressure measurements. A number of different sensors were mounted on the hydraulic hammer to register its characteristic behavior, see Table 2-1. The position of each sensor is indicated by a number. Figure 2-4 shows the position of each number in the hydraulic hammer.

Table 2- 1 Signals and sensors on the hammer.

Position	Type	Sensor
Piston (6)	Position	Laser
Housing (5)	Position	Laser
Housing (5)	Acceleration	Accelerometer
Housing (5)	Stress	Strain gauge
Control Valve (1)	Position	Capacitive
Tool	Stress	Strain gauge
Inlet	Oil pressure	Pressure
Outlet	Oil pressure	Pressure
Intelligent Hydraulic system (1)	Oil pressure	Pressure
synchronized internal distributor (2)	Oil pressure	Pressure

The requirements for the design and selection of sensors and other components for the operation of hydraulic hammers in harsh conditions have been evaluated. Based on the studies, the requirements for the design and selection of sensors, batteries, and other factors that affect the manufacturing of the remote monitoring device are identified and are listed below:

- Working pressure: P: 0 ÷ 100; 0 ÷ 250; 0 ÷ 350 bar.
- Temperature: T: -20 ÷ 120 ° C.
- Flow detection system: 0-200 l / min.
- GSM data transmission system or other similar technologies (3G / 4G).
- GPS position detection system.
- Impact and vibration resistance: 10-100 m / s^2.
- Operating temperature Tmax: 50-100 ° C.
- IEC 60068-2-6; IEC 60068-2-27; EN837.
- External mounting resistance class min IP66.
- Acceleration Range: ±200 g.
- Vibrating proof casing.

2.4 Conceptual Design and System Architecture

Conceptual design and system architecture are two key aspects of the early stages of a technology project, whether it's for software or hardware. While the two concepts are related, they refer to different aspects of the development process.

Conceptual design involves the creation of a high-level vision for the technology. At this stage, the focus is on understanding the problem that the technology is meant to solve and defining the goals of the project. This includes defining the target audience, understanding the user's needs, and identifying the key features and functionalities that the technology should have. The output of the conceptual design phase is usually a concept or a proposal document that outlines the vision and goals for the project.

System architecture, on the other hand, refers to the overall structure and organization of the technology. This includes the design of the different components of the technology, how they interact with each other, and how they work together to achieve the project goals. System architecture involves making decisions on the hardware and software components of the technology, the communication protocols, and the data flows. The output of the system architecture phase is usually a high-level system diagram that shows the different components and how they interact with each other.

Both conceptual design and system architecture are important aspects of the development process, and they are often interconnected. A strong conceptual design will help to guide the system architecture, while a well-designed system architecture will help to bring the conceptual design to life. Together, these two aspects of the development process lay the foundation for a successful technology project.

2.4.1 Proposed System Architecture

The growth of open source software and hardware over the past decade has democratized technology, making it more accessible and affordable to the public. Open source software refers to software whose source code can be modified and shared by anyone, while open source hardware refers to the physical components of a technology that can also be modified and manufactured by anyone. open-source hardware has allowed for the creation of a range of new technologies, from 3D printers to cyber-physicl system (CPS). With the availability of open-source hardware, anyone with the knowledge and resources can now create their own hardware, modify existing designs, or contribute to the development of new technology.

A cyber-physical system (CPS) is a type of system that integrates physical and computational elements to achieve a specific goal. CPSs combine hardware, software, and communication components to control and monitor physical processes, often in real-time. They typically involve the use of sensors, actuators, and other physical devices that interact with the environment to gather data and perform actions.

A key feature of a CPS is its sensing or monitoring part. The demand for remote monitoring and control of industrial processes, equipment, and machinery in a short time will grow rapidly due to the increasing use of technology and Internet access. The remote monitoring system provides data that can be used to optimize production and perform predictive maintenance.

CPSs can be found in a wide range of applications, such as industrial control systems, smart grids, autonomous vehicles, medical devices, and smart homes. They are designed to improve efficiency, reliability, and safety in various domains by automating processes and providing real-time feedback.

There are a variety of data loggers on the market today. A data logger is an electronic device that is used to collect and record data from various sources over time. It can be standalone or part of a larger system, and typically includes sensors or inputs that measure and record parameters such as temperature, humidity, pressure, voltage, current, or other environmental or system variables. The architecture of the proposed system is presented in Figure 2-6.

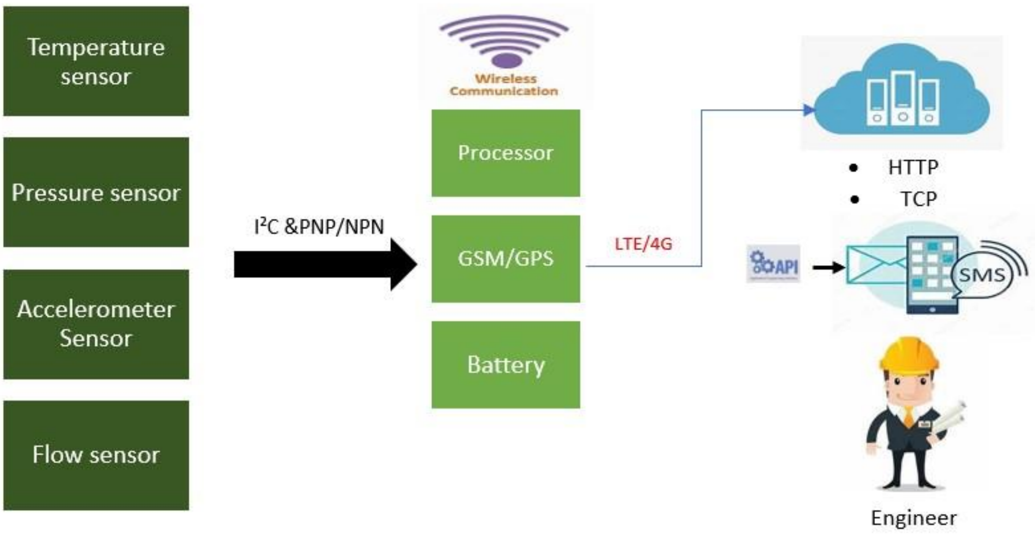


Fig2- 6. System Architecture.

Data loggers can store data locally in their internal memory or external memory devices, and can also transmit the data wirelessly to a computer or other data acquisition system for further analysis or processing. They are commonly used in a wide range of applications, such as monitoring environmental conditions in scientific research, tracking energy usage in buildings, logging flight data in aircraft, or monitoring production processes in industrial settings.

More specifically, the proposed system provides several advantages for monitoring. First, the engineer can observe the current status of each hydraulic breaker almost instantaneously. In addition, the data can be displayed graphically using appropriate programming libraries. For example, Figure 2-7 shows the accelerometer performance of a model HP6000 hydraulic breaker between January 2022 and February 2023. From this graph, it is clear that this device is utilized and used the most in January and February 2022. And this hydraulic breaker was used with very low work intensity in April and November 2022. (According to the accelerometer results, only the hydraulic hammer was moved between April and November).

Fig2- 7. The accelerometer performance diagram for a hydraulic breaker model HP6000.

In terms of predictive maintenance, the algorithms can estimate likely future failures by considering the remaining useful life (RUL) of the hydraulic breaker's major components. Currently, the end-user interface displays the data only in graphical form, but further development of the framework will enable visualization of the data using augmented reality.

2.4.2 System Design and Development

Data loggers play a critical role in IIoT by monitoring and collecting data from various industrial processes, equipment, and systems in real-time. In the case of monitoring hydraulic hammers in harsh environments, a data logger can provide valuable information about the hammer's performance, including impact force, frequency, duration, and energy. This information can be used to optimize the hammer's performance, prevent downtime, and improve safety.

In harsh environments, such as those found in mining, construction, or offshore drilling, data loggers are particularly important as they can operate reliably in extreme temperatures, humidity, and vibrations. A data logger can be designed with a rugged enclosure that can withstand the harsh conditions, ensuring that the device remains operational and continues to collect data.

To monitor hydraulic hammers in harsh environments, the data logger can be connected to sensors that measure various parameters of the hammer's operation. The logger can then collect and store this data, which can be analyzed later to identify patterns or trends that may indicate maintenance or repair needs.

To design a data logger for IIoT (Industrial Internet of Things), you will need a combination of a sensor, a microprocessor, battery, SD memory card, and GSM/GPS module.

2.4.3 Sensor

The selection of the right sensor for a data logger is crucial to the success of the data logging process. The sensor is responsible for measuring the physical or chemical parameter that you are interested in logging, such as temperature, pressure, humidity, or light. Choosing the right sensor is important for several reasons:

- **Accuracy:** The accuracy of the sensor is critical in ensuring that the data collected by the data logger is reliable. A sensor that provides inaccurate readings can lead to incorrect conclusions and bad decisions.
- **Range:** The range of the sensor is also important, as it determines the upper and lower limits of the parameter that can be measured. If the range of the

sensor is too narrow, it may not be able to capture all the variations in the parameter that you are interested in.

- **Sensitivity:** The sensitivity of the sensor is another important factor to consider. A highly sensitive sensor can detect small changes in the parameter being measured, while a less sensitive sensor may not be able to detect subtle changes.
- **Durability:** The durability of the sensor is important, especially in harsh environments. The sensor should be able to withstand extreme temperatures, humidity, and vibrations.
- **Cost:** Finally, the cost of the sensor is also an important consideration. You want to choose a sensor that provides the required accuracy, range, and sensitivity, while still being cost-effective.

When evaluating sensors, it's important to consider factors such as accuracy, resolution, range, linearity, and sensitivity, as well as any calibration or correction factors that may be required. It's also important to consider the physical size and design of the sensor, as well as its power requirements and communication interface. Considering the importance of sensors in the accuracy and final performance of the data logger, a large portion of industrial sensors available on the market were studied. Sensors are evaluated based on performance accuracy and pre-determined design requirements.In accordance with company, marketing, and copyright policies, it is only possible to provide technical details with limitations.

Therefore, three different models were examined for each sensor type, as shown in Table 2-2. An example of these sensors is also shown in Figure 2-8. In general, all of these sensors met all of the design requirements for hydraulic breakers and were mounted with minor modifications to the location of the sensors. The only important factors are the dimensions, price and size of the pressure and flow sensors, which, as mentioned earlier, require that these two sensors be installed directly on a portion of the hydraulic system. It is important to ensure that the sensors are installed in a location that is easily accessible for maintenance and calibration. Additionally, the sensors should be protected from damage and excessive vibration, which can affect their accuracy.

Table 2- 2 The list of sensors under consideration for monitoring hydraulic hammers.

Sensor ID	Brand	Temperature	pressure	Flow	Acceleration	Out Put	Signal	size
CAT34TS02 VP2GT4C	ON Semiconductor	-25 ℃ ~ 125 ℃				I^2C /SMBus	Digital	2 x 3 x 0.75 mm
LM95010CIMM NOPB	Texas Instruments	-25 ℃ ~ 125 ℃				I^2C / SMBus SensorPath	Digital	12 x 21 x 2 mm
ADT7408CCPZ-REEL7	Analog Devices Inc.	-25 ℃ ~ 125 ℃				I^2C / SMBus	Digital, local	3 × 3 ×1 mm
ADXL372BCCZ-RL7TR-ND	Analog Devices Inc.	-40 ℃ ~ 105 ℃			±200 g	SPI/I^2C	Digital	3 × 3.25 × 1.06 mm
3038-0200	TE Connectivity Measurement Specialties	-55 ℃ ~ 125 ℃			±200 g	Analog voltage	Analog	7.5 × 13 × 3 mm
ADXL375BCCZ-RL	Analog Devices Inc.	-55 ℃ ~ 125 ℃			±200 g	SPI/I^2C	Digital	3 × 5 × 1 mm
PTE7100 SERIES	Sensata Technologies	-40 ℃ ~ 100 ℃	0-600 bar			Analog voltage	Analog	24 × 24 × 63 mm
PT5400	ifm efector, inc.	-40 ℃ ~ 100 ℃	0-400 bar			Analog voltage	Analog	19 × 19 × 66 mm
803145	Amphenol i2s	-40 ℃ ~ 125 ℃	0-400 bar			Analog voltage	Analog	27 × 27 × 65 mm
FD-H series	keyence	-10 ℃ ~ 180 ℃		0 ~ 300 LPM		Analog voltage	Analog/Digital	120 × 980 × 90 mm
SU9004	ifm efector, inc.	-10 ℃ ~ 80 ℃		0 ~ 200 LPM		Analog voltage	Analog	130 × 100 × 90 mm
SM2000	ifm efector, inc.	-10 ℃ ~ 80 ℃		5 ~ 600 LPM		analogue	Analog/Digital	200 × 100 × 103 mm

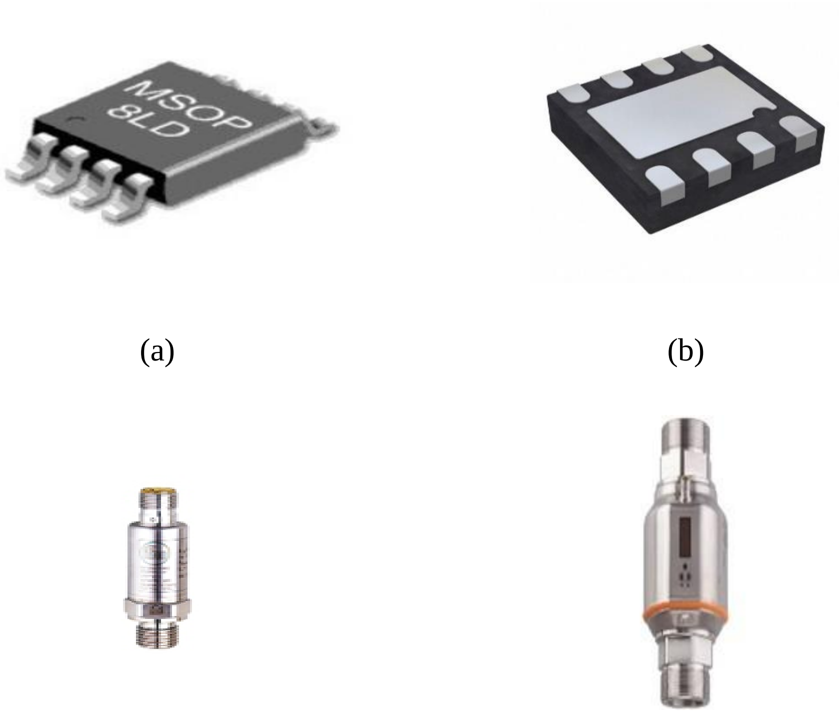
Temperature Sensor

Acceleration Sensor

Pressure sensor

Flow Sensor

The best location for a pressure sensor in a hydraulic hammer will depend on several factors, including the design of the hammer, the type of application, and the specific measurement requirements. In general, a pressure sensor should be located in a position that is representative of the hydraulic pressure within the hammer during operation. This may be at a point close to the hydraulic fluid inlet or outlet, or at a point where the pressure is known to be high. One common location for a pressure sensor in a hydraulic hammer is in the hydraulic circuit that controls the striking action. This can provide valuable feedback on the pressure and force generated during the hammer's operation, allowing operators to monitor and adjust the performance of the hammer as needed.



(a) (b)

(c) (d)

Fig 2- 8 A sampling of the sensors used in the Indeconnect data logger. (a) Temperature, (b) Acceleration, (c) Pressure Sensor, (d) Flow Sensor.

As mentioned before, Since the highest pressure in hydraulic breakers occurs in a cavity in the rear of the head, the sensors are placed in this location to obtain the most accurate pressure measurements. It must be emphasized that all sensors were tested for accuracy and performance at Indco's test facility. Figure 2-9 shows Keyence's Ultrasonic clamp-on flow meters technology flow sensor tested for accuracy, performance, and potential uses of the sensor. Ultrasonic clamp-on flow meters are widely used in various industries to measure the flow rate of fluids, including gases and liquids, in closed pipes without the need for invasive installation.

The working principle of ultrasonic clamp-on flow meters is based on the transit-time method, where two ultrasonic sensors are placed on the exterior surface of the pipe, facing each other, at a specific angle. One sensor acts as the transmitter, and the other acts as the receiver. The transmitter sends ultrasonic pulses that travel through the fluid and reach the receiver, where they are detected. The time it takes for the ultrasonic pulses to travel from the transmitter to the receiver, both upstream and downstream, is measured and used to calculate the flow rate of the fluid.

Ultrasonic clamp-on flow meters are known for their high accuracy, non-invasive nature, and the ability to measure bidirectional flow, making them suitable for a range of applications, including industrial processes, water and wastewater treatment, and hydraulic systems.

Fig 2- 9. Keyence FD-H series for monitoring the inlet of hydraulic hammers in the Indeco test bench house.

2.4.4 Microcontroller

A microcontroller is a type of small computer designed for specific embedded applications. It contains a microprocessor, memory, and input/output peripherals on a single chip, making it a highly integrated and compact solution. Microcontrollers are commonly used in a variety of electronic devices, such as appliances, automotive systems, medical devices, and industrial control systems, to provide control and monitoring functions.

Microcontrollers come in a variety of types, with different specifications and capabilities. Some of the factors to consider when selecting a microcontroller include:

- **Processing power:** The processing power of the microcontroller determines its ability to execute instructions and perform tasks quickly and efficiently.
- **Memory:** The amount and type of memory on the microcontroller is an important consideration when designing an embedded system, as it affects the program and data storage capabilities of the device.
- **Peripherals:** The types and number of input/output peripherals on the microcontroller determine its ability to interface with external devices, such as sensors, actuators, and communication modules.
- **Power consumption:** Power consumption is a critical factor in battery-powered devices, as it affects the device's operating time.
- **Cost:** The cost of the microcontroller is an important consideration when selecting a device for an embedded system, as it affects the overall cost of the product.

Today, two controllers are typically used in IIoT devices: ARM (Advanced RISC Machines) and FPGAs (Field-Programmable Gate Arrays). ARM controllers and FPGAs are both types of integrated circuits used in electronic devices, but they differ in their architectures, design philosophies, and applications.

ARM (Advanced RISC Machines) controllers are a type of microcontroller that is widely used in embedded systems and IoT devices. They are based on a Reduced Instruction Set Computing (RISC) architecture, which simplifies the instruction set and execution pipeline, resulting in high performance and low power consumption. ARM controllers are typically designed for specific applications and offer a range of features, such as multiple cores, advanced interrupt handling, and communication interfaces. They can be programmed using a variety of programming languages and development tools, and are suitable for a wide range of applications, including data logging.

FPGAs (Field-Programmable Gate Arrays) are hardware devices that can be programmed to implement specific logic functions. They offer high processing speeds, low latency, and a high degree of parallelism, making them suitable for high-speed data logging applications. FPGAs are highly configurable, making it possible to implement custom logic functions in hardware, resulting in highly optimized and efficient data logging systems.

In general, ARM controllers are more flexible and easier to program than FPGAs. Therefore, ARM controllers are more commonly used in IIoT devices than FPGAs due to their ease of programming and lower cost. In addition, ARM controllers are a good choice for data logging applications that require low power consumption and a high degree of programmability.

The microcontroller used is a STM32L496RET6 ARM-Cortex®- manufactured by ST (see Figure 2-10). It communicates with the server using Serial communication through a USB-to-serial bridge. The STM32L496RET6 is a microcontroller unit (MCU) based on the ARM-Cortex-M4 architecture. It is a 32-bit MCU with a core voltage of 1.7 volts and a clock speed of up to 80 MHz. The STM32L496RET6 features 512KB of flash memory and 320KB of SRAM, providing ample space for program and data storage, also features a range of peripherals, including up to 3 I2C, 3 USART, 4 SPI, 4 16-bit timers, and a real-time clock. It also includes advanced features such as a digital signal processor (DSP), an analog-to-digital converter (ADC) with up to 16-bit resolution, and a hardware encryption accelerator.

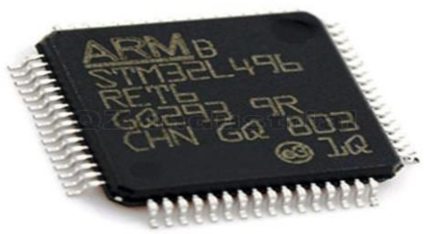


Fig 2- 10. Microcontroller STM32L496RET6 ARM-Cortex®-M4.

2.4.5 GSM/GPS Module

GPS (Global Positioning System) and GSM (Global System for Mobile Communications) are two distinct technologies that are commonly used in mobile devices and other applications. GPS is used for location tracking and navigation, while GSM is used for communication with cellular networks. In certain applications, GPS and GSM may be used together for location tracking and data transmission purposes. In the Industrial Internet of Things, (GPS and GSM technologies can be used to enable location tracking and remote communication for industrial equipment and devices. For example, GPS can be used to track the location of mobile assets such as vehicles or containers, while GSM can be used to transmit data from sensors or control systems to a central server or cloud-based platform. By integrating GPS and GSM into IIoT systems, businesses can improve operational efficiency, monitor assets in real-time, and enable remote management and control of industrial equipment.

The requirements for GPS/GSM modules in IIoT devices depend on the specific use case and application. Some of the common requirements may include:

- **Location accuracy:** GPS modules should provide accurate location data to enable real-time tracking and monitoring of assets.
- **Network coverage:** GSM modules should be compatible with the local cellular networks and provide reliable connectivity.
- **Low power consumption:** IIoT devices may be deployed in remote locations or have limited power sources, so GPS/GSM modules should be energy-efficient to prolong battery life.
- **Security:** GPS/GSM modules should have built-in security features to protect against unauthorized access or tampering of data.
- **Integration:** GPS/GSM modules should be compatible with the device's hardware and software architecture to enable easy integration and data transfer.
- **Update Rate:** The update rate refers to the frequency at which a GPS or GNSS module calculates and reports its position. Typically, devices have a standard update rate of 1Hz, which means that they update and report their position once every second. However, higher update rates of 5-10Hz may be necessary for faster-moving vehicles, although this is not typically required in most real-world scenarios.
- **Antenna:** It's important to keep in mind that the GPS module is receiving signals from satellites that are located approximately 12,000 miles away in

the sky. In order to ensure the best possible performance, it's important to have an unobstructed path between the antenna and most of the sky. While weather conditions like clouds and snowstorms generally don't affect the signal, trees, buildings, mountains, and even the roof over your head can all create unwanted interference that can lead to decreased GPS accuracy. Therefore, selecting the right antenna is critical to achieving optimal performance.

- **Cost:** The cost of GPS/GSM modules should be reasonable and affordable to ensure cost-effective deployment and scalability of IIoT solutions.

There are various types of GPS and GSM modules available in the market for IIoT devices, each with their own features and capabilities. Here are some examples:

GPS Modules:

1. **Standalone GPS modules:** These are self-contained GPS units that provide location data without requiring any external components or connections.
2. **GPS receiver modules:** These modules receive GPS signals from satellites and provide location data to the device's microcontroller for further processing.
3. **Assisted GPS (A-GPS) modules:** These modules use cellular network data to assist in GPS signal acquisition and improve location accuracy.

GSM Modules:

1. **2G GSM modules**: These modules use the second generation of GSM technology and provide basic voice and data connectivity.
2. **3G GSM modules:** These modules use the third generation of GSM technology and offer higher data transfer rates and improved network coverage.
3. **4G GSM modules:** These modules use the fourth generation of GSM technology and provide even higher data transfer rates and better network efficiency.
4. **Narrowband IoT (NB-IoT) modules:** These modules use low-power, wide-area (LPWA) cellular networks to provide low-cost, low-bandwidth connectivity for IIoT devices.

Considering that Internet coverage is now 3Gand 4G in most areas, the design and construction of the Indeconnect data logger used 3G and 4G module and A-GPS to improve location accuracy. However, since hydraulic breakers are used in uninhabited and remote areas where Internet access may be impossible, an external storage option was also provided to ensure that data could be collected and stored even when an Internet connection was not available. Since data is typically evaluated on a daily or weekly basis, there should be sufficient time to transfer the data from the external storage card to the cloud once Internet connectivity is restored. With this approach, you can ensure that important data is not lost even if the Internet connection is limited [34].

The GSM/GPS module used is SIM7600EI 4G / 3G / GSM / GPRS / GPS UART Modem–rhydoLABZ (Figure 2-11). The SIM7600EI 4G/GSM/GPRS/ GPS UART modem is a high-quality commercial grade product from rhydoLABZ, professionally developed with impedance-matched RF PCB designs and equipped with a multi-band LTE-TDD /LTE-FDD/HSPA+/UMTS/EDGE/GPRS/GSM module solution in an LCC type that supports LTE CAT1 with up to 10Mbps for downlink and 5Mbps for uplink data transfer.

Fig 2- 11. GPS/GSM module: SIM7600EI rhydoLABZ.

The SIM7600EI with its compact and unified form factor is compatible with SIMCom HSPA+ SIM5360 module/LTE CAT3SIM7100 and LTE CAT4 SIM7600E-H module, so you only need to design your application once for different technologies and can benefit from great time saving in development. This 3G and 4G Modem is coming with selectable interfacing voltage, which allows you to connect 2V8 to 5V, including 3V3. Microcontroller can directly connect without any extra level conversion chips irrespective of voltage level. Onboard TXB0108 voltage level translator IC which helps us to interface with 2.8V to 5V Micro Controllers.

2.4.6 Source Power

"Source power" generally refers to the power supply that provides electrical energy to a device or system. It can be a battery, an AC or DC power outlet, or any other source that is capable of delivering the required voltage and current to power the device or system. When selecting a battery for an Industrial Internet of Things (IIoT) device, several factors must be considered, including the power requirements of the device, the expected usage pattern, and the environmental conditions in which the device will be used. Here are some guidelines to help you choose the right battery for your IIoT device:

- **Determine the power requirements:** The first step in choosing a battery is to determine the power requirements of the device. It needs to know the voltage and current that thew device needs to operate, and the amount of power it will consume over time.
- **Consider the expected usage pattern:** How often will the device be used, and how long will it need to operate on a single charge? Will it be used continuously or intermittently? These factors will help to determine the size of the battery we need and how frequently it will need to be recharged.
- **Choose the right chemistry:** There are several types of batteries available, including lithium-ion, nickel-cadmium, and lead-acid. Lithium-ion batteries are a popular choice for IIoT devices because they offer high energy density, long life, and low maintenance requirements.
- **Consider the environmental conditions:** The environment in which device will be deployed can affect the performance and lifespan of battery. For example, extreme temperatures, humidity, and vibration can all affect battery performance.
- **Consider the cost:** The cost of the battery is an important factor to consider, as it will affect the overall cost of IIoT device. Lithium-ion batteries tend to be more expensive than other types of batteries, but they also offer longer life and higher energy density.

Due to the high voltage and current requirements of pressure and flow sensors, the use of a large battery is required. To solve this problem and improve system performance, a smaller battery is considered to support the system while the data logger is connected directly to the power source of the excavator or backhoe loader to operate the sensors. The battery is used to transfer data to the cloud when the machine is off. The battery life used by GPS, the accelerometer, the temperature sensor and the GSM/GPS module for data transmission is as follows:

- Rechargeable Li-Polymer 4800 mAh/3.7V battery
- Standby Time (2 hours active tracking per day)
- Every10 minutes reporting: 170 days
- Every 5 minutes reporting: 90 Days
- Every 1 minute reporting: 35 Days

2.4.7 IIoT enclosure

An IIoT enclosure is a protective casing that houses and protects electronic components and devices that are used in industrial settings for IIoT applications. These enclosures are designed to protect the electronic components from environmental factors such as dust, water, heat, and humidity.

IIoT enclosures play a crucial role in ensuring that IIoT devices function optimally in harsh industrial environments. They provide protection against physical damage, as well as electromagnetic interference and radio frequency interference. In addition to protecting the devices, IIoT enclosures also help to ensure the safety of workers by preventing accidental contact with live electronic components.

IIoT enclosures come in different sizes and shapes to accommodate different types of devices. Some enclosures are also designed to be mounted on walls or poles for easy installation. IIoT enclosures can be made from a variety of materials, depending on the specific requirements of the application. Some of the most common materials used for IIoT enclosures include:

1. **Plastic:** Plastic enclosures are lightweight and can be easily molded into different shapes and sizes. They are also resistant to corrosion and can provide good protection against water and dust.
2. **Stainless Steel:** Stainless steel enclosures are strong, durable, and resistant to corrosion. They can withstand high temperatures and harsh environmental conditions, making them ideal for use in industrial settings.
3. **Aluminum:** Aluminum enclosures are lightweight, easy to work with, and have excellent thermal conductivity. They can also provide good protection against electromagnetic interference (EMI) and radio frequency interference (RFI).

4. **Fiberglass:** Fiberglass enclosures are strong, durable, and resistant to corrosion. They can provide good protection against UV radiation, making them ideal for outdoor use.

The choice of material for an IIoT enclosure will depend on factors such as the environment in which it will be deployed, the size and weight of the device, and the level of protection required.

The IP (Ingress Protection) rating system is a standard used to classify the degree of protection provided by an enclosure against the entry of foreign objects, such as dust and water. The first digit of the IP rating indicates the level of protection against solid objects, while the second digit indicates the level of protection against liquids. When an IIoT enclosure is required to have a minimum IP66 rating for external mounting, this means that the enclosure must provide a high level of protection against the ingress of dust and water.

An IP66 rating means that the enclosure is dust-tight and can withstand powerful water jets from any direction without water entering the enclosure. This level of protection is suitable for outdoor applications where the enclosure is exposed to harsh environmental conditions.

To achieve an IP66 rating for an IIoT enclosure, the enclosure must be designed to meet specific criteria for ingress protection, such as having a gasket or seal to prevent water and dust from entering the enclosure, and having a robust construction that can withstand external impacts and vibration.

The indeconnect data logger with standard IP 66 housing is shown in Figure 2-12. The IIot enclosure is made of corrosion, water and dust resistant plastic by Trusted global company (Trusted A/S).

Figure 2-13 illustrates the placement of the Indeconnect data logger on the exterior of the machine, which makes it susceptible to damage from the harsh work environment. To safeguard against this, a protective fiberglass case was designed and fabricated. The case is intended to shield the data logger from potential harm caused by impacts, external vibrations, dust, and water. The key aspect of the case design is that it does not obstruct direct contact between the data logger and the machine, enabling the sensors, including accelerometers and vibration sensors, to function without interference.

Fig 2- 12. The Indeconnect data logger, manufactured by Trusted Global Denmark.

In harsh environments, data loggers used for monitoring hydraulic hammers may experience several external threats, such as:

- **Extreme temperatures:** Data loggers may be exposed to extreme temperatures, either hot or cold, which can cause damage to the device or reduce its battery life. High temperatures can cause the device to malfunction or even stop working, while low temperatures can cause the battery to drain quickly.
- **Moisture and humidity:** Data loggers used in harsh environments may be exposed to moisture and humidity, which can cause corrosion or damage to the device. Moisture can also affect the accuracy of the readings, especially if the device is not properly sealed.
- **Vibration and shock:** Hydraulic hammers generate high levels of vibration and shock, which can affect the accuracy of the data logger's readings. The device may need to be designed to withstand these conditions or secured in a protective casing.
- **Dust and debris:** Dust and debris can accumulate on the sensors of the data logger, leading to inaccurate readings. The device may need to be designed with a protective covering or shield to prevent dust and debris from entering the sensors.
- **Electromagnetic interference:** In harsh environments, there may be high levels of electromagnetic interference (EMI) from other nearby equipment, which can interfere with the device's sensors and affect the accuracy of the readings. The device may need to be designed with special shielding or filters to prevent EMI from affecting the readings.

Fig 2- 13. the positioning of the data logger on the hydraulic equipment and the protective cover for the data logger.

Due to Indeco's and Trusted global's business policy, it is not possible to publish the serial number of the devices as well as the components and the circuit board. But the basic electronic and schematic diagrams of the Indeconnect device are depicted in Figure 2-14. This device was outsourcing to Trusted Global to complete the final stages of product development and manufacturing.

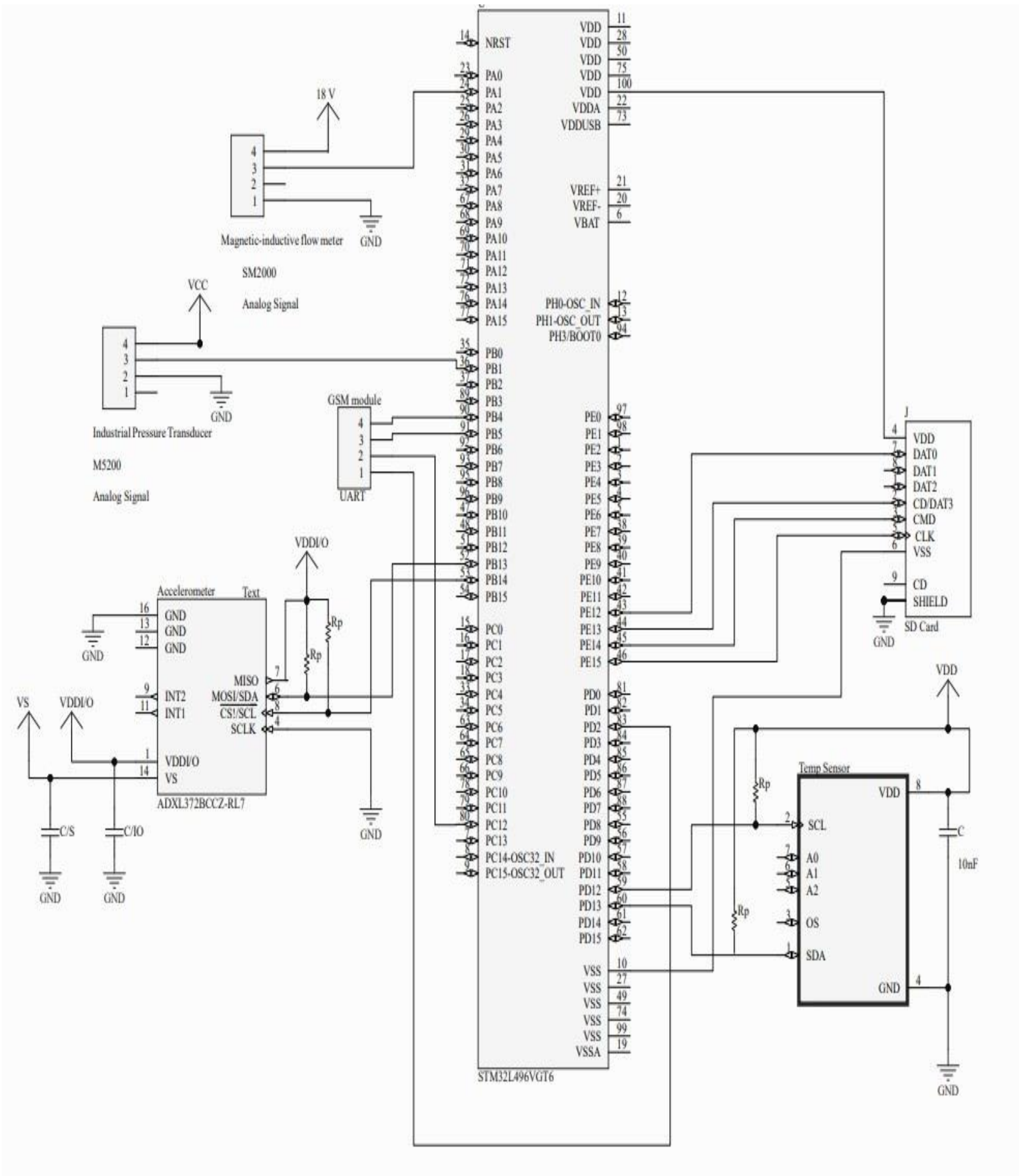


Fig 2-14. The Electronic Diagrams and Schematics of the INDECONNECT data logger.

2.5 A Cloud Architecture for Device monitoring and Data Collection

In the last decade, the growth of open source and hardware has led to technology playing an important role not only in industry but in all aspects of human society. A key feature of a cyber-physical system is its sensing or monitoring part. The demand for remote monitoring and control of industrial processes, equipment, and machinery in a short time will grow rapidly due to the increasing use of technology and Internet access. The remote monitoring system provides data that can be used to optimize production and perform predictive maintenance.

The cloud architecture for device monitoring and data collection typically involves the use of a cloud platform to collect and process data from connected devices. The devices are usually equipped with sensors that gather data on various parameters such as temperature, pressure, and flow rates. The data is then transmitted to a cloud-based platform where it is stored and processed using a variety of tools and services.

The cloud architecture typically consists of three main layers: the device layer, the communication layer, and the cloud layer. The device layer includes the connected devices and the sensors used to gather data. The communication layer facilitates the transfer of data from the devices to the cloud platform using wired or wireless connections. The cloud layer includes the various services and tools used to process and analyze the data collected from the devices.

Some of the key components of the cloud architecture for device monitoring and data collection include data storage and management systems, data analytics tools, machine learning algorithms, and visualization tools. These components work together to enable the collection, processing, and analysis of data from devices in real-time, which can help improve operational efficiency, reduce downtime, and optimize performance.

It is emphasized that the architecture is designed with sensors and data acquisition modules on each hydraulic control unit. Data transmission is one of the most important parts of the design of this device. For this purpose, the gateway is equipped with appropriate protocols. This device uses the gateway with the protocols IEEE 802.15.4 and IEEE 802.11 for communication between both sides of the connection. IEEE 802.15.4 and IEEE 802.11 are two data transmission protocols used in wireless communication.

IEEE 802.15.4 is a low-power, low-data-rate wireless communication protocol designed for low-cost, low-complexity applications. It operates in the 2.4 GHz band and uses a star or mesh network topology. It is commonly used in wireless sensor networks, industrial control, and home automation applications. The protocol supports data rates of 20, 40, or 250 kbps and can operate at distances of up to 100 meters.

IEEE 802.11, on the other hand, is a more powerful wireless communication protocol designed for higher data rates and more complex applications. It is commonly used in local area networks (LANs) and supports a wide range of devices, including laptops, smartphones, and tablets. It operates in the 2.4 GHz and 5 GHz bands and uses a peer-to-peer or infrastructure network topology. The protocol supports data rates of up to 10 Gbps and can operate at distances of up to 100 meters.

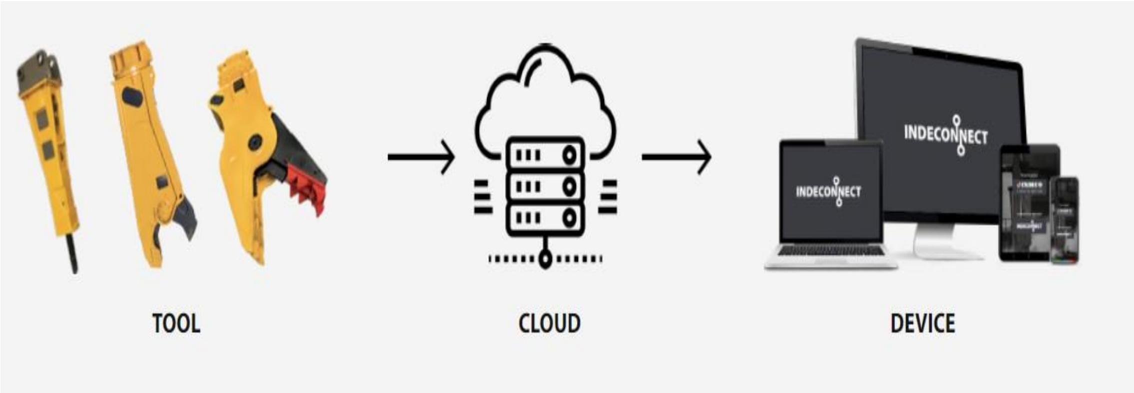


Fig 2- 15. Architecture overview.

When it comes to data transmission protocols for IIoT (Industrial Internet of Things) devices, there are several options available, each with its own advantages and disadvantages. Some of the most commonly used data transmission protocols for IIoT devices include:

1. **MQTT (Message Queuing Telemetry Transport):** MQTT is a lightweight and efficient publish-subscribe messaging protocol designed for use in IoT applications. It is ideal for low-power devices and can handle low-bandwidth networks. MQTT is widely used in IIoT applications, particularly those involving machine-to-machine communication.
2. **CoAP (Constrained Application Protocol):** CoAP is a lightweight application-layer protocol designed for use with constrained networks and devices. It is particularly useful for low-power devices with limited processing power and memory. CoAP is used in IIoT applications that require real-time communication and efficient use of network resources.

3. **OPC UA (Open Platform Communications Unified Architecture):** OPC UA is a standardized communication protocol designed for industrial automation applications. It provides a platform-independent way to exchange data between devices and systems, making it an ideal choice for IIoT applications.
4. **AMQP (Advanced Message Queuing Protocol):** AMQP is a message-oriented middleware protocol that provides reliable, asynchronous communication between devices and systems. It is particularly useful in IIoT applications that require high reliability and fault-tolerance.
5. **DDS (Data Distribution Service):** DDS is a middleware protocol designed for real-time, high-performance communication between devices and systems. It is particularly useful in IIoT applications that require real-time control and monitoring.

When choosing a data transmission protocol for IIoT devices, it is important to consider factors such as the type of application, network bandwidth, device capabilities, and the level of security and reliability required.

In this study, Message Queuing Telemetry Transport system is used for data transmission. Generally, It's difficult to say that MQTT is "better" than other data transmission protocols, as each protocol has its own strengths and weaknesses and is suited to specific use cases. However, MQTT is a popular choice for many IoT applications and has several advantages that make it a strong contender in the IIoT space.

Some of the advantages of MQTT include:

- **Lightweight and efficient:** MQTT is designed to be lightweight and efficient, which makes it well-suited for use in IoT applications. It uses a publish/subscribe model, which reduces network traffic and makes it more efficient than other protocols.
- **Low bandwidth requirements:** MQTT requires very little bandwidth to operate, which makes it ideal for use in low-power, low-bandwidth environments. This also makes it an ideal choice for remote devices that may have limited network connectivity.
- **Quality of Service (QoS) levels:** MQTT provides several different QoS levels, which allow devices to prioritize the delivery of messages based on the importance of the data being transmitted. This ensures that critical data is delivered in a timely and reliable manner.

- **Wide availability:** MQTT is widely available and supported by a large number of IoT platforms and devices. This makes it easy to integrate with existing systems and ensures that it will be supported for years to come.
- **Security:** MQTT provides several security features, including encryption and authentication, which help to protect data as it is transmitted over the network.

2.5.1 IoT Cyber Security Challenges

The idea of the Internet of Things is to connect every object to make these processes more efficient, provide more comfort, and improve our business and personal lives. But connecting objects such as cars, homes, and machines also exposes lots of sensitive data. Some of this data is not meant for the public and should be protected by the pillars of information security: confidentiality, integrity, and availability.

The challenges of data leaks, insecure communications, and software and firmware vulnerabilities are all significant risks facing Industrial Internet of Things (IIoT) systems. These risks can lead to data breaches, loss of sensitive information, and damage to a company's reputation and financial standing.

Software and firmware vulnerabilities are security weaknesses that can be exploited by attackers to gain unauthorized access to a system or device. These vulnerabilities can be caused by errors in the design, development, or implementation of software or firmware.

Software vulnerabilities can occur in any software application, including operating systems, databases, web applications, and mobile apps. Common software vulnerabilities include buffer overflows, SQL injection, cross-site scripting (XSS), and insecure authentication and authorization mechanisms.

Firmware vulnerabilities, on the other hand, are specific to embedded systems and IoT devices, which often run on specialized firmware. Firmware vulnerabilities can be caused by flaws in the firmware design or implementation, or by the use of insecure communication protocols or default passwords.

Both software and firmware vulnerabilities can be exploited by attackers to gain unauthorized access to a system or device, steal data, install malware, or launch other attacks.

To mitigate the risk of software and firmware vulnerabilities, it is important to implement security best practices throughout the software development and deployment process. This can include using secure coding practices, performing regular vulnerability assessments and penetration testing, keeping software and firmware up to date with the latest security patches, and implementing proper access control and authentication mechanisms. Additionally, it is important to have a plan in place for responding to security incidents and to regularly review and update security policies and procedures.

Insecure communications refer to the transmission of data over a network or communication channel in an unencrypted or poorly secured form, which can be intercepted and read by attackers. Insecure communications can occur in various forms of communication, including email, instant messaging, file sharing, and web browsing.

One of the most common examples of insecure communication is the use of unencrypted HTTP (Hypertext Transfer Protocol) for web browsing, which can allow attackers to intercept and read data transmitted between a user's browser and a web server. To address this issue, many websites now use HTTPS (HTTP Secure), which encrypts data transmitted between the user's browser and the web server.

Insecure communications can also occur in IoT devices and systems. For example, many IoT devices use unencrypted communication protocols, such as MQTT or CoAP, which can be intercepted and read by attackers. To address this issue, it is important to use secure communication protocols, such as TLS (Transport Layer Security), which encrypts data transmitted between devices.

To mitigate the risk of insecure communications, it is important to implement encryption and other security measures throughout the network and communication channels. This can include using secure communication protocols, implementing proper access control and authentication mechanisms, and regularly reviewing and updating security policies and procedures. It is also important to educate users and employees about the risks of insecure communications and how to avoid to data leaks.

Data leaks from Industrial Internet of Things (IIoT) systems can be a significant cyber security risk, as they can result in the unauthorized disclosure of sensitive or confidential data. These data leaks can occur in various ways, including through accidental disclosure, insider threats, and external attacks.

Accidental disclosure can occur when sensitive data is inadvertently made public or shared with unauthorized individuals, such as through misconfigured servers, unsecured file transfer protocols, or human error. Insider threats, on the other hand, can result from employees or contractors who have access to sensitive data and intentionally or unintentionally disclose it. External attacks, such as hacking, malware, or phishing attacks, can also result in data leaks from IIoT systems.

The consequences of a data leak can be severe, including damage to the company's reputation, loss of intellectual property, regulatory penalties, and financial loss. To mitigate the risk of data leaks from IIoT systems, it is important to implement comprehensive security measures, such as access control, encryption, and data loss prevention (DLP) technologies. It is also important to conduct regular vulnerability assessments and penetration testing to identify and address potential vulnerabilities.

2.5.2 MQTT (Message Queuing Telemetry Transport)

MQTT (Message Queuing Telemetry Transport) is a lightweight, publish-subscribe messaging protocol designed for efficient communication between devices in an IoT (Internet of Things) network. It was developed in 1999 by Andy Stanford-Clark and Arlen Nipper, and later became an OASIS standard in 2014.

MQTT uses a publish-subscribe model, where publishers send messages, or "publish" data, to a broker, and subscribers receive messages, or "subscribe" to specific topics on the broker. The broker acts as a mediator between publishers and subscribers, managing the flow of messages.

The MQTT protocol is based on TCP/IP (Transmission Control Protocol/Internet Protocol). Both the MQTT client and the broker need to have a TCP/IP stack to communicate with each other over a network [35].

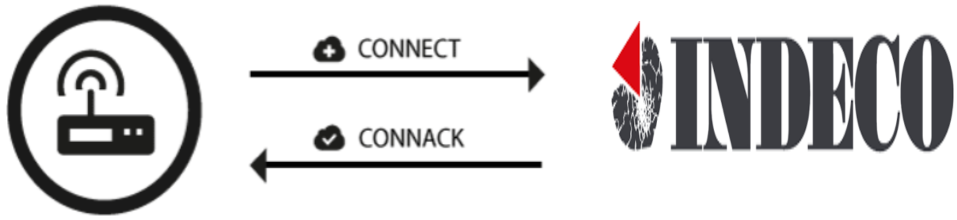
TCP is a reliable, connection-oriented protocol that provides guaranteed delivery of data, while IP is a connectionless, best-effort protocol that provides the basic routing and addressing functions for data transmission over the internet. Together, these two protocols form the foundation of the internet and are widely used for communication between devices on local and wide area networks.

MQTT is designed to be a lightweight protocol that uses minimal bandwidth and is well-suited for use in resource constrained IoT devices. It uses the TCP/IP stack to provide reliable delivery of messages between the MQTT client and broker, while also supporting features such as quality of service (QoS) levels and persistent sessions.

In the MQTT protocol, communication is always between a client and a broker. Clients never connect to each other directly. To initiate a connection, the client sends a CONNECT message to the broker, which includes the client's identification, username, password (if required), and the level of quality of service (QoS) required for the connection. The CONNECT message also specifies whether the connection is a clean session or a persistent session (see figure 2-16).

Once the broker receives the CONNECT message, it performs authentication and authorization checks, and then responds with a CONNACK message. The CONNACK message contains a status code that indicates whether the connection was successful or not, and if not, what the reason was.

Once the connection is established, the client can then send and receive messages from the broker using the PUBLISH, SUBSCRIBE, and UNSUBSCRIBE messages. The broker is responsible for routing the messages to the appropriate clients based on the topics to which they have subscribed, and for ensuring that messages are delivered according to the specified QoS level. Figure 2-17 shows the content of a connection message sent by an MQTT client.



MQTT Client **MQTT Broker**

Fig 2- 16. The MQTT connection between client and broker.

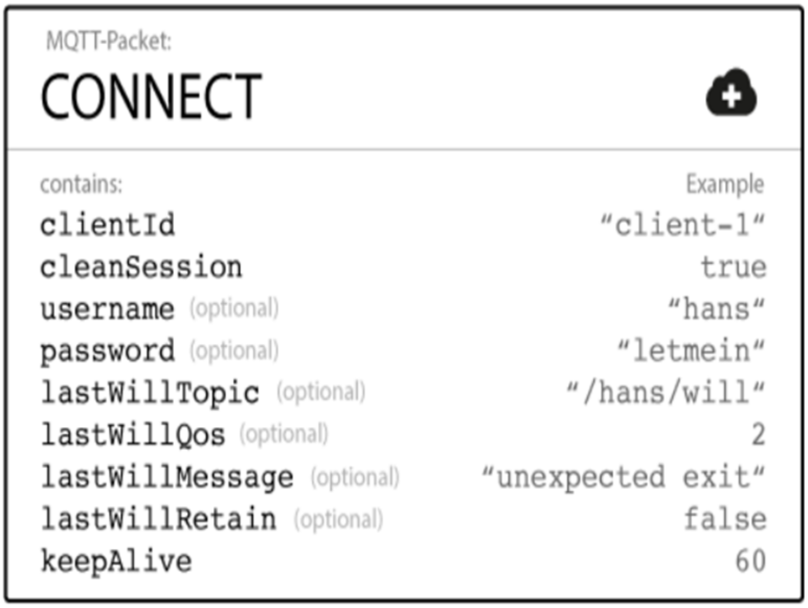


Fig 2- 17. A MQTT client-connect message.

When an MQTT client connects to an MQTT broker, it sends a CONNECT message to initiate the connection. The CONNECT message contains various parameters and options that the client uses to specify its connection details and configuration. Here is an example of a typical CONNECT message:

10 22 00 04 4D 51 54 54 04 C2 00 3C 00 0A 74 65 73 74 5F 63 6C 69 65 6E 74 00 05 61 64 6D 69 6E 00 05 70 61 73 73 77 64

The message is made up of several fields, each with its own purpose:

Byte 1: Control Packet type (0x10 for CONNECT)

Byte 2: Remaining Length (22 bytes)

Byte 3-4: Protocol Name (0x00 0x04 for MQTT)

Byte 5: Protocol Version (0x04 for MQTT version 3.1.1)

Byte 6: Connect Flags (0xC2 - this sets Clean Session flag and specifies Will Flag, Will QoS, and Will Retain Flag)

Byte 7-8: Keep Alive Timer (in seconds)

Byte 9-10: Client Identifier Length (0x00 0x3C or 60 bytes)

Byte 11-70: Client Identifier (in this case, "test_client")

Byte 71-72: Will Topic Length (0x00 0x05 or 5 bytes)

Byte 73-77: Will Topic (in this case, "admin")

Byte 78-79: Will Message Length (0x00 0x05 or 5 bytes)

Byte 80-84: Will Message (in this case, "passw")

The CONNECT message is the first step in establishing an MQTT connection and is essential for configuring the connection parameters and ensuring that the client and broker are able to communicate effectively.

MQTT provides security measures that are divided into multiple layers, with each layer designed to prevent different types of attacks. The goal of MQTT is to provide a lightweight and easy-to-use communication protocol for the Internet of Things, while also ensuring that the data transmitted over the protocol is secure.

The security measures in MQTT are divided into the following layers:

- **Authentication:** MQTT implementations can provide mechanisms for authenticating clients and brokers, such as through the use of usernames and passwords or digital certificates. This ensures that only authorized clients and brokers are able to connect and communicate over the MQTT network.

- **Transport Layer Security (TLS):** This layer provides encryption and authentication of data between the MQTT client and server. It is based on the SSL/TLS protocol, and provides protection against eavesdropping, tampering, and other types of attacks.
- **Access Control:** This layer provides authentication and authorization of clients based on their identity and permissions. It allows MQTT brokers to control which clients can access which topics, and what actions they can perform.
- **Message Filtering:** This layer provides the ability to filter messages based on content and topic, and can be used to prevent malicious messages from reaching subscribers. It is particularly useful for preventing attacks such as denial of service (DoS) attacks and message flooding.
- **Logging and Auditing:** MQTT implementations can provide mechanisms for logging and auditing MQTT communication, such as through the use of log files or event records. This can help identify and investigate security incidents or anomalies in the network.
- **Payload Encryption:** This layer provides encryption of the message payload, which contains the actual data being transmitted. This layer can be used to protect sensitive data from being intercepted or modified.

Overall, MQTT can be used to send sensor data to the cloud with high security and reliability. By using MQTT in combination with secure authentication and encryption mechanisms, sensor data can be transmitted securely over the internet to cloud-based servers for storage and analysis.

For instance, a typical IoT system using MQTT might include sensors that gather data, an MQTT client that sends the data to an MQTT broker, and a cloud-based server that subscribes to the data and stores it for later use. To ensure the security of the data being transmitted, the MQTT client might use a secure authentication mechanism such as username/password or digital certificates to authenticate itself to the MQTT broker, and encryption such as TLS/SSL to protect the data being transmitted.

Once the data is received by the cloud-based server, it can be stored and analyzed using cloud-based data processing and analytics tools. The data can also be shared with other systems and applications using MQTT or other communication protocols, allowing for interoperability and integration with other systems.

2.5.3 API (Application Programming Interface) and Data visualization

Providing an API (Application Programming Interface) means making it available to other applications or users. To do this, the API must be made accessible on a server or cloud platform, from where it can be accessed via HTTP requests. Deploying an API typically involves configuring the code that defines the API to run in a server environment that clients can access over the Internet.

The deployment process often involves setting up infrastructure components such as web servers, load balancers, and databases to support the API, as well as configuring security and access controls to protect the API from unauthorized access. It may also involve packaging the code into a deployable format, such as a Docker container or a zip file, and deploying it to the target environment using tools like Git, Jenkins, or AWS Code Deploy [36].

Once an API is deployed, other applications and developers can access it via HTTP requests. This allows other applications to interact with the API's functions and data, and the API provides a standard interface for integration with other systems. By using an API, developers can provide a reliable and scalable platform for accessing their services and data and facilitate integration with other applications and services [37].

Data visualization by MQTT API involves using MQTT protocol to receive real-time data from IIoT devices and visualizing it in real-time on a dashboard.

Here are the general steps involved in data visualization by MQTT API:

1. **Set up an MQTT broker:** The first step is to set up an MQTT broker that will receive, and store messages sent from IIoT devices. There are several open-source MQTT brokers available, such as Mosquitto and HiveMQ, or can be used a cloud-based MQTT broker like AWS IoT or Google Cloud IoT.
2. **Connect to the MQTT broker:** Once the MQTT broker has been set up, it needs to be connected to using an MQTT client, and authentication credentials will need to be provided to connect to the broker. Several MQTT clients are available in different programming languages, such as Python's Paho and Java's Eclipse..
3. **Subscribe to MQTT topics:** To receive data from IIoT devices, it needed to subscribe to MQTT topics. Topics are essentially channels on which messages are published. Topics related to the specific data that needs to be

visualized, such as temperature, pressure, or vibration data, can be subscribed to.

4. **Receive and process data:** Once the relevant MQTT topics have been subscribed to, data will start to be received in real-time, and it will need to be processed and prepared for visualization. This could involve data cleaning, preprocess, feature engineering, … .

There are also general procedures for displaying data via the cloud for IIoT devices:

- **Identifying the data to be visualized:** The type of data that needs to be visualized from the cloud for IIoT devices should be identified first, and it can include sensor data, machine logs, or any other type of data that requires analysis.
- **Choosing a cloud-based visualization tool:** There are several cloud-based visualization tools that can be used to visualize IIoT data, such as AWS QuickSight, Microsoft Azure and Google Data Studio.
- **Connect to the cloud-based data source:** Connect to the cloud-based data source where the IIoT data is stored. Depending on the data source, it may be needed to authenticate or provide other credentials to access the data.
- **Retrieving the data:** Retrieve the relevant data from the cloud-based data source. Depending on the data source, this can be done using SQL queries, RESTful APIs, or other methods of data retrieval.
- **Pre-process the data:** Pre-process the data, if necessary, to format it and remove missing or incorrect data points. This step could include filtering, sorting, grouping, and aggregating the data.
- **Create the visualization:** use the selected cloud-based visualization tool to create visualizations of the IIoT data. In doing so, you can create charts, graphs, maps, or any other type of visualization that best represents the data. Experiment with different types of visualizations to find the most effective way to communicate insights.
- **Publish the visualization:** publish the visualization on a dashboard or in a report that other team members or stakeholders can access. You can also set up real-time streaming of IIoT data to automatically update the visualization with new data as it arrives.

In this project, for security and marketing reasons, the company Indeco has created its own application and website for the visualization of data and direct communication with customers. You can download the applications of this project through the following links:

https://play.google.com/store/apps/details?id=com.teseo.indeconnect&hl=it&gl=US

https://apps.apple.com/gb/app/indeco/id1207350218?l=it

This program allows us to easily display the data received from the Indeconnect data logger device in graphs and tables. Below is an example of data visualization in this application. As you can see in Figure 2-18, each hydraulic hammer device has a dashboard that consists of different sections such as: Data Section, Location, Temperature, and Remaining Battery Charge, etc.

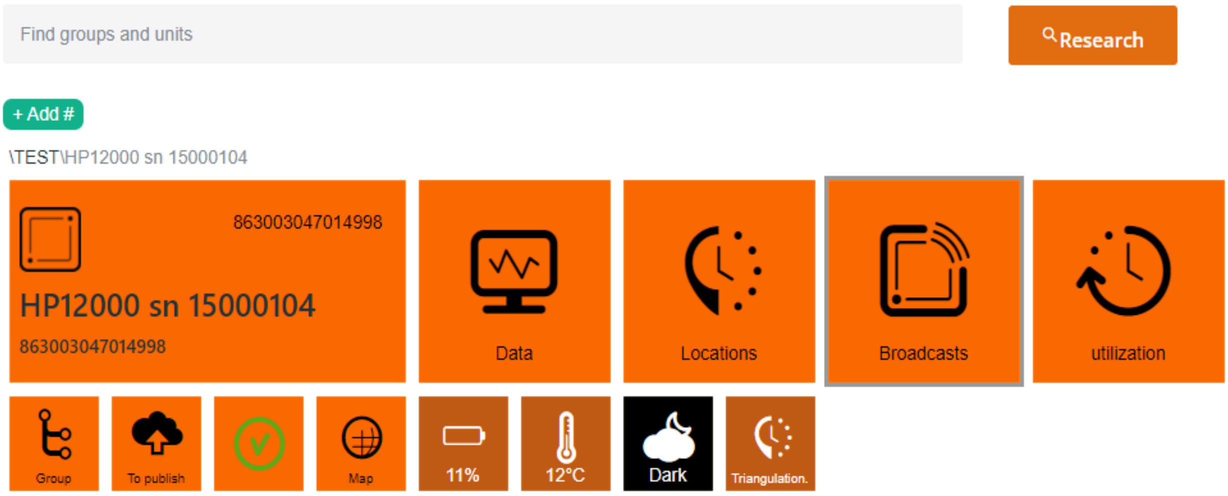


Fig 2- 18. On the Indecconect application dashboard, the data for the hydraulic hammer can be visualized.

By selecting the data section, you can access the data received from the sensors. Two examples of the sensors' performance can be seen in Figure 2-19. Due to the corona virus and the difficulty of the production process, the final product of the Indeconnect data logger has not been put into service at the time of writing, so the results of the pressure and flow sensor data are not available. However, the type and model of sensors, their requirements, and their locations have been discussed and described in detail in this thesis.

Fig 2- 19. Diagram of the accelerometer and the temperature sensor of the Indeconect data logger between January 2022 and March 2023.

In addition, in the section on utilization through accelerometer data analysis, the operating time of each hydraulic breaker can be easily viewed, and the results can be extracted as graphs and PDF files (see Figure 2-20).

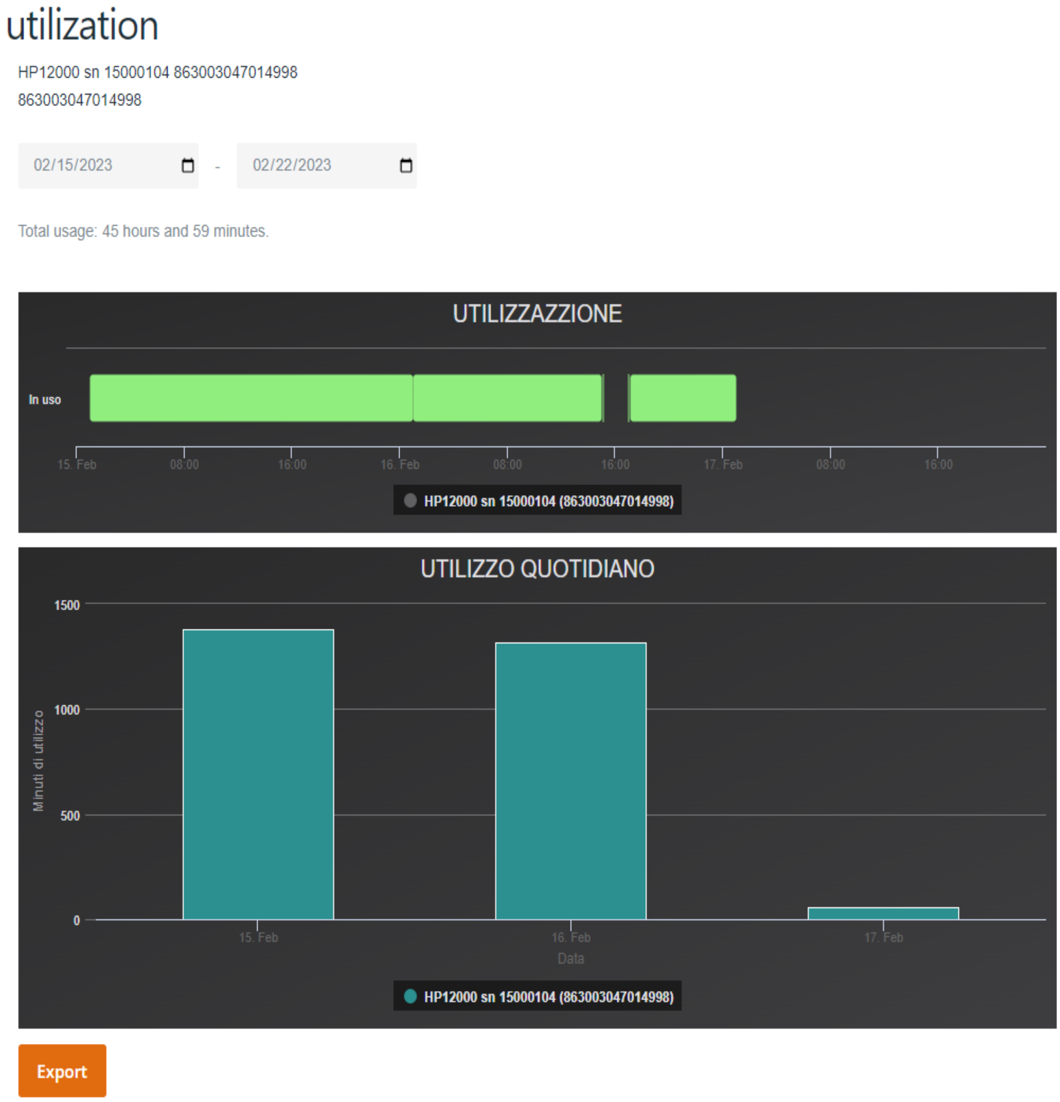


Fig 2- 20. charts of the running hours of the hydraulic hammer HP 12000 based on days and minutes.

2.6 Conclusion

The need for monitoring and controlling industrial processes, equipment, and machinery from a distance has increased rapidly due to advancements in technology and the availability of the Internet. Additionally, industrial devices are often used in challenging environments and are highly sensitive. As a result, a study was conducted to develop a data logger for remote monitoring of hydraulic hammers. This involved integrating sensors, meeting specific requirements, and utilizing platforms for data analysis and maintenance prediction, in collaboration with the Research and Development of INDECO Ind. SpA and Trustedglobal (Trusted A/S) company, all the required and effective parameters were determined by the test bench and the sensors were selected accordingly.

The Indeconnet data logger monitors key system parameters such as vibration rate, operating hours, pressure and flow intensity, oil temperature, and location. In modern equipment management, it is very important to determine the operating hours of the equipment. In fact, the operating hours are the actual usage and operation of machinery and equipment. Here are some of the most important advantages.

- By calculating the operating hours based on the actual usage of the machines and studying the maintenance standards of the equipment, the expected maintenance of the machines can be predicted.
- Avoid machine and critical equipment failures. Overuse of equipment is prevented by calculating actual operating hours.

The advantages of indeconnect are briefly listed below:

1. **Productivity monitoring**: Make sure each tool is working as planned.
2. **Control of operations:** Check in real time the different parameters internal and external to the equipment to make sure it is working in optimal conditions and appropriately.
3. **Greater security:** Remotely check the position of tools through the GPS geo-localisation. An actual anti-theft system.
4. **Maintenance:** Monitor the health of tools in real time, order spare parts and plan maintenance to minimise machine downtime.

In addition, data collection can help improve maintenance by using machine learning and deep learning algorithms to analyze the data collected and make predictions. Collecting data via sensors, blueprints, and observation by the operator can provide a more comprehensive view of a system or process, which can lead to more accurate analysis and better decision-making.

sensors can collect real-time data on various parameters such as temperature, pressure, and flow rate. This data can be automatically logged by Indeconnect data logger, providing a continuous record of the system's performance. This can help identify trends and anomalies that may be missed through manual observation alone. Also, bservation by the operator can provide valuable insights into how the system is actually being used in practice. This can help identify issues that may not be apparent from sensor data or blueprints alone, such as operator error or unforeseen circumstances.

As a result, by collecting data via sensors, blueprints, and observation by the operator, the Indeconnect data logger can provide a more complete picture of the system's performance, allowing for more accurate analysis and better decision-making.

Finally, the next section discusses the use of data analysis techniques and artificial intelligence algorithms, data visualization, statistical analysis, and finally comparing the results of different predictive maintenance models.

Chapter 3

Artificial Intelligence Algorithms for Fault Detection

3.1 State of the Art

This chapter describes the background and literature on various machine learning and deep learning algorithms for predictive models for maintenance. The preprocessing of data, feature engineering, feature selection, hyperparameters, data visualization and the results of each algorithm are examined. An innovative method to overcome the limitations of an imbalanced data set is also presented.

3.1.1 Introduction

Systems operating in industrial environments require a high level of availability and reliability. This applies to complex machinery used in processes such as production lines, oil and gas wells, and Hydraulic hammer facilities. Unplanned maintenance can result in lost production revenue, along with other consequences such as increased expenses for repairs and cleanup. In more severe cases, malfunctions can also pose risks to people and the environment. Reliability engineering has been employed for some time to minimize the likelihood of such failures occurring in these domains. About one-third of equipment maintenance expenditures in the United States are unnecessary and only drive up costs [38]. Thus, maintenance directly affects human resources and material consumption and is a major concern of the fourth industrial revolution [12].

In industry, four types of maintenance are generally distinguished: "reactive maintenance," "preventive maintenance," "predictive maintenance," and "predictive maintenance." One type of maintenance performed after equipment failure is called reactive maintenance [39]. Although reactive maintenance is one of the simplest and oldest maintenance methods, but large and global companies have completely abandoned this method due to its cost and reliability [40]. Figure 3-1 shows schematically the reactive maintenance. Then, in the early 1960s, preventive maintenance was explored, involving regular equipment inspection and maintenance [39]. This method reduces the risk of failure and increases the equipment's life. Today, reactive and preventive maintenance are less important for large and global companies due to low reliability and high maintenance costs.

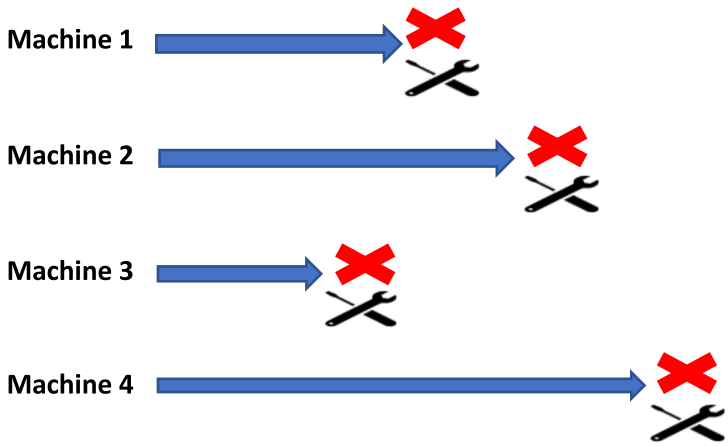


Fig 3- 1. Schematic representation of reactive maintenance

As part of preventive maintenance management, machine repairs or overhauls are planned on the basis of MTTF (Mean Time to Failure) statistics. The MTTF curve, also known as the bathtub curve, states that a new machine is likely to fail during its first few weeks of operation due to installation problems. After this initial phase, the probability of failure is relatively low for an extended period of time. After this normal life of the machine, the probability of failure increases rapidly over time. How can we determine the ideal time to maintain a system with numerous interdependent components so that no components need to be replaced prematurely and the system as a whole continues to function reliably, given that predictive maintenance cannot completely eliminate failures due to the complexity of the machine? Predictive maintenance (PdM), where we try to build predictive models based on observational data such as vibration, pressure, stress, equipment metadata, etc. Building predictive models that quantify the risk of failure of a machine at a given point in time and using this information to improve maintenance planning should provide an answer to this question.

In the 1980s, technological advances made it possible to monitor equipment and collect data. As a result, a new condition-based method called predictive maintenance was introduced [41, 42]. Also, the method reduces maintenance costs and increases reliability by accurately estimating equipment downtime and remaining useful life (RUL) [39].

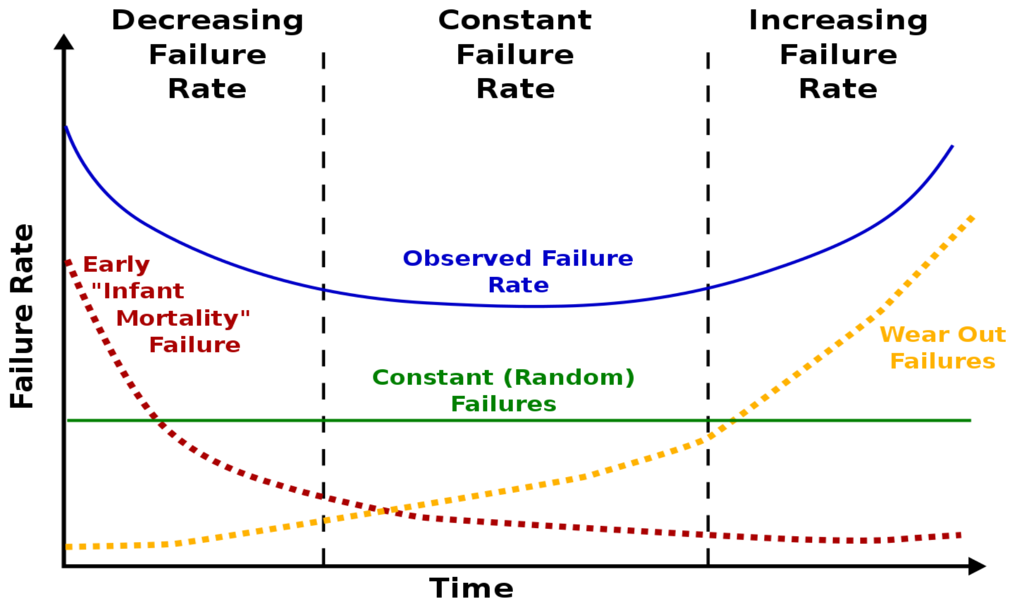


Fig 3- 2. Typical bathtub curve **.** The 'bathtub curve' hazard function (blue, upper solid line) is a combination of a decreasing hazard of early failure (red dotted line) and an increasing hazard of wear-out failure (yellow dotted line), plus some constant hazard of random failure (green, lower solid line) **[43]**.

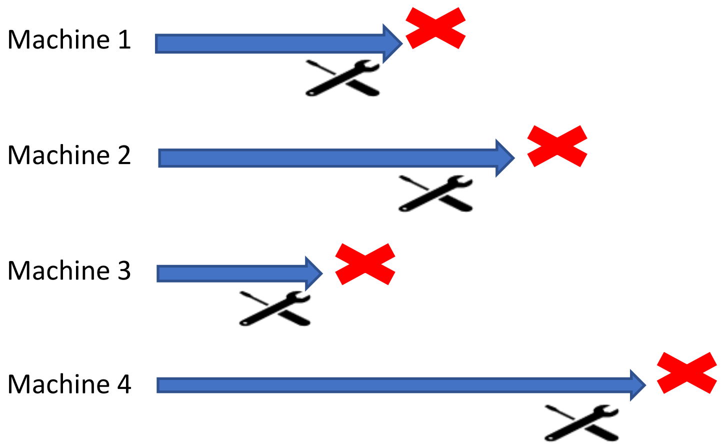


Fig 3- 3. Schematic representation of predictive maintenance.

Proactive maintenance is a maintenance method that investigates and identifies the causes of system failures and eliminates failure factors. Today, this method is highly valued in conjunction with predictive maintenance [44].

As a result, the Internet of Things (IoT) is playing a key role in industry due to its ability to continuously monitor machines and devices, collect data, and use it as a critical factor in evaluating system performance. In addition, predictive maintenance (PdM) analysis, which uses data-driven techniques such as machine learning and deep learning to predict quality in the manufacturing and industrial complex, is an essential tool [13, 14].

Therefore, artificial intelligence tools, particularly machine learning and deep learning, due to their great potential in creating automatic models for Big Data analysis, allow us to reduce repair and maintenance costs, maximize components' working lives, reduce machine downtime, increase performance and operational safety, and improve decision-making capabilities regarding the ideal timing and actions for machine maintenance [15, 16, 17].

Many articles work on PdM, which can be divided into three parts based on the system prediction method: physical, data-driven, and hybrid models [45] : Physical model approaches use prior knowledge of the system to create a mathematical description of the system degradation [46, 47, 48, 49]. The system concept (Physical meaning) is simple but challenging to execute when it is complicated.

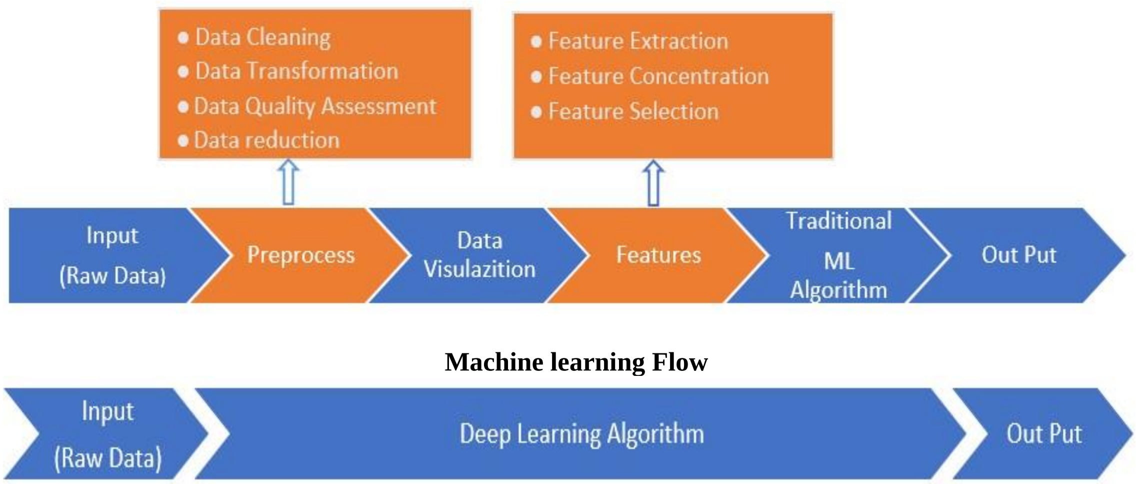
Data-driven methods use computational functions, algebraic rules, algorithms, and artificial intelligence methods, as well as state analysis and monitoring, where the solution is learned from historical data to predict the state of a system [50, 51, 52]. This method does not require understanding the operation of the system, which makes it suitable for complex systems. However, it is usually difficult to relate the results to the physical meaning.

The hybrid approach considers and combines both previous methods [45, 53]. The Industry 4.0 revolution in machine monitoring and continuous data acquisition is critical for data-driven methods, especially machine learning and deep learning which has opened the possibility of developing PdM models and increasing their accuracy [19]. The flowchart of machine learning and deep learning is shown in Figure 3-4.

As shown in Figure 3-5, the data-driven PdM system is divided into two phases. First, a learning procedure (i.e., training the model) using previous raw sensor data is required. The trained model is then used to anticipate goals and make decisions. Each step typically consists of one of the three sub-processes listed below:

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

To train models, traditional ML approaches such as logistic regression (LR), decision trees (DT), and random forests (RF) often require the collection of a significant amount of data from various failure events [54, 55]. The device state representation is then trained using features extracted from the time, frequency, and time-frequency domains. Deep learning , on the other hand, applied to a variety of neural networks (NNs), avoids the complex feature engineering described above and can be learned with an end-to-end learning strategy achieved by adding deep layers between the raw input and the prediction output. This is the main difference between ML and DL, and deep models can be considered as a "black box" that outputs the prediction result directly from the input. Because of all these factors, both ML and DL are widely used in PdM applications.



Deep Learning Flow

Fig 3- 4. Flow Chart of ML and DL.

In this work, a set of data, coming from different sources, available online [18] was evaluated. First, 10 of the most popular machine learning algorithms used in various works for PdM were applied. Machine learning algorithms used:

- Random Forest Classifier (RFC);
- eXtreme Gradient Boosting Classifier (XGB Classifier);
- Logistic Regression (LR);
- Extra Trees Classifier;
- Bagging Classifier;
- Support Vector Classifier (SVC);
- Linear Support Vector Classifier (Linear SVC);
- Stacking Classifier;
- Adaptive Boosting Classifier (AdaBoost);
- Decision Tree Classifier.

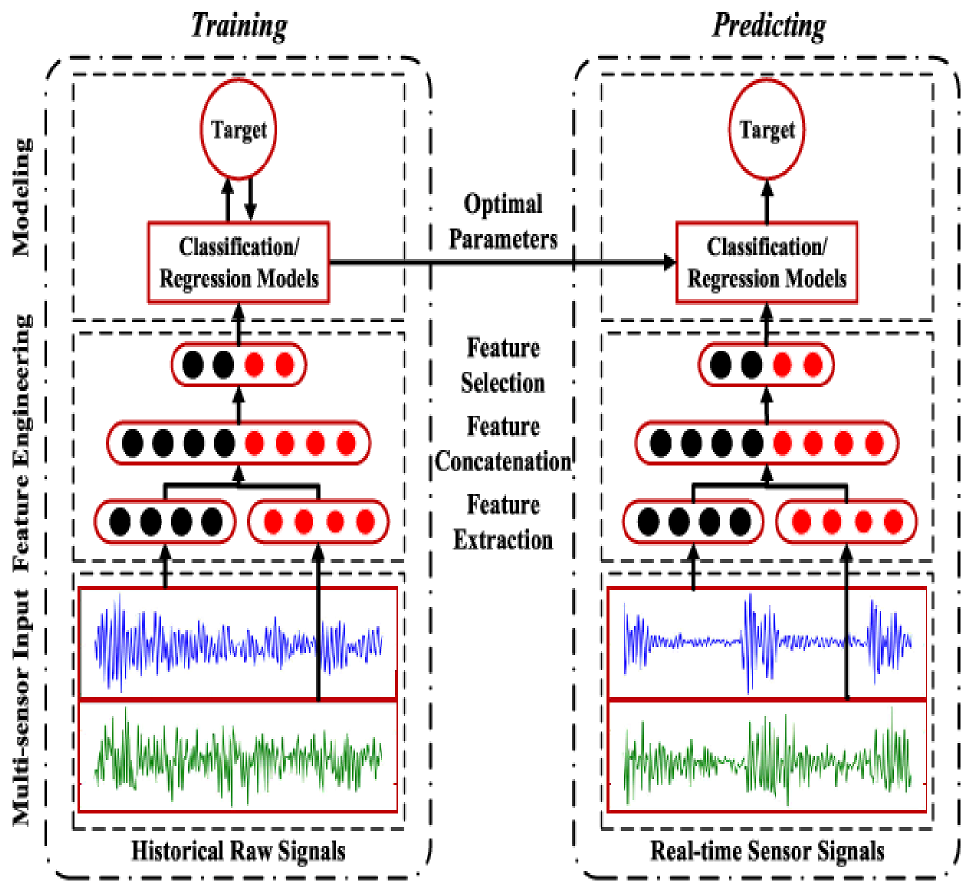


Fig 3- 5. Flowchart of the data-driven method for predictive maintenance (PdM) **[56]**.

Second, an innovative Deep Learning method was applied to the dataset. In this work, two types of Deep Learning algorithms are used:

- Convolutional Neural Networks (CNN);
- Long Short-Term Memory networks (LSTM);

First, the results of the Machine Learning algorithms and Deep Learning models were compared and discussed over the next 24 hours. In today's industrial landscape, the maintenance and repair of equipment is an essential aspect of ensuring the smooth functioning of industrial processes. However, it can also be a time-consuming and complex process that requires careful attention and planning. This is particularly true in cases where the equipment is located in remote areas, such as road construction machinery or hydraulic breakers. One of the biggest challenges of maintaining and repairing industrial equipment in remote areas is the procurement and supply of spare parts. In many cases, the parts required to repair the equipment may not be readily available in the local area, which can cause delays and disrupt the repair process. This can lead to prolonged downtime, decreased productivity, and increased costs. [19].

Predictive maintenance can help reduce the downtime of equipment by identifying potential failures before they occur. This allows technicians to plan and schedule maintenance activities in advance, thereby reducing the need for emergency repairs and minimizing the impact of equipment failure on operations. predictive maintenance can also help extend the lifespan of equipment in remote areas. Equipment in remote areas is often subject to harsh environmental conditions, which can lead to premature wear and tear. By identifying potential failures before they occur, predictive maintenance can help address issues before they cause permanent damage to the equipment, thereby extending its lifespan. In addition, the availability of spare parts and skilled technicians in remote areas can be a challenge. Predictive maintenance can help optimize the use of available resources by identifying the most critical equipment that requires maintenance and prioritizing repairs accordingly. This helps ensure that the limited resources available are utilized in the most effective manner. Therefore, we improved PdM models with traditional learning machines and Deep Learning algorithms for up to 7 days and compared the results.

3.1.2 Challenges

With technological advances in Industry 4.0, sensors, and the Internet of Things, a new concept called condition-based maintenance (CBM) has been considered. In this approach, inspections performed by engineers and technicians are automatically performed by tools and devices that measure industrial physical parameters such as flow, vibration, pressure and temperature signals [57].

As a result, interference and measurement errors are much lower, and by defining the operating range for each sensor, necessary action is taken when system performance is reported outside that range. Although a more effective and sophisticated approach CBM is used in the industry today, such as predictive maintenance, which uses data from sensors, cyber-physical systems , maintenance and fault reports [58]. It analyzes the process using machine learning and deep learning algorithms. It predicts the failure and remaining useful life (RUL) of machines and components and schedules maintenance and replacement of components [59].

One of the biggest challenges in predicting maintenance considering harsh industrial environments is the integrated collection of industrial data. In industrial environments, the likelihood of sensor errors and noise in the data is high [60]. In addition, the amount of data in industrial equipment monitoring is very large, so it is necessary to develop a suitable architecture for real-time processing of data sets for data analysis. Another challenge in maintenance forecasting is the collection of information on equipment replacement and repair by technicians.

This portion of the data is typically gathered by operators, resulting in a higher likelihood of errors. Furthermore, the maintenance forecast dataset is intrinsically imbalanced. Machines typically experience failures in less than 2% of their lifespan, meaning that the data collected from machines mainly consists of 98% of regular machine operations and less than 2% of instances of failure or errors. This imbalance disrupts the algorithm's training and significantly impacts the eventual outcomes.

3.2 Data Set

Despite the growth of IIoT and sensors, access to datasets is difficult and rare due to security and business competition. However, Microsoft has published maintenance data from a large industrial project as a dataset [18]. The dataset consists of five subsets:

- real-time telemetry;
- error log;
- maintenance history;
- fault history; and
- Metadata of Machines

This dataset refers to 100 machines monitored in real-time for one year (between 2015-01-01 and 2016-01-01) with four pressure, stress, vibration, and rotation sensors. Each machine has four main parts that need repair and maintenance. 876.100 hourly telemetry records were collected for the machines. The number of fault records is 3919, and the number of maintenance records is 3286. In addition, five types of errors and a total of 3920 errors were recorded for the machines.

As shown in Table 3-1 , the sensors monitor all four parts of each machine in real-time and report the average of the measurements for each hour. The implementation described in this chapter is carried out in Python programming language using the Matplotlib, Numpy, Pandas, and Scikit-Learn packages [61, 62].

Table 3- 1 An example of a real-time telemetry recording.

	Datetime	**Machine ID**	**Volt**	**Rotate**	**Pressure**	**Vibration**
15429	2015-06-10 00:00:00	2	166.553160	442.727933	115.759418	41.81844021
15430	2015-06-10 01:00:00	2	183.765522	293.171668	114.854098	35.66955239
15431	2015-06-10 02:00:00	2	210.505958	403.550754	102.978774	42.85403616
15432	2015-06-10 03:00:00	2	180.469035	484.030160	88.0963987	50.50278907
15433	2015-06-10 04:00:00	2	188.328311	441.091774	90.4839496	42.02409627

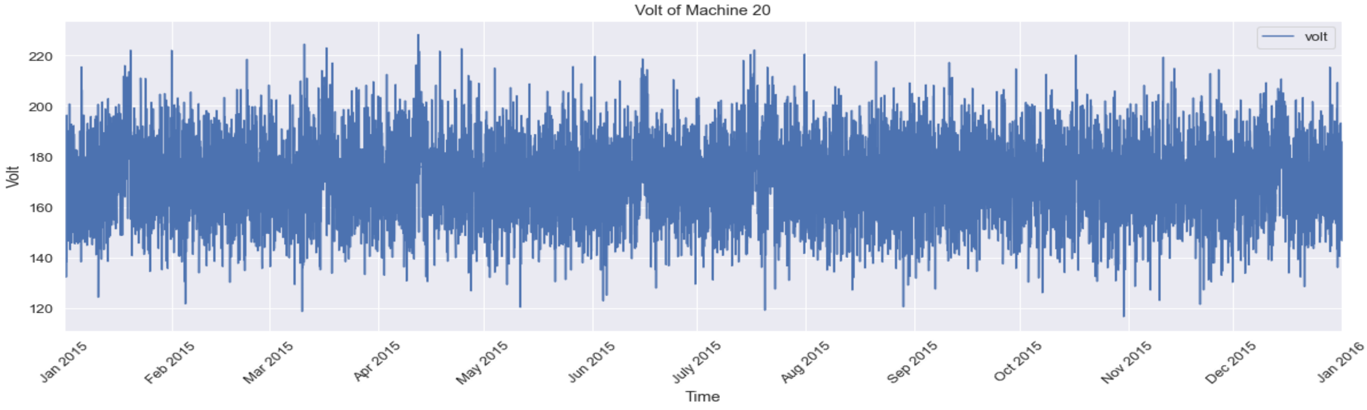
3.2.1 Exploratory data analysis of the telemetry

Exploratory Data Analysis (EDA) is a type of data analysis in which an analyst investigates, summarizes, and visualizes data to gain insights, understand patterns, and uncover relationships. It is an iterative process that involves cleaning and transforming the data, identifying trends, and creating visualizations to communicate findings. EDA is an important step in the data science process, as it helps to develop hypotheses, validate assumptions, and identify potential areas for further analysis.

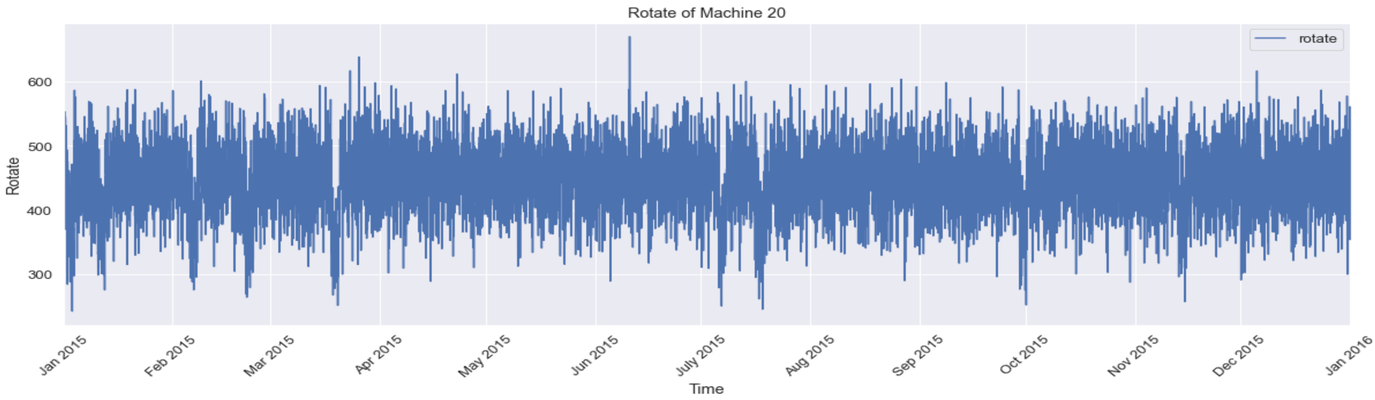
The automatic exchange of data from many sources is called telemetry. In 2015, hourly averages of voltage, rotation, pressure, and vibration were collected from 100 machines. For a better understanding of the behavior of each sensor, a simple statistical study is performed in Table 3-2. Where in it, the parameters for voltage ("Volt"), rotation ("Rotate"), pressure ("Pressure") and vibration ("Vibration") between 01/01/2015 and 01/01/2016 are given, as well as their mean values, standard deviations, minimum and maximum values. As an example, Figure 3-6 shows the graphical evolutions of the voltage (Figure 3-6a), rotation (Figure 3-6b), pressure (Figure 3-6c), and vibration (Figure 3-6d) for machine 20 (machineID =20)

Table 3- 2 Statistical analysis of telemetry data in real time.

	count	mean	std	min	25%	50%	75%	max
MachineID	876100	50.5000	28.8660	1.0000	25.7500	50.5000	75.2500	100.0000
volt	876100	170.7777	15.5091	97.3336	160.3049	170.6073	181.0044	255.1247
rotate	876100	446.6051	52.6738	138.4320	412.3057	447.5581	482.1766	695.0209
pressure	876100	100.8586	11.0486	51.2371	93.4981	100.4255	107.5552	185.9519
vibration	876100	40.38500	5.37036	14.8770	36.7772	40.2372	43.7849	76.791

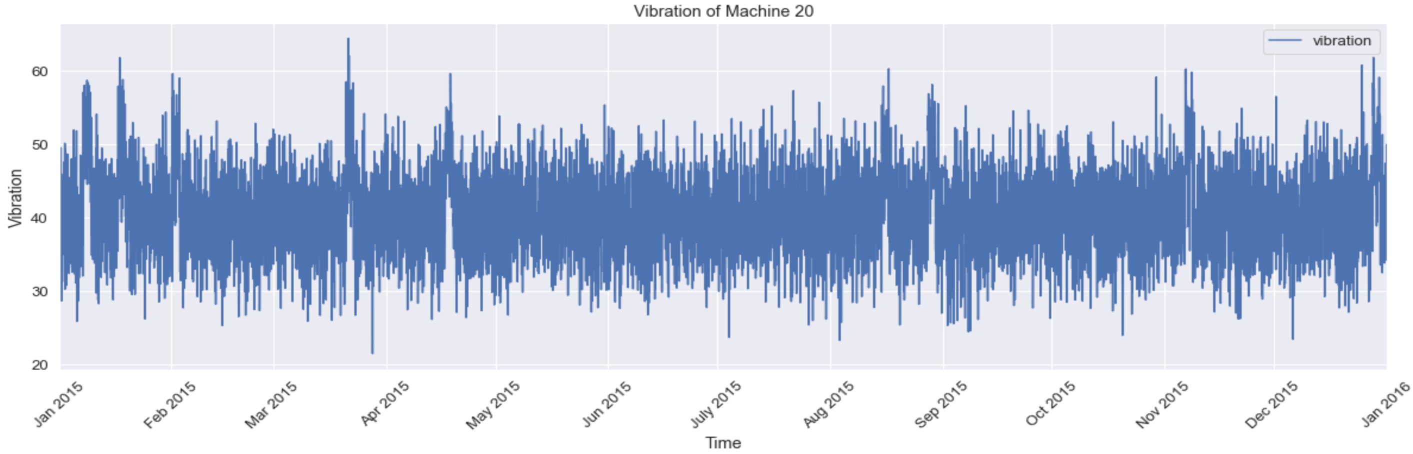


(a) Volt Signature



(b) Rotate Signature

(c) Pressure signature



(d) Vibration signature

Fig 3- 6. Evolution of Telemetry data for the machine 20.

The voltage of the machines does not change during the month, as shown by the plot of the voltage distribution over the different months (Figure 3-7). Since we only have data for one day in 2016, we can discard the 2016 entry.

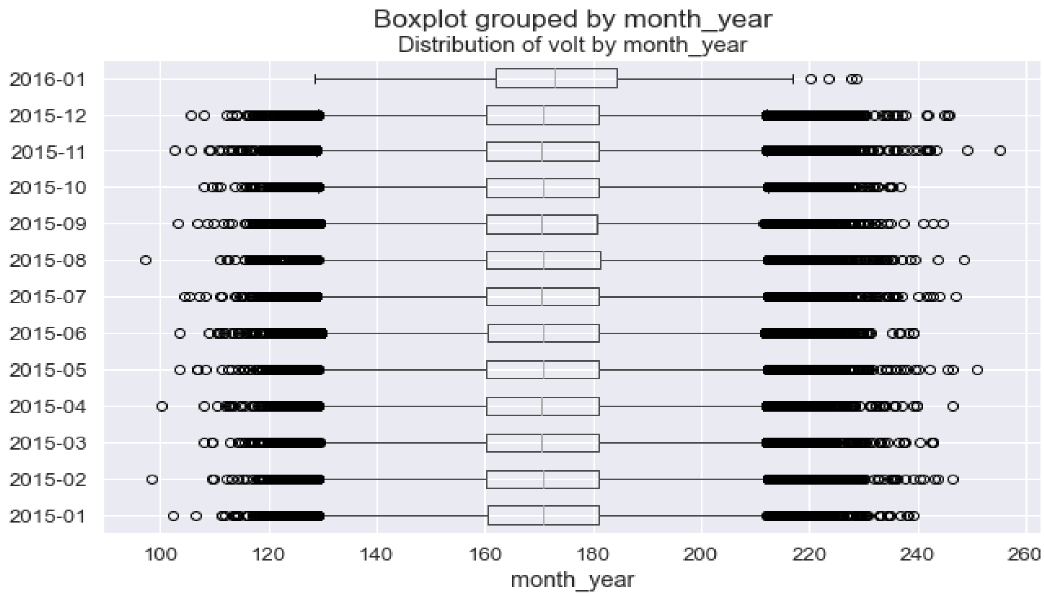
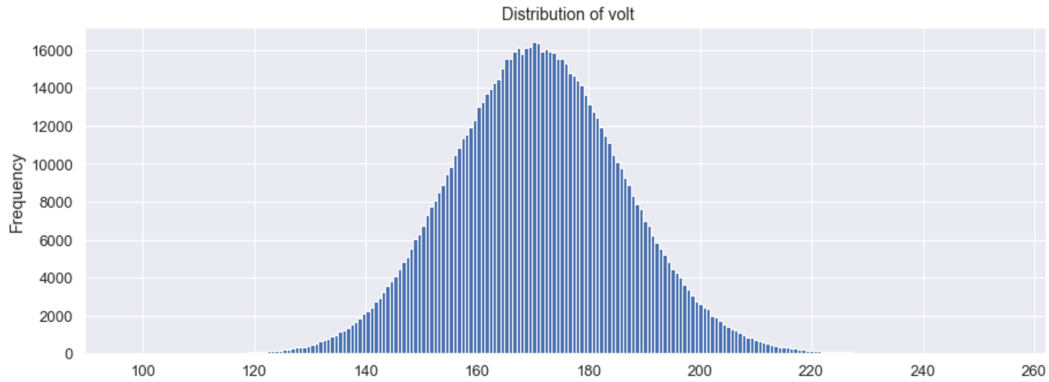
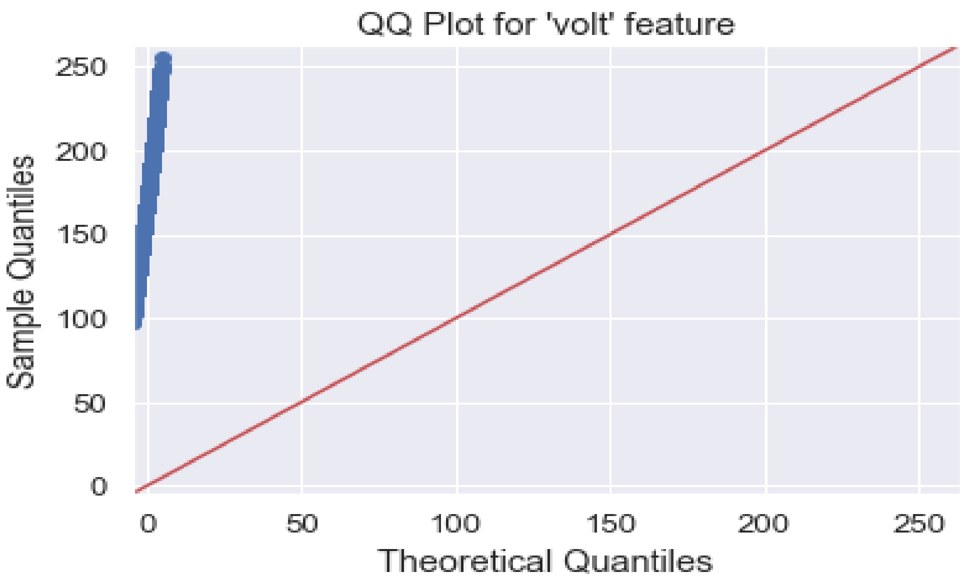


Fig 3- 7. plot the distribution of voltage across various months.

To verify the voltage distribution over the machines, the plots of the normal distribution were examined (Figure 3-8a). The voltage distribution resembles the normal distribution, but for further investigation, the Anderson-Darling test and the quantile-quantile (QQ) plot are required (Figure 3-8b).



(a)



(b)

Fig 3- 8. QQ and distribution of Voltage plots across Machines**.**

The distribution of vibration, rotation, voltage, and pressure appears normal. However, after applying the Anderson-Darling test and reviewing the QQ plots, it was found that the data for rotation, pressure, voltage, and vibration data do not belong to the normal/Gaussian distribution.

3.2.2 Exploratory data analysis of the error

The error logs contain errors that do not immediately cause the system to fail, but the investigation showed that the failure occurred shortly after the errors in many cases. These are operational errors experienced by machines that do not result in machine shutdowns and are recorded with rounded time stamps due to the hourly telemetry data collection. This system contains five types of errors: error 1, error 2, error 3, error 4, and error 5. See Table 3-2 for an example of a documented error.

Table 3- 3 An example of a errors recording.

	Datetime	**Machine ID**	**ErrorID**
0	2015-06-10 00:00:00	1	Error2
1	2015-06-10 01:00:00	1	Error1
2	2015-06-10 02:00:00	1	Error2
3	2015-06-10 03:00:00	1	Error3
4	2015-06-10 04:00:00	1	Error3

The histogram of errors and distribution of errors based on the Machine types is displayed on Figure 3-9. It can be concluded from reviewing the diagrams and analyzing the error data:

1. Type 1 and Type 2 errors are the most frequent, occurring more than double the number of Type 5 errors.
2. Machine ID 22 has the highest number of errors, approximately 60, with the highest frequency of Type 4 errors (15 occurrences). The lowest frequency of errors was Type 5 with 9 occurrences.
3. An average of 12 errors per day occur across the 100 machines.

4. The highest number of errors (less than 25) occurs across days. In 2016 and 2015, only one error occurred on a given day.
5. An average of 12 errors per day occur across the 100 machines.

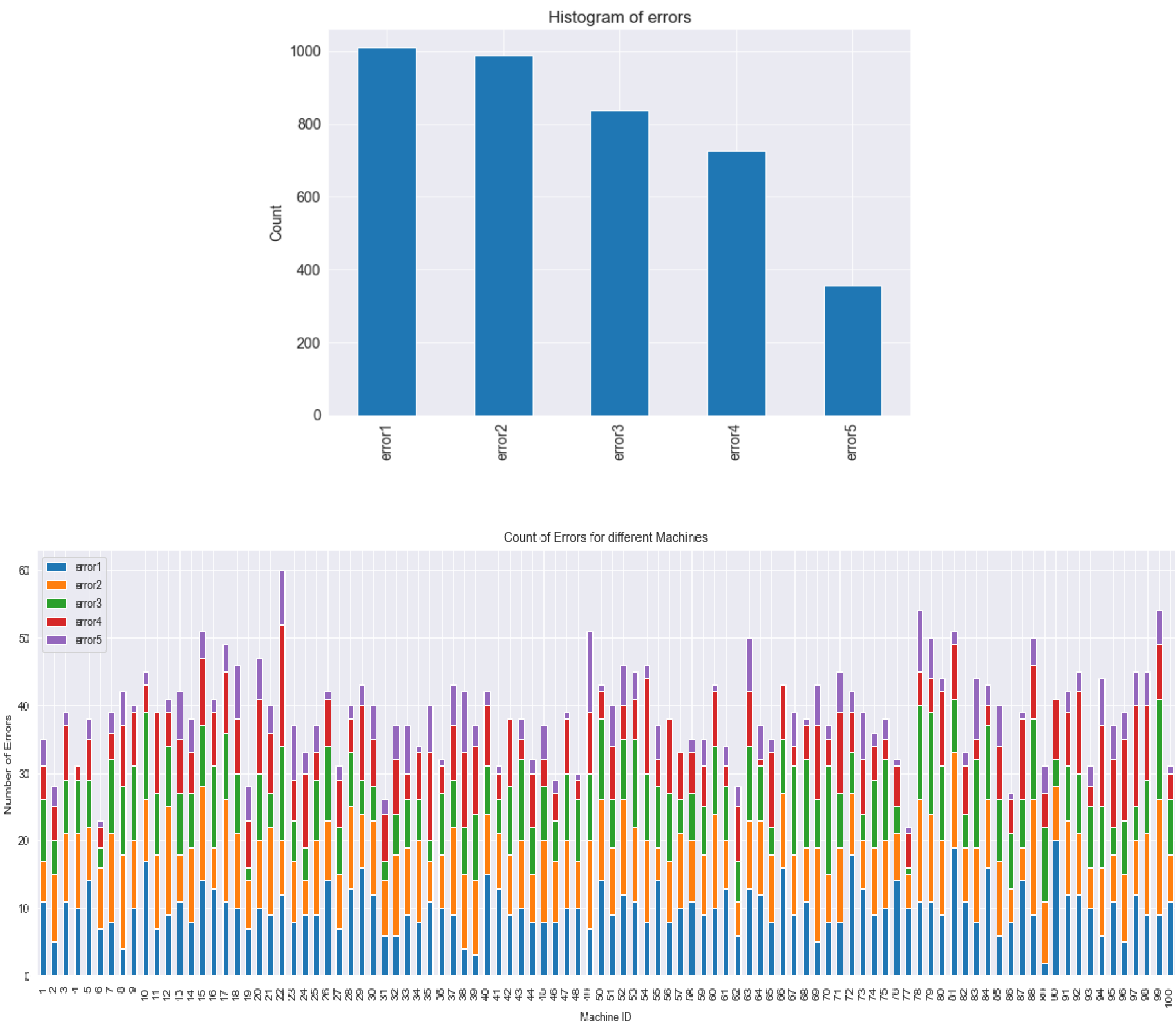
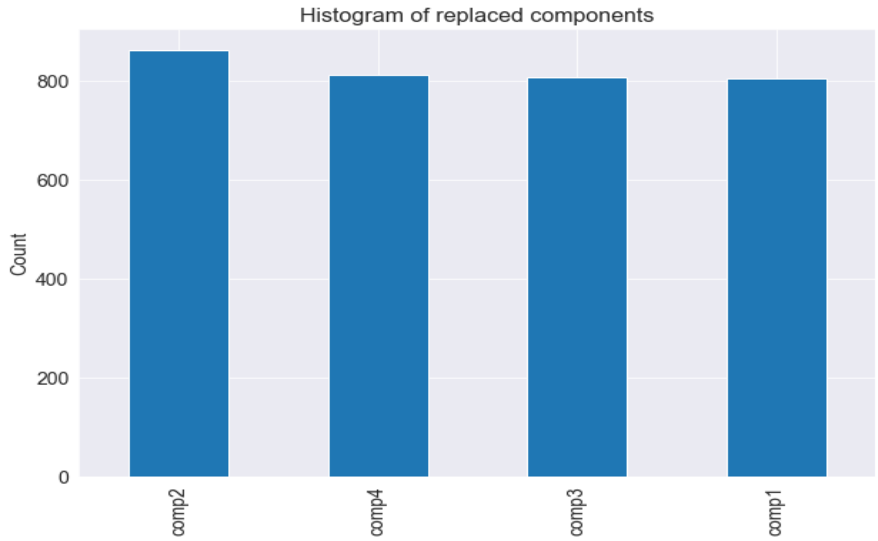


Fig 3- 9. The histogram of errors and distribution of errors based on the Machine types.

3.2.3 Exploratory data analysis of the Maintenance

The dataset also includes records of maintenance, component replacements, and regular and unscheduled inspections. In the case of maintenance performed due to failure, there is an entry in the log. Each machine has four parts: comp1, comp2, comp3, and comp4. Figure 3-10 indicates that component replacements for all four parts of the machines were almost equal in 2015.



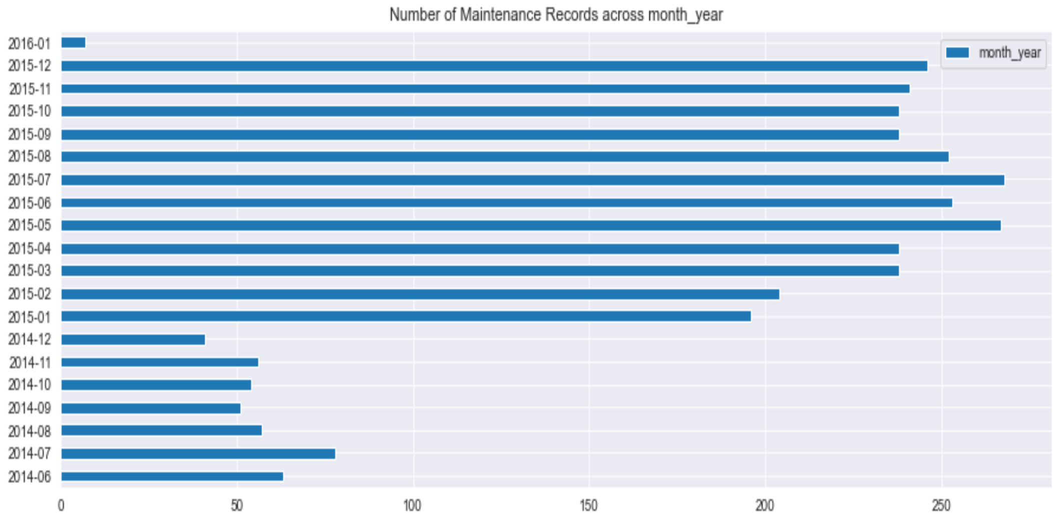


Fig - 10. Histogram diagram of replaced components by type and number of maintenance records across months.

It can be concluded from reviewing the diagrams and analyzing the maintenance data:

1. All four types of components were replaced approximately 800 times each.
2. Maintenance records are available from June 2014 to January 2016.
3. The number of components replaced in 2015 was significantly higher compared to 2014.
4. In 2015, the highest number of maintenance records were in the months of May and July.
5. The data for 2016 can be disregarded since there is only one day's worth of data.
6. Machine ID-66, 68 & 70 are the highest number of Maintenance Records machines.

3.2.4 Exploratory data analysis of the Machines

The dataset contains information about the machines, such as model type and years in operation. The histogram of the machines' metadata is shown below.

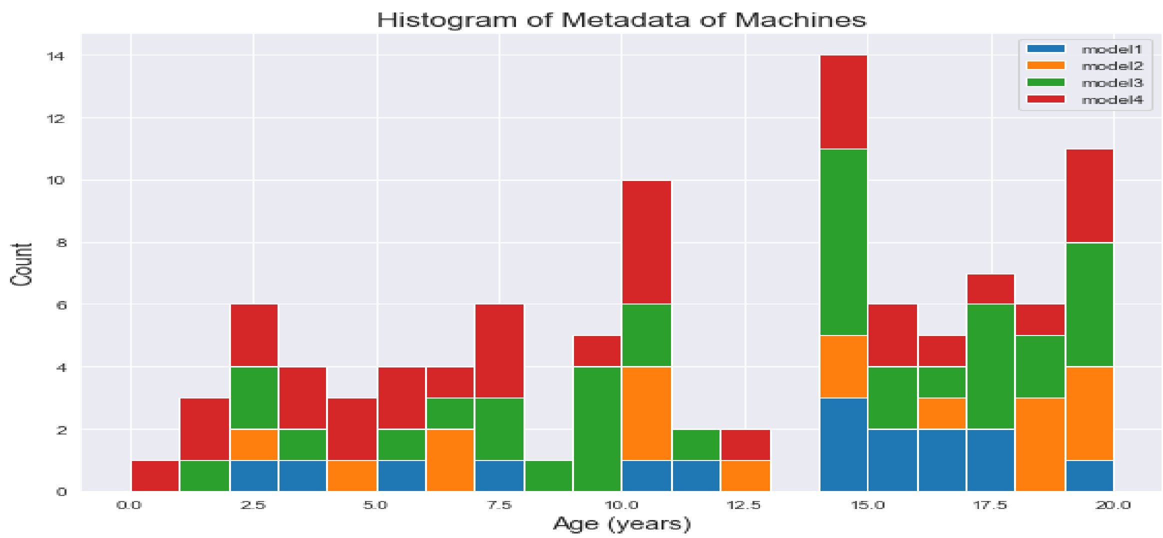


Fig 3- 11. Histogram of metadata of machines, by model.

The analysis of the data suggests that the number of failures is slightly correlated with the age of the machine. The information from Table 13 and the correlation values supports the conclusion that the number of failures is linked to both the number of errors and the machine's age.

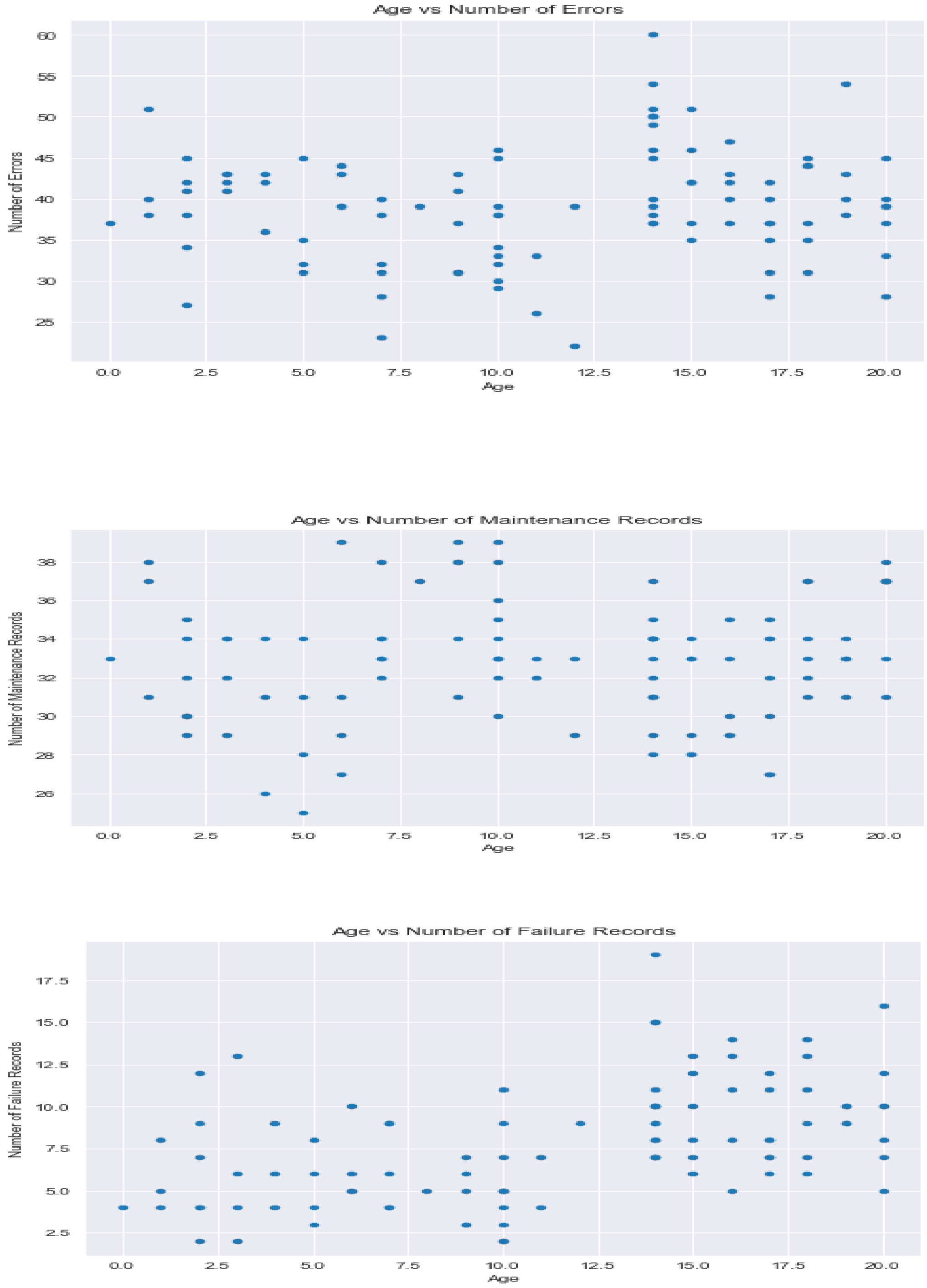


Fig 3- 12. plots of the number of failures, errors, and maintenance logs over the life of the machine.

Table 3- 4 correlation of the different parts of the data to each other.

	machineID	**age**	**num_errors**	**num_maint**	**num_failure**
machineID	1.000000	0.100196	0.107982	0.077903	0.096496
age	0.100196	1.000000	0.106931	0.075445	0.476459
num_errors	0.107982	0.106931	1.000000	0.026558	0.483735
num_maint	0.077903	0.075445	0.026558	1.000000	-0.030258
num_failure	0.096496	0.476459	0.483735	-0.030258	1.000000

3.2.5 Exploratory data analysis of the Failures

This maintenance data describes the replacement of components due to failures and is collected on an hourly basis. The data has been rounded to the nearest hour, indicating that it is a summarized version of the raw data. The maintenance data is a subset of the larger data set, which likely contains additional information about component maintenance and repair.

Understanding the correlation between the different parts of the maintenance data can provide valuable insight into the causes of component failures and inform preventive maintenance strategies to reduce the frequency of replacement. The number of records due to failures for all four machine components in 2015 was 761. The histogram of component failures is shown on Figure3-13.

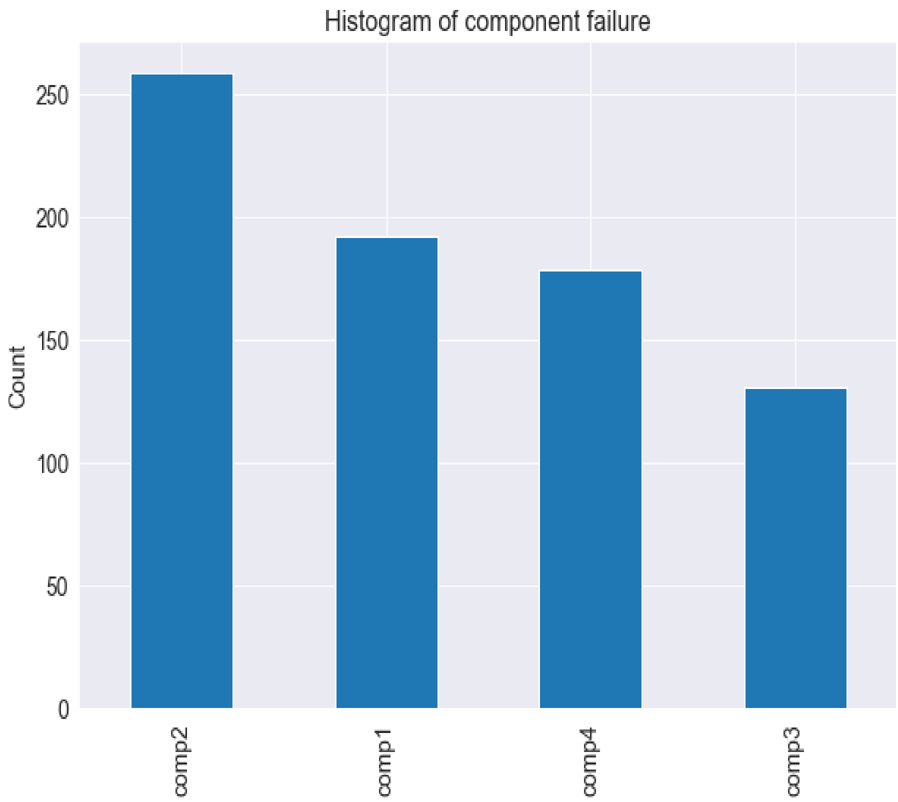


Fig 3- 13. Histogram diagram of failures records.

The histogram in Figure 3-13 provides valuable information about the frequency of component failures. The data shows that component-2 experiences the most failures, with the number of failures more than double those of component-3. This highlights the importance of focusing on understanding the root cause of component-2 failures in order to minimize equipment failures overall.

By addressing the root cause of component-2 failures, it may be possible to reduce the frequency of replacements and improve the reliability of the equipment. This information can be used to inform maintenance strategies, such as regular inspections or preventative maintenance procedures, that can reduce the likelihood of future component failures.

3.3 Analytics Methodology

Predictive maintenance (PdM) models are machine learning and deep learning algorithms that use data from various sensors, such as temperature, pressure, vibration, and others, to predict equipment failures. The goal of PdM is to avoid unscheduled downtime and reduce maintenance costs by only performing maintenance on equipment when it is actually needed, rather than on a predetermined schedule. PdM models can be either supervised or unsupervised, and can be built using techniques such as regression, classification, or clustering. The quality of a PdM model depends on the availability and quality of the data used to train it, as well as the choice of algorithms and hyperparameters.

Here are several challenges and problems associated with predictive maintenance (PdM) models:

1. **Data quality:** The accuracy of PdM models depends on the quality of the data used to train them. If the data is noisy or incomplete, the model may not accurately predict equipment failures.

2. **Data availability:** PdM models require large amounts of data to train effectively. If data is not available for all equipment or for all possible failure modes, the model may not be able to make accurate predictions.

3. **Overfitting:** Overfitting occurs when a model is too closely fit to the training data and does not generalize well to new data. This can be a problem for PdM models, as it can lead to false positive predictions.

4. **Model interpretability:** Some machine learning models, such as neural networks, can be difficult to interpret. This can make it challenging to understand why a model is making a particular prediction, and to identify potential biases in the data.

5. **Data privacy and security:** PdM models often require access to sensitive equipment data, which can pose privacy and security concerns. Ensuring the protection of this data is critical to the success of a PdM program.

6. **Cost:** Implementing a PdM program can be expensive, as it requires investment in hardware, software, and personnel. Additionally, the cost of maintaining and updating the models over time must also be considered.

For predictive maintenance analysis, we need to create a framework that fits the data set and algorithms. we consider the goal variables and outcomes, the predictive variables, and the algorithms for model building and model validation. Based on this, we need to convert the dataset into a machine-learning-friendly format. In this study, the rows represent the examples of the dataset to be predicted or learned from, while the columns represent the target and predictive variables. This framework is one of the most common formats for implementing various machine learning and deep learning algorithms.

By analyzing the data set and looking at predictive maintenance from an industry perspective, we have established two target variables. The first step is to determine the prediction of machine failures. In this case, we can predict machine failures only 24 hours and 7 days in advance. Since each machine consists of four different components, the objective of the second step is to determine which component of the machine will fail. As for the timing, the predictions were studied in two time periods: in the next 24 hours and in the next 7 days. In this research, 10 common and standard machine learning algorithms and two deep learning methods, CNN and LSTM, were used.

The target variable chosen plays a key role in the size of the training data set and affects the performance of the predictive models. A larger dataset usually leads to better models because better predictive models can be trained and more iterations are possible [63]. With the above goals in mind, our task is to determine/predict which machine and which component will fail in the next 24 hours and 7 days. A closer look at the dataset shows that we are dealing with an unbalanced dataset; for example, 98.11% of the data (Table 3-5) fall into the "Stable" category (failure = none), meaning that there were no failures.

Table 3- 5 Example of the imbalance between the different classes for the ‘failure’ feature in the total data set.

	Failure	%
none	285,684	98.06
Comp1	1464	0.50
Comp2	1985	0.68
Comp3	968	0.33
Comp4	1240	0.43

Dealing with imbalanced datasets is a common challenge in predictive maintenance (PdM) applications. An imbalanced dataset is one where the number of instances belonging to the minority class (equipment failures) is much smaller compared to the majority class (normal operations).

This imbalance can result in several problems for PdM models:

- **Biased models:** Models trained on imbalanced datasets can be biased towards the majority class and under-predict the minority class, leading to a high rate of false negatives.

- **Inefficient use of data:** The majority class may dominate the training data, which can lead to the model not effectively learning the patterns in the minority class.

- **Unreliable performance metrics:** Common metrics such as accuracy can be misleading when evaluating models trained on imbalanced datasets, as they do not take into account the imbalance in the data.

There are several methods to address class imbalance in PdM datasets, including:

- **Undersampling:** This involves randomly removing instances from the majority class to balance the classes.

- **Oversampling:** This involves randomly replicating instances from the minority class to balance the classes.

- **Synthetic oversampling:** This involves generating new synthetic instances of the minority class using methods such as SMOTE (Synthetic Minority Over-sampling Technique).

- **Weighted loss functions:** This involves assigning different weights to the loss function for each class, to emphasize the importance of correctly classifying instances from the minority class.

In this study, for short-term behavior, a 3-hour time window is used, for long-term behavior a 24-hour time window, but for seven-day forecasting, a different time frame is needed that covers more than 24 hours. The issue is that this results in some failure data being ignored and that this can negatively impact the accuracy of machine learning algorithms when training them.

Due to the imbalance of the data, methods for balancing datasets such as RUSBoost [64] and SMOTEBoost [65] have been studied in Python language programming.

The term "RUSTBoost imbalance dataset" likely refers to a dataset that has an imbalanced class distribution, meaning that the number of samples in different classes is not equal. In such a dataset, one class may have significantly more samples than others. This can cause problems in machine learning as algorithms may tend to predict the majority class more often, leading to poor performance on the minority class. The "RUSTBoost" part of the term could refer to a method for handling imbalanced datasets, such as the "Random Under-Sampling and Synthetic Over-sampling Technique (RUSBoost)" algorithm.

SMOTEBoost is an ensemble learning algorithm for binary classification problems that combines the Synthetic Minority Over-sampling Technique (SMOTE) with boosting. SMOTE is a data augmentation method used to balance the class distribution of an imbalanced dataset by synthesizing new samples for the minority class. Boosting is an ensemble learning technique that trains multiple weak learners in a sequential manner, where each subsequent learner focuses on the misclassified samples from the previous learner. By combining these two techniques, SMOTEBoost aims to improve the accuracy of classifiers on imbalanced datasets by balancing the class distribution and reducing overfitting.

SMOTEBoost and RUSBoost are two popular ensemble techniques for addressing class imbalance in machine learning tasks. However, these techniques may not work well for time-series datasets in predictive maintenance tasks due to several reasons [64] [65] :

- **Bias towards the majority class:** Even after over-sampling the minority class and under-sampling the majority class, the model may still exhibit a bias towards the majority class due to the larger number of samples.

- **Overfitting:** Over-sampling the minority class may lead to overfitting, especially if the number of synthetic samples generated is large.

- **Loss of information:** The under-sampling step in RUSBoost discards a portion of the majority class samples, which may result in a loss of important information. The synthetic samples generated in SMOTE may not accurately reflect the real-life distribution of the minority class, which can also result in a loss of information.

- **Difficulty in selecting appropriate parameters**: The choice of the ratio of samples to be under-sampled and over-sampled, as well as the choice of the over-sampling method, can greatly affect the performance of these algorithms, and finding the optimal combination of parameters can be challenging.

- **Temporal dependencies:** In time-series data, observations are ordered by time, and there is often a temporal dependence between observations. This means that oversampling or resampling techniques that randomly duplicate or remove observations can break the temporal dependence and lead to unrealistic data distributions.

- **Limited data availability:** In predictive maintenance tasks, data is often scarce, and collecting more data is expensive or impossible. Oversampling and resampling techniques require duplicating or removing observations, which can be problematic when data is limited.

- **Potential data leakage:** Oversampling and resampling techniques may introduce data leakage, where information from the future is used to predict the past. This can lead to overestimating the performance of the model during testing.

- **Outlier removal:** Resampling techniques such as undersampling can remove rare events that are critical for predictive maintenance tasks. These rare events may provide important insights into the health of the system and their removal may hinder the performance of the model.

These problems can be mitigated by using techniques such as cross-validation and hyperparameter tuning to find the optimal parameters, and by combining these algorithms with other techniques to get better results. In addition, Studies have shown that these methods affect the time factor in the dataset, so they did not improve the performance of machine learning algorithms results. In this study, two strategies are used to solve the problem of predictive maintenance models for more than 24 hours:

1. For predictions longer than 24 hours, machine learning algorithm results are given as a weighted average. As mentioned earlier, machine learning algorithms perform poorly on imbalanced datasets and biased class data. However, when training algorithms on unbalanced datasets with skewed distributions, we can apply the concept of weighting to improve this problem. In this case, the weighting difference is used to classify the classes in the training models.

   The goal is to correct the misclassification of the minority class by increasing its weight and decreasing that of the majority class. It should be emphasized that the weighting of each class must be applied appropriately and carefully, since a high weighting of the minority class (in this dataset, the failures) risks that the algorithm toward the minority class and increases the errors of the majority class. Fortunately, we can make appropriate and optimal use of minority class weighting by integrating modeling libraries such as sklearn, catboost, and LightGBM into Python scripts.

2. Predictive maintenance utilizes deep learning and machine learning as key techniques to forecast equipment failures and perform maintenance in a timely manner. This helps to minimize downtime and prolong equipment lifespan. Deep learning trains artificial neural networks on large data sets to identify patterns and make predictions, particularly in condition monitoring by examining sensor data for anomalies. On the other hand, machine learning uses algorithms to analyze data and make predictions.

   In predictive maintenance, machine learning algorithms are used to study patterns in historical equipment maintenance data to predict future failures. Both of these techniques offer valuable insights for improving equipment reliability and reducing maintenance costs. Figure 3-14 shows the differences between artificial intelligence, machine learning, and deep learning.

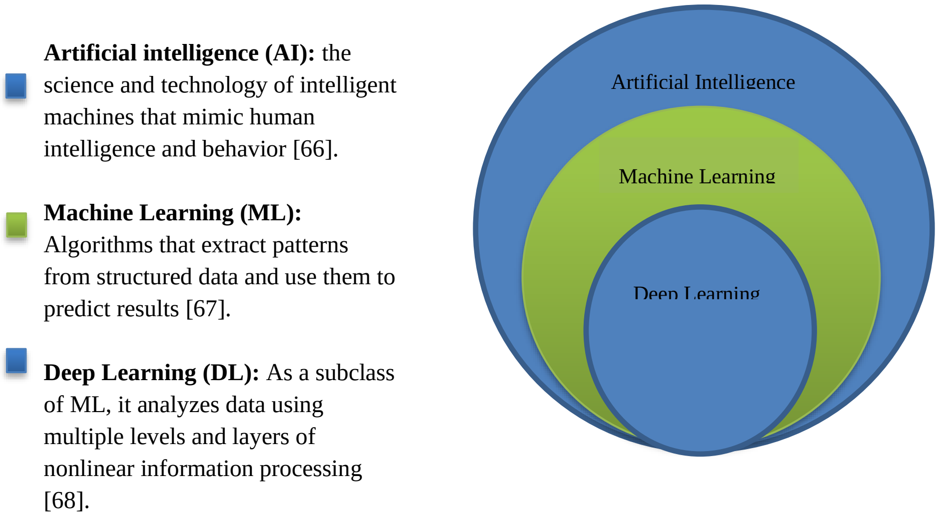


Fig 3- 14. Onion diagram for AI, ML, DL .

In this study, LSTM networks and CNN algorithms were employed to predict maintenance operations for the next 7 days based on the dataset. However, due to limitations in the data, a two-step approach was taken in which the dataset was divided into two classes. The two-step technique for deep learning predictive maintenance involves using two separate steps to analyze and predict equipment failures. The first step only analyzed the data between the two classes using artificial intelligence algorithms, showing the probability of failure for each machine in the next 7 days. In the second step, the data is further divided and analyzed to improve the accuracy of the prediction, taking into account the prediction errors from the first step. This two-step approach is used to ensure more accurate predictions and to maximize the efficiency of the predictive maintenance process. In the initial step, classes 0 to 3 linked with components 1 to 4 were merged into one class. Only AI algorithms were employed in this step to study the data between the two classes, and the results only displayed the likelihood of failure for each machine in the subsequent 7 days. The division of the dataset into these two classes is depicted in Figure 3-15.

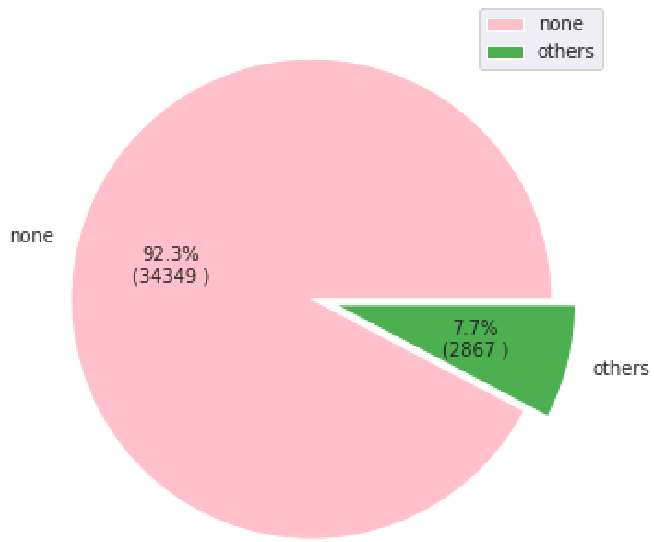


Fig 3- 15 Amount and percentage of data divided between classes 1 and 2.

The maintenance department's work is time-sensitive, so accurately predicting and identifying the component in need of maintenance is crucial. This allows the maintenance team to obtain the necessary resources and support for repairing or replacing the component. In the second step, the algorithms are trained, tested, and validated using the hourly and daily (windowing) data from the second class and the prediction errors from the first class. Windowing involves dividing the time-series data into fixed-length windows and treating each window as a separate instance. This technique can help to increase the number of instances in the failure or maintenance class and improve the balance of the dataset.

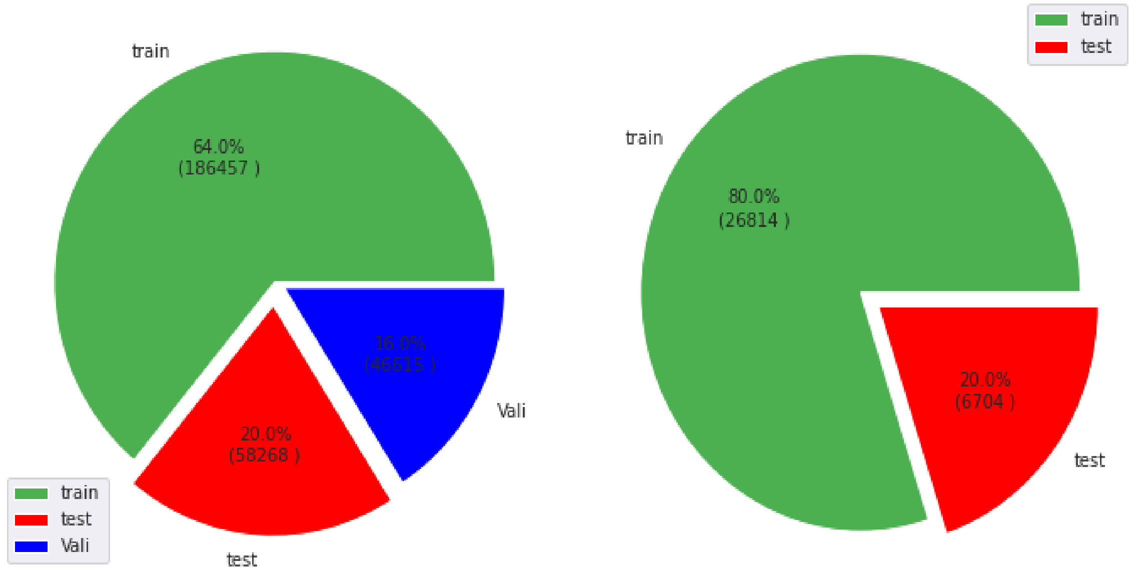
3.3.1 Data partitioning

Data partitioning is the process of dividing a dataset into separate subsets for training, validation, and testing. The goal of data partitioning is to create a more accurate and reliable machine learning model by preventing, overfitting and ensuring that the model is generalizable to new, unseen data. Splitting the data set into smaller subsets can improve algorithm efficiency and performance and reduce the risk of overfitting. In fact, overfitting occurs when an algorithm is too tightly fitted to the training data. The dataset can be split randomly or used special techniques such as stratified sampling to ensure that the data is representative of the entire dataset.

Here are some reasons why data partitioning is important:

- **Improved performance:** When working with large datasets, processing all the data at once can be time-consuming and computationally intensive. By partitioning the data into smaller subsets, it becomes possible to process each subset independently and in parallel. This can lead to significant performance improvements, as each subset can be processed more quickly and efficiently.
- **Scalability:** Data partitioning also enables scalability. As the dataset grows in size, it may become impractical or impossible to process it all at once. By partitioning the data, it becomes possible to distribute the processing across multiple machines or nodes in a distributed system, allowing for greater scalability.
- **Reduced memory usage:** Partitioning can also help reduce the amount of memory required to process the data. By processing only one subset at a time, the amount of memory needed to hold the data in memory is reduced, making it possible to work with larger datasets on machines with limited memory.
- **Improved data quality:** In some cases, partitioning can also help improve the quality of the data. By partitioning the data into smaller subsets, it becomes easier to identify and correct errors or inconsistencies in the data.
- **Simplified maintenance:** Finally, data partitioning can simplify maintenance tasks, such as backups and data migration. By dividing the data into smaller subsets, it becomes possible to perform these tasks more efficiently and with less disruption to ongoing processing.

As shown in Figure 3-16a, the data set is usually randomly partitioned into three parts to evaluate the models for predicting future 24-hour maintenance (short-term): training set (64%), test set (20%), and validation set (16%). For the two-step mode (the models to predict the next 7 days, long-term), we split the data into two parts, as shown in Figure 3-16b. Binary dataset partitioning involves dividing a dataset into two parts. This technique is widely used in machine learning to create separate training and testing sets. The training set is used to train the model, and the testing set is used to evaluate its accuracy and performance. This type of partitioning is effective in dividing the data into distinct sets for more efficient machine learning algorithm training and testing.



(a) (b)

Fig 3-16. Amount of data attributed to each of the sets. (a) short-term, (b) long-term.

3.3.2 Normalization

Normalization is a pre-processing stage of any type of problem statement. Indeed, Normalization is an important technique in data science that can help to improve the accuracy, efficiency, performance, interpretability, and compatibility of data. By transforming data into a common scale, normalization can help data scientists to extract valuable insights and knowledge from complex datasets.

There are different techniques and types of normalization that can be used in data analysis, depending on the nature of the data and the objectives of the analysis. Here are some common normalization techniques and types used in data analysis:

- **Min-max normalization:** This technique scales the data to a range between 0 and 1. It involves subtracting the minimum value of the data and dividing by the range (maximum value - minimum value).
- **Z-score normalization:** This technique transforms the data into a standard normal distribution with a mean of 0 and a standard deviation of 1. It involves subtracting the mean of the data and dividing by the standard deviation.

- **Decimal scaling normalization:** This technique involves scaling the data by dividing each value by a power of 10. The power of 10 is chosen such that the largest absolute value in the data is less than 1.
- **Log transformation:** This technique involves taking the logarithm of the data values. This can be useful for data that has a wide range of values or is skewed.
- **Unit vector normalization:** This technique scales each row or column of the data to have a unit length. It involves dividing each value by the Euclidean norm of the row or column.
- **Softmax normalization:** This technique is commonly used in classification problems where the data represents probabilities. It scales the data such that the sum of the probabilities in each row is equal to 1.

These are just a few examples of the normalization techniques and types used in data analysis. The choice of normalization technique depends on the nature of the data and the objectives of the analysis. The goal of normalization is to transform the data into a common scale that facilitates better comparison and analysis.

Min-max scaling normalization is a data normalization technique used in data preprocessing to scale numerical features to a fixed range [69, 70]. The goal of this technique is to transform the data so that it has a similar scale and distribution, which can improve the performance of some machine learning algorithms [71].

In min-max scaling normalization, each feature is scaled to a fixed range, usually between 0 and 1. The formula for min-max scaling normalization is shown in Equation 3-1 :

$$x\_scaled = (x - x\_min) / (x\_max - x\_min) \qquad \text{(Equation 3-1)}$$

*where **x** is the original feature value, **x_min** is the minimum value of the feature, and **x_max** is the maximum value of the feature.*

Considering this, min-max normalization is a simple and straightforward normalization technique that can be beneficial for data with a well-defined range of values, as well as for some machine learning and data visualization algorithms. Min-max normalization can also be used for data sets with a known range of values, non-normal distributions, and non-binary and non-negative data that are not sensitive to outliers. In this study, min-max normalization was used to preprocess the data.

3.3.3 Model Creation

This section presents how the ML and DL model should be trained and tested. model parameter tuning is an essential step in the development of machine learning and deep learning models. It involves defining a performance metric, choosing a search algorithm, defining a search space, conducting the search, and validating the results. With careful parameter tuning, the performance of a model can be significantly improved, leading to more accurate and effective predictions. In greedy search optimization, all the possible combinations of hyperparameters for the ML and DL models are tested, while in randomized search a randomly generated set of combinations is considered and evaluated. This step is crucial to obtain an optimized ML and DL models to handle the PdM problem.

As soon as the most appropriate hyperparameters are identified, the model is fine-tuned using all available data in real-world applications. In contrast, in research studies, data are often split to evaluate the performance of the model. In the training phase of the model, the training set is used, while the test set is used to evaluate the effectiveness of the model.

In the next step, the test set created in the previous step is used. There is no one-size-fits-all approach for evaluating machine learning (ML) and deep learning (DL) models. The choice of evaluation metrics depends on the problem being solved and the nature of the data. However, there are four common metrics used to evaluate ML and DL models:

- **Accuracy**: Accuracy is the most basic evaluation metric, and it measures the proportion of correctly classified samples. It is calculated as the number of correctly classified samples divided by the total number of samples (Equation 3-2).

$$\text{Accuracy} = \frac{1}{n_{samples}} \sum_{i=1}^{n_{samples}} (\hat{y}_i = y_i) \qquad \text{(Equation 3-2)}$$

- **Precision:** Precision is the proportion of true positive samples among all positive samples, and it is a useful metric when the cost of false positives is high. It is calculated as the number of true positive samples divided by the sum of true positive and false positive samples (Equation 3-3).

$$\text{Precision} = \frac{t_P}{t_P + f_P} \qquad \text{(Equation 3-3)}$$

- **Recall**: Recall is the proportion of true positive samples among all actual positive samples, and it is a useful metric when the cost of false negatives is high. It is calculated as the number of true positive samples divided by the sum of true positive and false negative samples (Equation 3-4).

$$\text{Recall} = \frac{tp}{t_P + f_n} \qquad \text{(Equation 3-4)}$$

- **F1 score:** The F1 score is the harmonic mean of precision and recall, and it is a useful metric when the classes are imbalanced. It is calculated as 2 * (precision * recall) / (precision + recall) (Equation 3-5).

$$\text{F1} = 2 * \frac{(Precision * Recall)}{(Precision + Recall)} \qquad \text{(Equation 3-5)}$$

where $\hat{y}_i$ is the predicted class, y_i is the expected class, t_p stands for true positives, f_p for false positives, and f_n for false negatives.

3.3.4 Machine Learning and Deep Learning Algorithms

Algorithms are a crucial component of machine learning as they determine how the model will learn from the data and make predictions. In essence, an algorithm is a set of instructions that a computer uses to perform a specific task. In the context of machine learning, algorithms are used to train models on data and make predictions based on that training. There are many different algorithms that can be used for machine learning, and the choice of algorithm will depend on the problem being solved and the characteristics of the data.

The importance of choosing the right algorithm lies in its ability to learn effectively from the data and make accurate predictions. A well-chosen algorithm can lead to better model performance and more accurate predictions, while a poorly chosen algorithm can lead to overfitting, underfitting, or poor performance. It is also important to keep in mind that no single algorithm is the best for all problems, and that it is often necessary to try several different algorithms and evaluate their performance to determine the best one for a specific problem. Additionally, different algorithms may be more or less computationally expensive, and the trade-off between accuracy and computational cost should be considered when selecting an algorithm. In this work, a set of data, coming from different sources, available online [18] was evaluated. First, 10 of the most popular machine learning algorithms used in various works for PdM were applied. Machine learning algorithms are used:

- **Random Forest Classifier (RFC):**

  Random Forest Classifier (RFC) is a popular machine learning algorithm used for classification tasks. It is an ensemble learning method that combines multiple decision trees, where each tree is built using a different subset of the training data and a random subset of the features [72]. The RFC algorithm works by creating a forest of decision trees, where each tree is built using a random subset of the training data and a random subset of the features. Each tree is trained using a different subset of the data, which helps to reduce overfitting and improve the accuracy of the model.
  During prediction, the RFC algorithm combines the predictions of all the trees in the forest and returns the class with the most votes. This approach helps to improve the accuracy of the model and reduce the risk of overfitting.

- **eXtreme Gradient Boosting Classifier (XGB Classifier):**

  eXtreme Gradient Boosting Classifier (XGB Classifier) is a powerful machine learning algorithm used for classification and regression tasks. It is a variant of the gradient boosting algorithm that uses a tree-based approach to build a predictive model.

  The XGB Classifier works by iteratively building an ensemble of decision trees, where each tree is built to correct the errors made by the previous tree. It uses a gradient descent optimization algorithm to train each tree, which helps to improve the accuracy of the model and reduce overfitting.

  The XGB Classifier is a popular algorithm because of its ability to handle large datasets with a high number of features. It is also known for its speed and scalability, making it a popular choice for solving problems related to image and speech recognition, text classification, and anomaly detection. One of the key advantages of the XGB Classifier is its ability to handle missing data and outliers. It also supports a wide range of loss functions and can be customized to handle specific problems [73].

- **Logistic Regression (LR):**

  Logistic Regression (LR) is a popular machine learning algorithm used for classification tasks. It is a statistical model that is used to estimate the probability of a binary outcome based on one or more input variables.

  In LR, the output variable is binary (i.e., it can take one of two possible values), and the algorithm estimates the probability of the output variable taking a particular value based on the input variables. The LR algorithm models this relationship using a logistic function, which is a special type of S-shaped curve**.** One of the advantages of LR is that it is a relatively simple algorithm that is easy to implement and interpret. It also performs well on small to medium-sized datasets with a moderate number of input variables. However, it may not be suitable for complex problems or datasets with a large number of features [74].

- **Extra Trees Classifier:**

  Extra Trees Classifier is a machine learning algorithm that is similar to the Random Forest Classifier (RFC). Like RFC, it is an ensemble learning method that combines multiple decision trees to make predictions. The Extra Trees Classifier algorithm works by creating a forest of decision trees, where each tree is built using a random subset of the training data and a random subset of the features. However, unlike RFC, the Extra Trees Classifier algorithm selects the splitting points for each node of the decision tree randomly, without considering the optimal split. This approach helps to reduce the variance of the model and can improve the performance of the algorithm, especially when dealing with noisy data.

  During prediction, the Extra Trees Classifier algorithm combines the predictions of all the trees in the forest and returns the class with the most votes, similar to RFC. One of the key advantages of the Extra Trees Classifier algorithm is its ability to reduce overfitting by randomly selecting the splitting points of the decision trees. It is also a relatively fast algorithm, making it a good choice for large datasets with many features [75].

- **Bagging Classifiers:**

  Bagging Classifier is a machine learning algorithm that is used for ensemble learning, similar to the Random Forest Classifier and Extra Trees Classifier. Bagging stands for Bootstrap Aggregating, which refers to the technique of sampling the training data with replacement to create multiple subsets of the data, and then training a separate classifier on each subset. One of the key advantages of Bagging Classifier is its ability to reduce overfitting, which is a common problem in machine learning. By training multiple classifiers on different subsets of the data, Bagging Classifier can produce a more stable and robust model that generalizes well to new, unseen data.

  One limitation of Bagging Classifier is that it can be computationally expensive, especially when dealing with large datasets or complex models. It also requires careful selection of hyperparameters to ensure optimal performance [76].

- **Support Vector Classifier (SVC):**

  Support Vector Classifier (SVC) is a type of supervised machine learning algorithm that can be used for classification tasks. It is a type of Support Vector Machine (SVM) algorithm that finds a hyperplane in a high-dimensional space that maximally separates the data into different classes.

  The main goal of SVC is to find the best possible decision boundary that separates the data into different classes. It does this by first transforming the data into a higher dimensional space, where it is easier to find a hyperplane that separates the data. The decision boundary is then chosen as the hyperplane that maximizes the margin, which is the distance between the hyperplane and the closest data points from each class. SVC can handle both linearly separable and non-linearly separable data by using different types of kernel functions, such as linear, polynomial, radial basis function (RBF), and sigmoid. The kernel function is used to transform the data into a higher dimensional space, where it may be easier to find a hyperplane that separates the data [77].

- **Linear Support Vector Classifier (Linear SVC):**

  Linear Support Vector Classifier (Linear SVC) is a variant of the Support Vector Machine (SVM) algorithm used for solving linearly separable classification problems. Linear SVC aims to find the optimal hyperplane that separates the training data points into different classes in a linearly separable manner. The hyperplane is chosen such that it maximizes the margin, which is the distance between the hyperplane and the closest data points from each class.

  Linear SVC works by finding the weights and biases that define the hyperplane equation. The training process of Linear SVC involves minimizing the sum of squared weights subject to the constraint that all data points are correctly classified. This optimization problem is solved using the Lagrange multiplier method, resulting in a set of coefficients that can be used to define the hyperplane.

  Unlike the regular SVM algorithm, Linear SVC is more efficient and faster for large-scale datasets, as it doesn't require transforming the data into a higher-dimensional space. It can handle a large number of features and is robust to noise and outliers in the data [78].

- **Stacking Classifier:**

A stacking classifier is a type of ensemble learning algorithm that combines multiple individual classifiers or models to make a final prediction. The idea behind stacking is to use the outputs of individual classifiers as input features for a final classifier, which then makes the final prediction.

Stacking can be used for a variety of tasks, including classification and regression. It is particularly useful when the base classifiers have different strengths and weaknesses, as combining them can lead to better overall performance.

One potential issue with stacking is overfitting, where the model performs well on the training data but poorly on new, unseen data. To address this, it is important to use cross-validation during training and to ensure that the base classifiers are diverse enough to capture a wide range of features in the data [79].

- **Adaptive Boosting Classifier (AdaBoost):**

Adaptive Boosting Classifier, also known as AdaBoost, is a popular ensemble learning algorithm that combines weak classifiers to form a strong classifier. AdaBoost works by iteratively training a sequence of weak classifiers on the same dataset, each time giving more weight to the misclassified samples from the previous iteration. In this way, AdaBoost focuses on the hard-to-classify examples and trains the weak classifiers to perform better on those examples.

AdaBoost is a versatile algorithm that can be used with any classification algorithm as long as it can handle weighted examples. Some popular base classifiers used with AdaBoost include decision trees, SVMs, and neural networks. One of the key benefits of AdaBoost is that it can achieve high accuracy with a relatively small number of weak classifiers, making it computationally efficient.

However, AdaBoost is sensitive to noisy data and outliers, which can reduce its performance. To address this, techniques such as outlier detection and data preprocessing can be used. Additionally, AdaBoost can be prone to overfitting, so it is important to use cross-validation to select the optimal number of weak classifiers and to avoid overfitting [80].

- **Decision Tree Classifiers:**

  A Decision Tree Classifier is a supervised learning algorithm used for classification and regression tasks. It is a tree-structured model that partitions the input space into disjoint regions and assigns a class label or regression value to each region. The decision tree algorithm works by recursively splitting the input space into smaller regions based on the values of input features. The splitting is performed based on the features that provide the most information gain, which is a measure of the reduction in entropy or impurity of the data.
  Decision Trees are powerful algorithms that can handle both categorical and continuous input features, and can handle missing values. They are also interpretable, meaning that the learned model can be easily visualized and understood by humans. Additionally, decision trees can handle non-linear relationships between input features and the output variable.
  However, decision trees can be prone to overfitting, where the model performs well on the training data but poorly on new, unseen data. To address this, techniques such as pruning, setting a minimum number of samples per leaf node, or using ensemble methods such as Random Forests can be used [81].

Second, innovatives Deep Learning method was applied to the dataset. In this work, two types of Deep Learning algorithms are used:

- **Convolutional Neural Networks (CNNs):**

  Convolutional Neural Networks (CNNs) are a type of deep learning algorithm that is particularly well-suited for image recognition tasks. CNNs use a hierarchical architecture of layers that learn increasingly complex features from the input image data. The key innovation of CNNs is the use of convolutional layers, which apply a set of filters to the input image and produce feature maps that capture local patterns and structures. The basic steps of a CNN algorithm are as follows:

1. **Convolutional Layers:** Applying a set of convolutional filters to the input image to produce feature maps that capture local patterns and structures.

2. **Pooling Layers:** Reduceing the dimensionality of the feature maps by downsampling them using techniques such as max pooling or average pooling.

3. **Activation Layers:** Applying a non-linear activation function to the pooled feature maps to introduce non-linearity into the model.

4. **Fully Connected Layers:** Combining the output of the activation layers into a vector and pass it through a series of fully connected layers to make the final classification decision.

CNNs are particularly well-suited for image recognition tasks and time series because they can capture local patterns and structures in an image, regardless of their location. This allows the model to be invariant to small shifts and distortions in the input image.
However, training a CNN can be computationally intensive and requires a large amount of labeled training data. Additionally, CNNs are often considered to be black box models, making it difficult to understand how the model is making its decisions [82].

- **Long Short-Term Memory networks (LSTM):**

Long Short-Term Memory networks (LSTMs) are a type of recurrent neural network (RNN) that is particularly well-suited for modeling sequential data, such as time series or natural language text. LSTMs use a memory cell and a set of gates to control the flow of information through the network over time, allowing it to selectively remember or forget information as needed.
The basic components of an LSTM network are:

1. **Memory Cell**: Stores information over time.

2. **Input Gate:** Controls which information is allowed into the memory cell.

3. **Forget Gate:** Controls which information is discarded from the memory cell.

4. **Output Gate:** Controls which information is output from the memory cell.

   The LSTM network is trained on a sequence of input data and learns to predict the output at each time step. During training, the weights of the network are adjusted to minimize the difference between the predicted output and the actual output.

   LSTMs are particularly well-suited for tasks such as speech recognition, language translation, and sentiment analysis because they can capture long-term dependencies in the input sequence. Unlike traditional RNNs, which can suffer from the "vanishing gradient" problem, LSTMs use gates to selectively control the flow of information through the network, making them more effective at capturing long-term dependencies.

   However, training an LSTM can be computationally expensive and requires a large amount of training data. Additionally, LSTMs are often considered to be black box models, making it difficult to understand how the model is making its predictions.

Overall, the key to choosing a good AI algorithm for predictive maintenance is to have a clear understanding of the problem and the data, and to evaluate different algorithms based on their performance, scalability, and interpretability.

3.4 Feature Engineering

Feature engineering is one of the most important steps in machine learning and deep learning. Feature engineering directly affects code execution time and results. Feature engineering is the process of selecting, transforming, and creating features from the raw data that are relevant and informative for a machine learning or deep learning model. Feature engineering is a critical step in the machine learning and deep learning pipeline because the quality of the features can have a significant impact on the performance of the model.

Initially, statistical and machine learning approaches are utilized to scrutinize the features and correlations of all the variables. Afterward, the procedure of feature selection is executed. In this stage, depending on the dataset, unsuitable and irrelevant features are disregarded. Essentially, feature engineering is the utilization of statistical methods and techniques to analyze the data and the relationships between variables and transform them into features that are appropriate for machine learning and deep learning algorithms [83].

In machine learning, feature engineering typically involves selecting a subset of the input features that are most relevant for the prediction task and transforming them into a more useful representation. For example, if the input data includes a date field, feature engineering might involve extracting the day of the week or the time of day as separate features [84].

In deep learning, feature engineering can involve using pre-trained models such as CNNs or LSTMs to extract high-level features from the raw input data. These pre-trained models are often trained on large amounts of data and can capture complex patterns and structures in the input data that are difficult to extract manually [85].

In this data set, especially telemetry data, the data is recorded in real time by the sensors and reported as an hourly average. One of the appropriate techniques to reduce the effects of undesirable factors and minimize noise is to use a time window. Given the dataset and the project requirements, we need to determine how large the lookback should be for the model, which is called lag features. In this work, two different time windows are used: 3 hours (short-term) and 24 hours (long-term). An example of lag features for real-time telemetry data can be found in Table 3-6.

Table 3- 6 An example of Lag Features (short-term) for telemetry data..

	Machine ID	Datetime	Volt min_3h	Rotate min_3h	Pressure min_3h	Vibration min_3h	Volt max_3h	Rotate max_3h	Pressure max_3h	Vibration max_3h
0	1	1/2/2015 6:00	158.2714	403.2359	92.4391	32.5168	200.8724	495.7779	96.5354	52.3558
1	1	1/2/2015 9:00	160.5288	384.6459	86.9442	29.5276	197.3631	486.4590	114.3420	42.9925
2	1	1/2/2015 12:00	147.3006	412.9656	90.7113	34.2030	173.3945	439.5794	110.4089	37.1171
3	1	1/2/2015 15:00	152.4207	385.3549	99.5068	30.6651	185.2053	497.8406	105.9932	47.8624
4	1	1/2/2015 18:00	145.2484	424.5426	93.7438	37.4222	180.0307	495.3764	111.9505	43.0997

Finally, the records related to machines, such as the error log and the maintenance log, as well as the model and years of operation of each machine, can be used as features. It should be noted that while the error log is associated with date/time, unlike telemetry data, it is categorical and not numerical. Therefore, averaging over time intervals should not be carried out. Instead, we count the number of errors in each category in a lagging window.

Also, the different parts of this dataset allow us to extract new features; For example, the maintenance log contains information about replacing different parts of the machine, but we know from experience and industry knowledge that the longer a component is used, the more likely it is to fail. This type of feature engineering, which leads to creating specific and relevant features using engineering knowledge, plays a crucial role in predictive maintenance models.

After developing the features, it is important to select the most relevant features for the analysis or model building. However, before the feature selection step, it is essential to integrate and consolidate the data from different subsets and data collection conditions.

Data integration involves combining data from different sources, such as databases, files, or web services, into a single, unified dataset. This process may involve resolving differences in data structures, formats, or naming conventions across different sources. Data consolidation involves cleaning and transforming the data to make it consistent and usable for analysis or modelling. This may involve removing duplicates, filling in missing values, or transforming data into a common format.

After the data integration and consolidation steps, the next step is to select the most relevant features for analysis or model building. This can help to reduce the complexity of the dataset and improve the accuracy and interpretability of the analysis or model. As mentioned earlier, there are various methods for feature selection, and the choice of method depends on the specific problem and dataset. It is important to evaluate the performance of the selected features on a validation set to ensure that the selected features are robust and generalize well to new data [86].

Figure 3-17 shows the linear correlation between variables. Feature correlation is a statistical measure that describes the degree of association or relationship between two or more variables in a dataset. In machine learning and data analysis, feature correlation is often used to identify which variables are strongly related to each other, and which variables are less related or unrelated. There are several methods to measure feature correlation, including:

- **Pearson correlation coefficient:** This method measures the linear relationship between two continuous variables. It ranges from -1 to 1, where -1 indicates a perfect negative correlation, 0 indicates no correlation, and 1 indicates a perfect positive correlation.
- **Spearman's rank correlation coefficient:** This method measures the monotonic relationship between two variables, which means that it measures whether the variables tend to increase or decrease together, but not necessarily at a constant rate.
- **Kendall's rank correlation coefficient:** This method also measures the monotonic relationship between two variables, but it is less sensitive to outliers than Spearman's method

From this plot, it is clear that the correlation between variables and features is very low and the correlation coefficient of a large number of features is close to zero (less than 0.01).

In order to select the appropriate features, various minimum correlation coefficients have been studied. Research has shown that prediction models using machine learning and deep learning perform better when features with a correlation coefficient greater than 0.01 are used. In this article, 34 features were selected. To improve the performance of artificial intelligence algorithms, we used statistical concepts in this article to create additional features for the telemetry features over a 3-hour and a 24-hour lag window for the dataset:

- Minimum values;
- Maximum values;
- Mean values;
- Standard deviation;

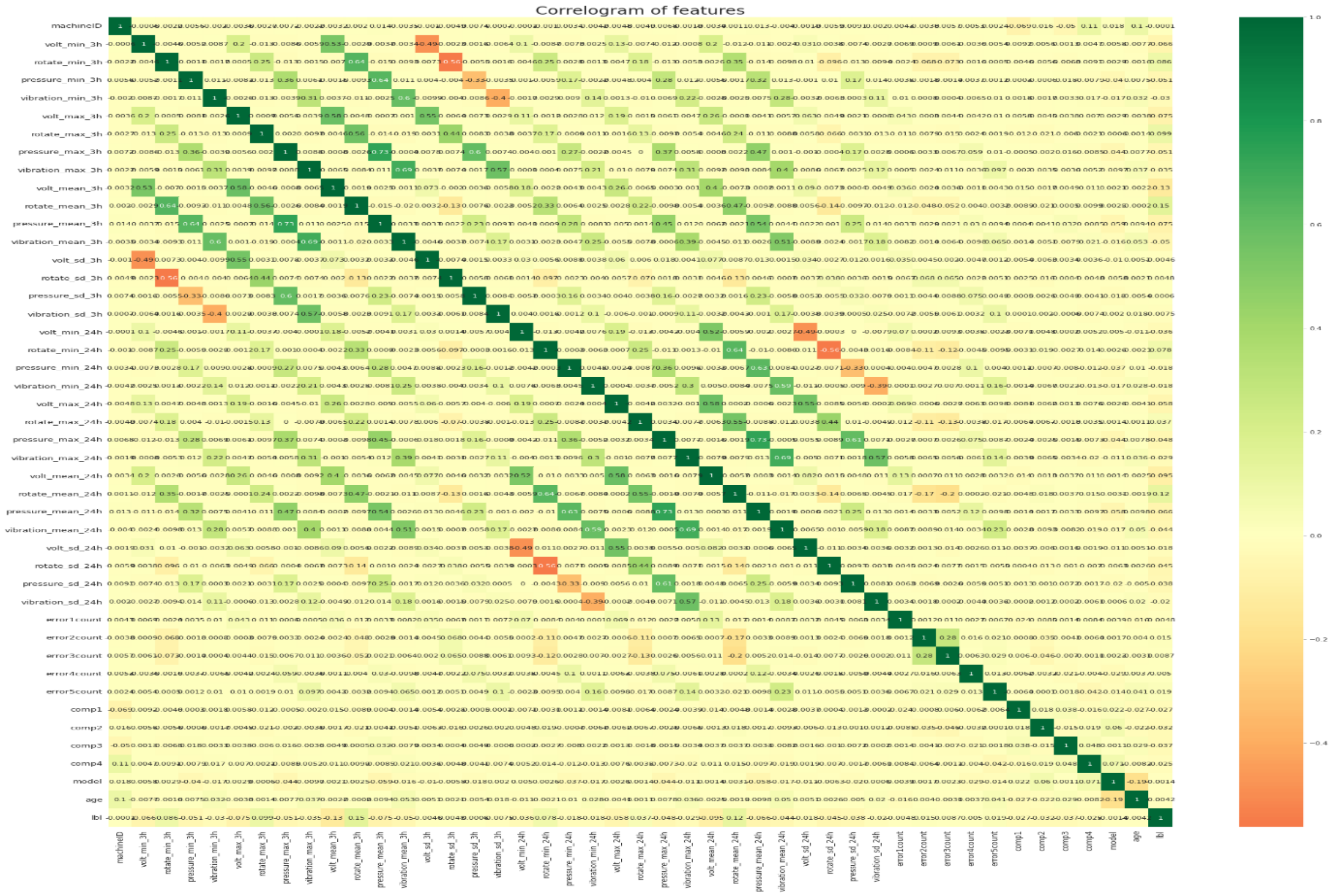


Fig 3-17. Features correlation.

3.5 Deep learning model construction

As mentioned earlier, the dataset is a time series where the data was recorded in real time. Two powerful deep learning algorithms for time series data that are commonly used for modeling PdM are CNN and LSTM. A brief introduction to the structure of the deep learning model used in this article follows.

Convolutional Neural Networks (CNNs) is a multilayer deep learning algorithm commonly used for PdM models and image classification. The CNN model consists of three layers: Convolutional Layer, Pooling Layer and Flattening Layer [87]. Here is a CNN model structure for predictive maintenance:

1. **Input layer:** This layer takes in the time series sensor data from the equipment as input.
2. **Convolutional layers:** These layers apply a set of learnable filters to the input data, producing a set of feature maps that capture different aspects of the data. The filters are designed to detect patterns in the sensor data that are indicative of impending equipment failures. Each filter performs a convolution operation on the input data, which involves sliding the filter over the time series data and computing the dot product between the filter weights and the corresponding data points. The output of a convolutional layer is a set of feature maps that are then passed through activation functions such as ReLU to introduce non-linearity.
3. **Pooling layers:** These layers reduce the temporal resolution of the feature maps by performing a downsampling operation, such as max pooling or average pooling. The purpose of pooling is to reduce the amount of data that the network needs to process while preserving the important patterns in the data.
4. **Fully connected layers:** These layers perform a matrix multiplication between the flattened feature maps and a set of learnable weights to produce the final output. These layers are used to make a prediction about the likelihood of an equipment failure based on the patterns detected in the input data.
5. **Dropout layers**: These layers randomly drop out a fraction of the neurons in the previous layer during training, which helps to prevent overfitting.
6. **Batch normalization layers:** These layers normalize the activations of the previous layer to improve the stability and speed of training.
7. **Output layer:** This layer produces a binary prediction (failure or non-failure) based on the output of the fully connected layers.

The overall architecture of a CNN for predictive maintenance typically consists of multiple convolutional and pooling layers, followed by one or more fully connected layers and an output layer for prediction. The exact number and size of the layers depend on the specific equipment and sensor data being used for prediction. Training a CNN for predictive maintenance requires a large amount of historical data from the equipment to learn the patterns that are indicative of equipment failures. Once trained, the CNN can be used to make real-time predictions about the likelihood of equipment failures based on the latest sensor data. Figure 3-18 shows the structure and details of the CNN network used in Python and in this work.

```
Layer (type)                 Output Shape              Param #
=================================================================
conv1 (Conv1D)               (None, 46, 128)           384

conv2 (Conv1D)               (None, 46, 256)           65792

maxpool1 (MaxPooling1D)      (None, 23, 256)           0

conv3 (Conv1D)               (None, 23, 512)           262656

AvgPool2D (MaxPooling1D)     (None, 11, 512)           0

flatten_1 (Flatten)          (None, 5632)              0

batch_normalization_2 (Batc   (None, 5632)             22528
hNormalization)

drop2 (Dropout)              (None, 5632)              0

dense_2 (Dense)              (None, 256)               1442048

out_layer (Dense)            (None, 4)                 1028

=================================================================
Total params: 1,794,436
Trainable params: 1,783,172
Non-trainable params: 11,264
```

Fig 3- 18. Details of CNN model structure and layers.

The Long Short-Term Memory (LSTM) algorithm is one of the most commonly used algorithms for PdM models. LSTM is a type of recurrent neural network (RNN) that is commonly used for predictive maintenance tasks involving time series data. Feedback connections are the most important aspect of LSTM models, and unlike recurrent neural networks, RNNs can maintain time dependencies between input data over time. Because of these characteristics, LSTM algorithms are incredibly effective at processing time series data [88]. Here is a LSTM model structure for predictive maintenance:

1. **Input layer:** This layer takes in the time series sensor data from the equipment as input.
2. **LSTM layers:** These layers contain memory cells that allow the network to retain information over time and make predictions based on the history of the data. Each LSTM cell contains a set of learnable weights that determine how the input data and the previous cell state are combined. The cell state can be updated or reset based on the input data

and the output of the previous cell. The output of an LSTM layer is a set of hidden states that capture the relevant features of the time series data.

3. **Fully connected layers:** These layers perform a matrix multiplication between the hidden states and a set of learnable weights to produce the final output. These layers are used to make a prediction about the likelihood of an equipment failure based on the patterns detected in the input data.
4. **Dropout layers:** These layers randomly drop out a fraction of the neurons in the previous layer during training, which helps to prevent overfitting.
5. **Batch normalization layers:** These layers normalize the activations of the previous layer to improve the stability and speed of training.
6. **Output layer:** This layer produces a binary prediction (failure or non-failure) based on the output of the fully connected layers.

Figure 3-19 shows the structure and details of LSTM used in Python and this article.

Layer (type)	Output Shape	Param #
lstm_3 (LSTM)	(None, 46, 32)	4352
lstm_4 (LSTM)	(None, 46, 64)	24832
lstm_5 (LSTM)	(None, 46, 64)	33024
flatten_2 (Flatten)	(None, 2944)	0
dense_3 (Dense)	(None, 512)	1507840
batch_normalization_3 (BatchNormalization)	(None, 512)	2048
drop2 (Dropout)	(None, 512)	0
out_layer (Dense)	(None, 4)	2052

Total params: 1,574,148
Trainable params: 1,573,124
Non-trainable params: 1,024

Fig 3- 19. Details of CNN and LSTM model structure and layers.

3.6 Results

In recent years, PdM models, ML algorithms and techniques have greatly improved. However, it is still a big challenge for companies and organizations. This is because the implementation of PdM models requires extensive and meticulous planning between hardware and software and requires detailed training of personnel to accurately record all details related to maintenance, such as faults, periodic and non-periodic replacement of equipment and components, component downtime, and errors. Therefore, even when outsourcing to expertise companies, many requirements should be met, such as:

- Identification of critical and sensitive components for the performance of the system and their monitoring
- Determination of parameters affecting the performance and their measurement with sensors
- Selection of the best technique and ML and DP models.
- Determining the location(s) of sensor(s)

In addition, it is important to consider that in each company different influencing factors can affect the preference of models and hyperparameters. For example, the price of each component and the time required to replace and access each component can significantly influence feature selection and algorithm performance.

In this paper, we compare the performance of different ML and deep learning models in two time periods of 24 hours and seven days ahead. The results are divided into short-term (24 hours) and long-term (seven days) predictive maintenance models.

3.6.1 Short-term predictive maintenance models

In this study, ten different machine learning algorithms by examining the dataset, the features, and the selection of appropriate hyperparameters are investigated. Table 3-7 and Figure 3-20 show the results of the models based on accuracy, recall, precision, and F1 score with minimum-maximum scaling normalization. The results show that the machine learning models based on the identified features for time windows of three hours (3h) and twenty-four hours (24h) predict well the failure probability for each component.

Table 3- 7 Performance of ML models with min-max scaling normalization.

ML Model	Accuracy	Precision	Recall	F1
Random Forest Classifier	0.9987	0.966	**0.9775**	**0.9714**
Stacking Classifier	0.9987	0.9635	**0.9786**	**0.9709**
Extra Trees Classifier	0.9986	0.9658	**0.9776**	**0.9715**
SVC	0.9984	0.9519	0.9783	0.9649
Bagging Classifier	0.9983	0.9507	0.9762	0.9632
XGB Classifier	0.9984	0.9615	0.9775	0.9714
Linear SVC	0.9959	0.8807	0.9830	0.9259
Decision Tree Classifier	0.9985	0.9661	0.9578	0.9619
Logistic Regression	0.9896	0.7466	0.9730	0.8348
AdaBoost Classifier	0.9589	0.6423	0.9003	0.6886

In general, the value of the recall parameter is essential for predictive maintenance. The value of the recall parameter (and consequently F1) indicates the number of real failures that AI models can predict [89]. This parameter is more critical when we are dealing with an unbalanced dataset and the number of actual failures that the algorithm cannot predict (false negative) is higher than the number of failures that the algorithm incorrectly predicts (false positive) [90]. Since the occurrence of failures during the life cycle of a machine is very small compared to normal operation without failures (Table 3-5), we are constantly faced with an unbalanced data set in maintenance prediction. This imbalance of data, which appears as an imbalance between classes, causes a disturbance (illusion) of the algorithms. In fact, algorithms tend to create an equilibrium between the classes. Since the value of the failure class is much lower, they split some of the most frequent examples (normal operation of the machine) into rare examples (occurrence of failure). As a result, the value of accuracy can be high compared to recall.

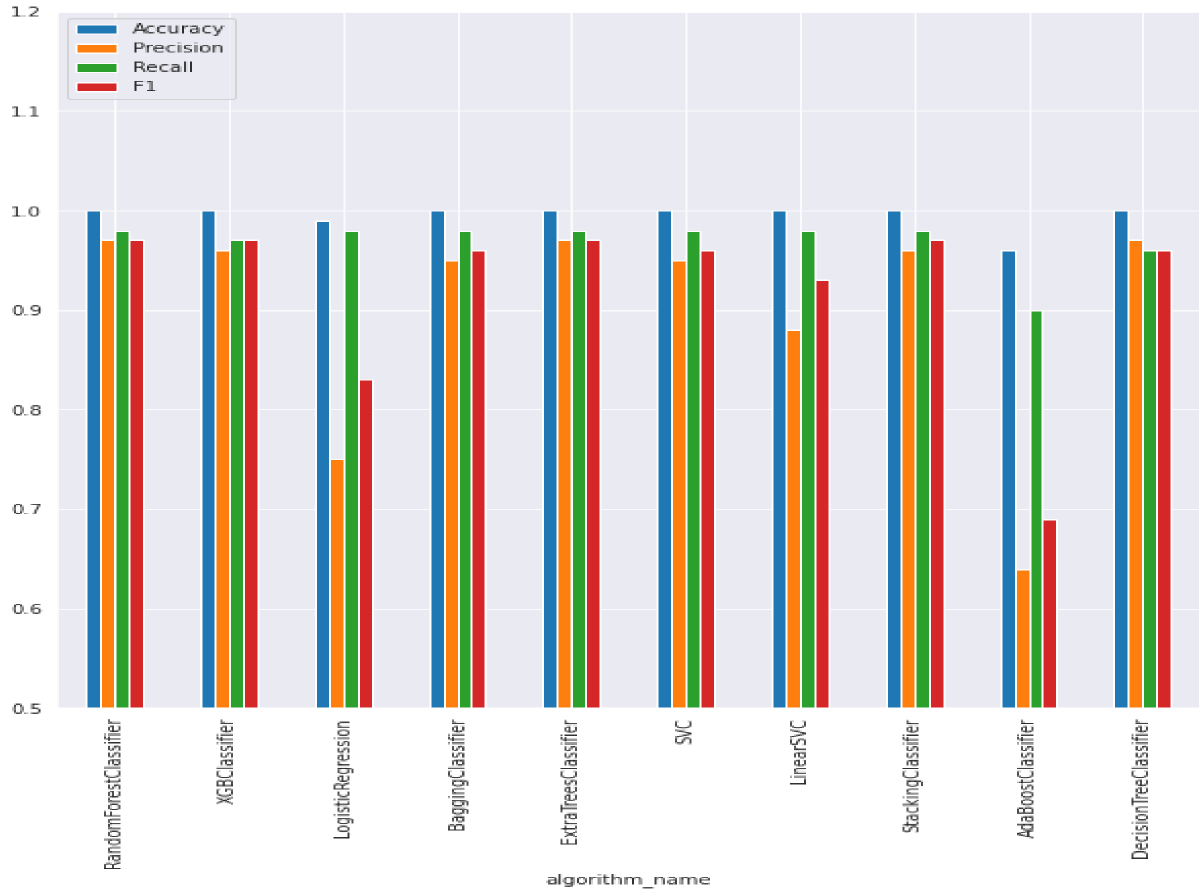


Fig 3- 20. Diagram of ML models performance with min-max scaling normalization for short-time prediction.

Comparing the results of the short-term machine learning predictive maintenance models in Table 3-7 and Figure 3-20, it is clear that the performance of the algorithms (based on recall and F1) was excellent except for two algorithms, the AdaBoost classifier and logistic regression. The three algorithms Random Forest Classifier, Stacking Classifier, and Extra Trees Classifier have similar and better performances than the other machine learning algorithms. Since machine learning models are excellent for predicting component failures, deep learning methods were not used for maintenance prediction.

Table 3-8 shows the recall values for each component. In this paper, we considered the importance of all four components to be equal because information such as the price of each component, the time required for replacement, the importance of each component in the process, and access to spare parts are not available, but in industry, these factors can be effective in analyzing the results and making the final decision on the maintenance plan.

Table 3- 8 Performance of the ML models (short-term) with min-max scaling normalization for components based the on Recall parameter.

ML Model	Comp1	Comp2	Comp3	Comp4	None
Random Forest Classifier	**0.9840**	**0.9862**	**0.9483**	**0.9697**	**0.9993**
Extra Trees Classifier	**0.9491**	**0.9977**	**0.9909**	**0.9610**	**0.9992**
Stacking Classifier	**0.9423**	**0.9954**	**0.9818**	**0.9740**	**0.9993**
Decision Tree Classifier	0.9487	0.9724	0.9331	0.9351	0.9996
XGB Classifier	0.9423	1.00	0.9301	0.9784	0.9991
Logistic Regression	0.9519	0.9677	0.9909	0.9913	0.9899
Bagging Classifier	0.9153	0.9977	0.9970	0.974	0.9989
SVC	0.9263	0.9954	0.9970	0.9740	0.9989
Linear SVC	0.9487	0.9977	0.9939	0.9784	0.9963
AdaBoost Classifier	0.8045	0.9424	0.9240	0.8701	0.9605

3.6.2 Long-term predictive maintenance models

As mentioned earlier, imbalanced datasets are a common challenge in predictive maintenance applications, where the number of failure cases is typically much smaller than the number of normal operating cases. This can lead to models that are biased towards the majority class, resulting in poor performance on the minority class. Based on Microsoft Azure Predictive Maintenance data set [18], a new time window needs to be created if predictions for more than 24 hours are desired It is shown in Tables 3-1 and 3-5 that the data related to failure accounts for less than 2% and is presented as an hourly average. If the time window is extended beyond 24 hours, a portion of the failure-related data will be missed, leading to a high imbalance between classes, which will disturb the algorithm and result in erroneous machine learning outcomes. Since timing is critical in Industry 4.0, this paper presents two methods, the weighted average factor (weighted loss functions), and the two-step method, for long-term predictive models for maintenance.

The concept of weighting is one of the typical strategies for dealing with unbalanced classes in a data set. In this dataset, class 4 (class without error=none) accounts for 98% of the data, while classes 0, 1, 2, and 3 (in conjunction with comp1, comp2, comp3, and comp4) account for the remaining 2%. The idea behind this technique is to give more importance to the minority class instances, so that the model can learn to better distinguish between the minority and majority classes. The minority group receives a higher weighting coefficient, while the majority group receives a lower one. There are various algorithms that support the use of the weighted average technique, such as decision trees, support vector machines, and neural networks. Additionally, some libraries, such as Scikit-learn in Python language programming, provide built-in support for weighted average classification. The weighting of each class is shown in Table 3-9.

Table 3- 9 The average weight coefficients for each class.

Components	Class Label	Class Weight
Comp1	0	10.483
Comp2	1	9.661
Comp3	2	10.405
Comp4	3	10.375
None	4	0.217

Generally, Machine learning algorithms often assume that the classes are balanced, meaning that there are roughly equal numbers of instances in each class. If the dataset is unbalanced, with a significant difference in the number of instances between classes, the model may be biased towards the majority class and may not perform well on the minority class.

The weighted average technique, also known as class weighting, is one approach to dealing with imbalanced datasets. Table 3-10 and Figure 3-21 show the results of using the weighted average technique in machine learning and deep learning, respectively. It's important to note that the effectiveness of this technique may vary depending on the specific dataset and the algorithm used.

Table 3- 10 Performance of ML models for long-term prediction by using the weighted average technique.

ML Model	Accuracy	Precision	Recall	F1
Bagging Classifier	0.93	0.92	**0.93**	**0.93**
Stacking Classifier	0.93	0.91	**0.93**	**0.92**
Decision Tree Classifier	0.92	0.92	0.92	0.92
XGB Classifier	0.93	0.90	0.93	0.91
AdaBoost Classifier	0.93	0.90	0.93	0.91
Linear SVC	0.90	0.92	0.90	0.91
SVC	0.92	0.85	0.92	0.88
Random Forest Classifier	0.92	0.87	0.92	0.89
Extra Trees Classifier	0.92	0.86	0.92	0.89
Logistic Regression	0.88	0.93	0.88	0.90

The weighted average technique is a simple method that provides a good estimate of the probability of failure for the next seven days, but it is still not reliable for predicting component failures. Research has shown that this technique could only predict the failure of component 3 in the XGB Classifier and AdaBoost Classifier algorithms, while it has acceptable accuracy in detecting machine failures in the next seven days, as shown in Figure 3-22. The weighted average coefficient is a statistical technique that affects the final results of machine learning algorithms and hyperparameters, While the weighted average coefficient can help to balance class distribution, it may not be effective in training the algorithms. The technique only adjusts the weight of the samples during training, but it does not change the underlying model structure or the way the model learns from the data. This means that if the underlying model is not able to capture the patterns and characteristics of the minority class, the weighted average coefficient may not be sufficient to improve the model performance, so the results may be accompanied by errors.

In addition, using the weighted average coefficient may introduce errors in the final results. For example, if the minority class is heavily underrepresented in the training set, the model may learn to overfit to the minority class and perform poorly on new data. This is known as the overfitting problem, and it can lead to poor generalization and unreliable results. Therefore, an innovative two-step method for predicting the maintenance of machines and components is investigated.

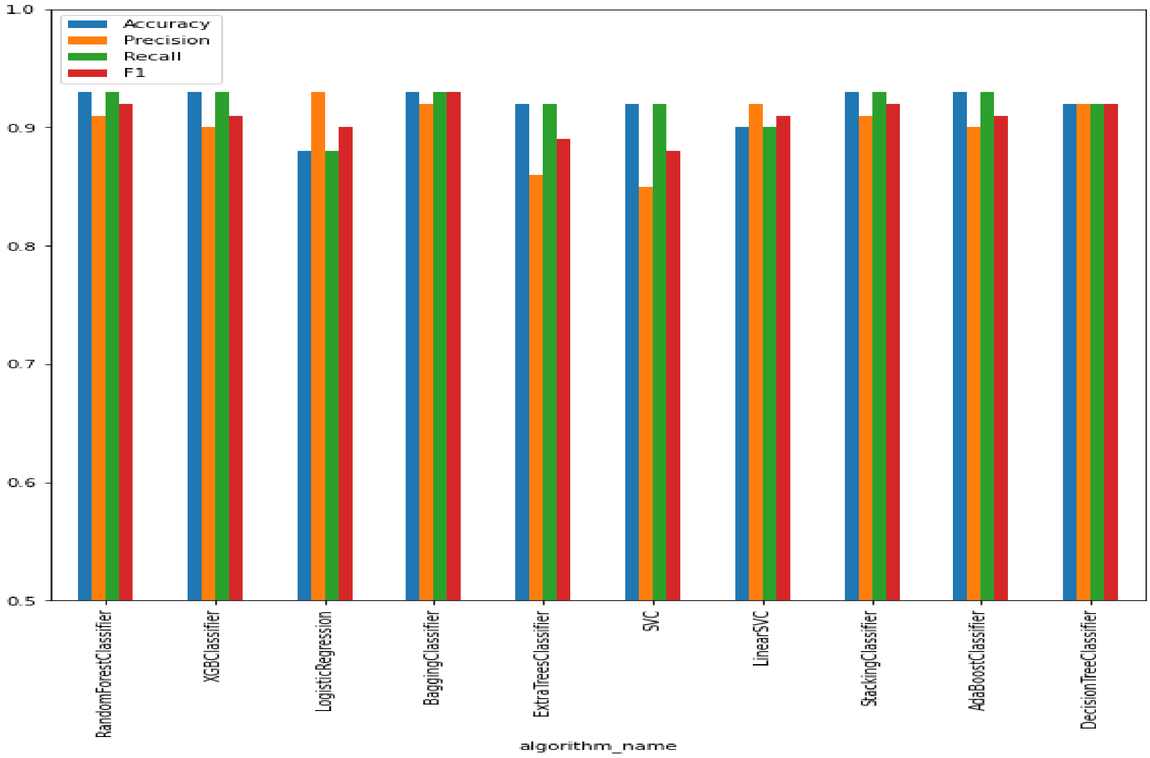


Fig 3- 21. Performance of the ML models after applying different class weights.

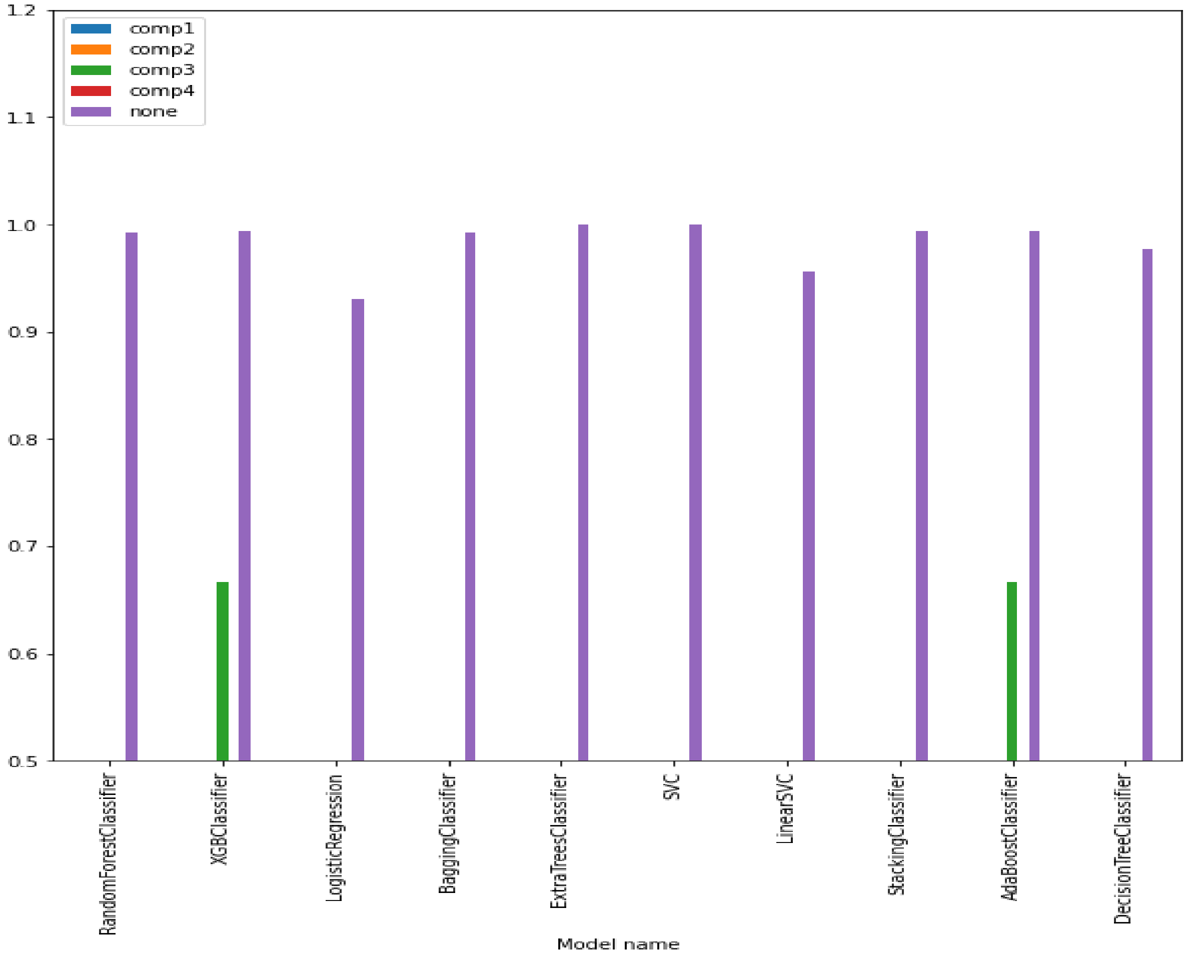


Fig 3- 22. The performance of machine learning algorithms based on the recall parameter by using the weighted average technique.

As mentioned earlier, we face many challenges in predicting machine and component failures due to the severe imbalance of the dataset. In this work, we used a two-step method. In the first step, we divided the dataset into two parts: the first class, the normal machine data without fault reports, while the second class, the dataset with reported faults for all machines and components. In fact, in the initial step, classes 0 to 3 linked with components 1 to 4 were merged into one class. Only AI algorithms were employed in this step to study the data between the two classes, and the results only displayed the likelihood of failure for each machine in the subsequent seven days.

The results of the machine learning and deep learning algorithms are shown in Table 3-11.

Table 3- 11 Performance of ML and DL models for long-term prediction in first step.

ML Model	Accuracy	Precision	Recall	F1
CNN	0.95	0.95	**0.95**	**0.95**
LSTM	0.93	0.96	**0.93**	**0.94**
Logistic Regression	0.95	0.80	**0.95**	**0.86**
Random Forest Classifier	0.97	0.89	0.89	0.89
Stacking Classifier	0.97	0.92	0.87	0.89
Bagging Classifier	0.97	0.88	0.88	0.88
Linear SVC	0.95	0.80	0.93	0.85
Extra Trees Classifier	0.97	0.93	0.86	0.89
Decision Tree Classifier	0.96	0.86	0.87	0.86
XGB Classifier	0.97	0.93	0.84	0.88
AdaBoost Classifier	0.96	0.85	0.87	0.86
SVC	0.97	0.93	0.84	0.88

Merging component classes into one class increases the amount of data in the minority class, which improves the conditions for the learning algorithms and increases the number of hyperparameters. The results in Table 3-11 show that all algorithms have acceptable performance in detecting the probability of failure in binary mode and the CNN and LSTM algorithms have the best performance in predicting the failure of machines in the next 7 days for the recall and F1 parameters.

The second step involves extracting the hourly and daily data from the second class, which pertains to reported failures, along with the prediction errors from the first class (also known as "class none"), based on the outcomes obtained in the previous step. The extracted data is used to investigate the features and hyperparameters in two-time windows of 3 hours and 24 hours.

The results of the machine learning and deep learning algorithms are shown in Table 3-12.

Table 3- 10 Performance of ML and DL models for long-term prediction in second step.

ML Model	Accuracy	Precision	Recall	F1
CNN	0.95	0.89	**0.98**	**0.94**
Extra Trees Classifier	0.95	0.89	**0.98**	**0.92**
Stacking Classifier	0.95	0.89	**0.98**	**0.92**
LSTM	0.95	0.89	0.98	0.91
Logistic Regression	0.95	0.88	0.97	0.91
Random Forest Classifier	0.95	0.89	0.98	0.91
Bagging Classifier	0.95	0.88	0.98	0.91
Linear SVC	0.95	0.89	0.97	0.91
Decision Tree Classifier	0.95	0.89	0.98	0.91
XGB Classifier	0.97	0.91	0.98	0.93
SVC	0.95	0.89	0.98	0.91
AdaBoost Classifier	0.95	0.88	0.97	0.91

As expected, changing the classification of the data from 5 classes (comp1, comp2, comp3, comp4, and None) to 4 classes and changing the dataset from an imbalanced dataset to a more balanced dataset significantly improved the results of the machine learning and deep learning algorithms in detecting faults in components. Table 3-12 and 3-13 show that the CNN algorithm performed slightly better than the other algorithms used in this work.

It is assumed that the importance of all four components is the same in terms of cost and performance in the machine, and the features and hyperparameters are selected based only on sensor data and technical data from maintenance, but in a real industrial application, many external factors can affect the selection of features and the final results of the algorithms.

Table 3- 13 Performance of the ML and DL models for the long-term prediction of components in the two-step procedure based on the recall parameter.

ML Model	Comp1	Comp2	Comp3	Comp4
CNN	0.99	0.99	**1.00**	**0.98**
Extra Trees Classifier	1.00	0.99	**0.98**	**0.97**
Stacking Classifier	0.99	0.99	**0.98**	**0.98**
Logistic Regression	0.97	0.99	1.00	0.96
Random Forest Classifier	0.98	0.99	0.98	0.98
Bagging Classifier	0.97	0.99	0.98	0.98
Linear SVC	0.99	0.99	0.98	0.95
Decision Tree Classifier	0.98	0.99	0.98	0.97
XGB Classifier	0.98	0.99	0.99	0.98
SVC	0.96	0.99	0.98	0.97
LSTM	0.97	0.98	0.99	0.98
AdaBoost Classifier	0.95	0.99	0.96	0.95

The comparison of the results shows that the three algorithms CNN, Extra Trees Classifier and Stacking Classifier performed the best. A quick look at Table 3-13 shows that all algorithms predicted the third component better than the other components. The best maintenance prediction for each component for the next seven days was provided by the CNN algorithm, which identified the third component with a recall value of 100%. The findings of the study show that the recall value for predicting the failure of the second component is consistently higher than for the other components. This could be due to the fact that the number of reported failures for this component is higher than for the others. Specifically, there were approximately twice as many reported failures for the third component compared to the second.

Despite this disparity in the number of reported failures, all of the algorithms were still able to predict the probability of failure with an accuracy of 99%, with the exception of the LSTM algorithm. These results highlight the importance of having accurate and comprehensive documentation of equipment failures, as it can provide valuable insights for predictive maintenance strategies.

The ability of the algorithms to accurately predict equipment failures is promising, but it is important to note that they are only one part of an overall maintenance and risk management strategy. Other factors, such as regular inspections and preventive maintenance, should also be considered to ensure the safe and reliable operation of equipment. Nevertheless, the findings of this study emphasize the critical role of documentation in improving equipment reliability and reducing the likelihood of failures.

3.7 Conclusion

In this study, an imbalanced dataset published by Microsoft was used to implement several machine learning and deep learning algorithms. The codes were written in Python and run in the Google Collaboratory environment. The researchers studied 10 different machine learning algorithms and two neural networks (CNN and LSTM). The results showed that the failure prediction for machines and components using machine learning algorithms was excellent, with three algorithms (Random Forest Classifier, Stacking Classifier, and Extra Trees Classifier) predicting failure probability with about 98% accuracy. However, due to the extreme imbalance of the dataset, some failure data were lost when the time window changed by more than 24 hours, which increased the imbalance of the dataset and led to many errors in training the algorithms. To overcome this issue, two methods were used for failure prediction for the next seven days: the weighted average coefficient method and the two-step method.

The weighted average coefficient method used different coefficients to reduce the difference between the minority and majority classes. It was good at predicting the probability of machine failure in the next seven days but poor at determining the failure class (component). In the two-step procedure, deep learning algorithms were also used to study the dataset. The goal of this method is to predict the failure of machines in the first step and the failure probability of each class (component) in the next step by modifying the dataset, features, and hyper-parameters. The results show that the two-step method is able to improve the results of machine learning and deep learning algorithms in general. The study showed that the CNN algorithm provided the best maintenance prediction for each component for the next seven days, with a recall value of 100% for the third component. Additionally, the recall value for predicting the failure of the second component was consistently higher than for the other components, possibly due to the higher number of reported failures for this component. Despite this disparity, all algorithms were still able to predict the probability of failure with an accuracy of 99%, except for the LSTM algorithm.

These results emphasize the importance of maintaining accurate and comprehensive documentation of equipment failures, which can provide valuable insights for predictive maintenance strategies. While the algorithms used in the study show promise for predicting equipment failures, they should not be the only tool used for maintenance and risk management.

Chapter 4

Conclusion and Future Works

The IIoT device for remote monitoring of hydraulic hammer is an innovative and effective solution for improving the performance, reliability, and safety of industrial machinery. Through the integration of advanced sensors, connectivity technologies, and data analytics, the device enables real-time monitoring of key operational parameters such as vibration, temperature, and pressure, which can help detect potential issues before they turn into major problems. Additionally, the device allows for remote monitoring and management of the equipment, enabling maintenance teams to access critical data and make informed decisions from anywhere, at any time. Additionally, data logging can aid in identifying patterns or trends in equipment usage, leading to improved operational efficiency and better decision-making. In harsh environments, where equipment is subject to extreme temperatures, vibrations, and other stresses, a robust data logger can provide critical insights into the health of the equipment and potential issues before they become major problems.

Therefore, chapter 2 presents the design and development of a data logger for remote monitoring of hydraulic hammers based on the integration of sensors, requirements, and convenient platforms for data analysis and maintenance prediction, in collaboration with the Research and Development of INDECO Ind. SpA and Trustedglobal (Trusted A/S) company, all the required and effective parameters were determined by the test bench and the sensors were selected accordingly.

There were many challenges in developing a remote control for hydraulic hammers. One of the most important challenges is defining the design requirements and selecting the sensors. The Indeco bench test was used to determine all the critical parameters for the design and selection of sensors, such as oil pressure, vibration, etc. The next step was to test the selection of sensors according to the design requirements. Another challenge was the selection of the location for the oil pressure sensor and the flow sensor, which was discussed in detail in Chapter 2.

The presented device is one of the best industrial Internet of Things devices for remote monitoring of equipment in harsh environments, which, depending on the application, provides the possibility of productivity monitoring, operation control, data acquisition, and finally the possibility of data verification, analysis, repair prediction and maintenance. The adaptability of these sensors is a significant advantage, indicating potential for numerous future applications, such as usage in challenging environments like mines or at sea. This suggests that point clouds will likely be generated with a greater number of samples, highlighting the need for research on appropriate processing methods to handle the challenges that will arise in the coming years.

Chapter 3 demonstrates the effectiveness of using machine learning and deep learning algorithms in predicting maintenance needs in industrial settings on an imbalanced dataset published by Microsoft. Through the implementation of various models such as Random Forest, XGBoost, LSTM and CNN, we were able to accurately predict equipment failures and maintenance needs, thus minimizing downtime and reducing maintenance costs.

The experiments conducted on real-world datasets show that deep learning algorithms such as CNN and traditional machine learning algorithms, achieving higher accuracy and lower error rates. The results also demonstrate the importance of feature selection and preprocessing techniques in improving the performance of predictive maintenance models.

Moreover, this thesis presents a practical implementation of the predictive maintenance model, providing a user-friendly and accessible platform for maintenance professionals to utilize. This implementation can serve as a valuable resource for companies seeking to improve their maintenance practices and reduce costs.

The research utilized Python and the Google Collaboratory environment to implement 10 distinct machine learning algorithms and two neural networks, namely Convolutional Neural Network and Long Short-Term Memory, in predicting failures for machines and components. The outcomes of the study were remarkable, and the three models, Random Forest Classifier, Stacking Classifier, and Extra Trees Classifier, achieved a failure probability prediction of approximately 98% within the next 24 hours.

However, the data is highly imbalanced, so that some failure data are lost if the time window changes by more than 24 hours. This imbalance leads to numerous errors in the training process of the algorithm. Therefore, two methods were implemented to predict failures within the next seven days: the weighted average coefficient and the two-step method.

The weighted average coefficient approach applied various coefficients to decrease the variation between the majority and minority classes, resulting in good performance in predicting the probability of machine failure within the next seven days. However, this method was not effective in identifying the failure class or component. On the other hand, the two-step method demonstrated superior performance in predicting both the failure probability and the failure class for machines and components. The first step of the method deployed a binary classification algorithm to predict the machine's failure, while the second step used a multiclass algorithm to predict the component failure. The study aimed to compare the effectiveness of different machine learning algorithms in predicting equipment failures for a manufacturing plant. The results showed that the CNN algorithm provided the most accurate maintenance prediction for each component for the next seven days, achieving a recall parameter value of 100% for the third component. This means that the algorithm was able to correctly identify all instances of failures for this component.

Additionally, the recall value for predicting the failure of the second component was consistently higher than for the other components, possibly due to the higher number of reported failures for this component. This finding approves that having more data on equipment failures can lead to more accurate predictions. However, it is important to note that the number of reported failures does not necessarily reflect the actual failure rate, as some failures may go unreported.

Despite the high accuracy of the algorithms in predicting equipment failures, they should not be relied on as the only tool for maintenance and risk management. Other strategies, such as regular inspections, preventive maintenance, and contingency plans, should also be implemented to ensure the safe and reliable operation of equipment. These strategies can complement the predictive capabilities of the algorithms and provide a more comprehensive approach to equipment maintenance and risk management.

The study also highlights the importance of maintaining accurate and comprehensive documentation of equipment failures. This documentation can provide valuable insights for predictive maintenance strategies, such as identifying patterns in equipment failures and predicting future failures. Furthermore, having good documentation can also help to track the maintenance history of equipment and ensure that maintenance is carried out in a timely and effective manner.

The presented work will serve as the foundation for further research and development in:

- Providing standard technical information for different modes of hydraulic hammer.
- Definition of fault detection, health monitoring, predictive maintenance techniques.
- Definition of ITC maintenance systems for on-demand manuals and verification of predictive maintenance operations also with techniques of Augmented Reality (AR).
- Innovative Big Data Analytics and Robust algorithms.
- Consideration of the concept of "feature learning" instead of "feature engineering.
- Investigation of the factors that cause the priority and importance of one component over another in the dataset (e.g., cost, ease of replacement) by combining CNN and LSTM.

Bibliography

[1] Jaidka, H., Sharma, N., & Singh, R. (2020, May). Evolution of iot to iiot: Applications & challenges. In Proceedings of the international conference on innovative computing & communications (ICICC).

[2] Mourtzis, D. (2020). Simulation in the design and operation of manufacturing systems: state of the art and new trends. International Journal of Production Research, 58(7), 1927-1949.

[3] Mourtzis, D., Fotia, S., Boli, N., & Vlachou, E. (2018). An approach for the modelling and quantification of PSS customisation. International Journal of Production Research, 56(3), 1137-1153.

[4] Mourtzis, D., Angelopoulos, J., & Panopoulos, N. (2021). Design and development of an IoT enabled platform for remote monitoring and predictive maintenance of industrial equipment. Procedia Manufacturing, 54, 166-171.

[5] Hermann, M., Pentek, T., & Otto, B. (2015). Design principles for Industrie 4.0 scenarios: a literature review. Technische Universität Dortmund, Dortmund, 45.

[6] Masoni, R., Ferrise, F., Bordegoni, M., Gattullo, M., Uva, A. E., Fiorentino, M., ... & Di Donato, M. (2017). Supporting remote maintenance in industry 4.0 through augmented reality. Procedia manufacturing, 11, 1296-1302.

[7] Sala, R., Zanetti, V., Pezzotta, G., & Cavalieri, S. (2017, June). The role of technology in designing and delivering Product-service Systems. In 2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC) (pp. 1255-1261). IEEE.

[8] Kulkarni, K., Devi, U., Sirighee, A., Hazra, J., & Rao, P. (2018, June). Predictive maintenance for supermarket refrigeration systems using only case temperature data. In 2018 Annual American Control Conference (ACC) (pp. 4640-4645). IEEE.

[9] Aboelmaged, M. G. (2014). Predicting e-readiness at firm-level: An analysis of technological, organizational and environmental (TOE) effects on e-maintenance readiness in manufacturing firms. International Journal of Information Management, 34(5), 639-651.

[10] Holmberg, K., Adgar, A., Arnaiz, A., Jantunen, E., Mascolo, J., & Mekid, S. (Eds.). (2010). E-maintenance. Springer Science & Business Media.

[11] Mobley, R. K. (2002). An introduction to predictive maintenance. Elsevier.

[12] Kobbacy, K. A., & Murthy, D. P. (2008). Complex system maintenance handbook (pp. 417-436). London: Springer.

[13] Taisch, M., Casidsid, ML., & Despeisse, M., Luglietti, R., & May, G. (Eds.). (2019). World Manufacturing Foundation.

[14] Bousdekis, A., Lepenioti, K., Apostolou, D., & Mentzas, G. (2019). Decision making in predictive maintenance: literature review and research agenda for industry 4.0. IFAC-PapersOnLine, 52(13), 607-612.

[15] Paolanti, M., Romeo, L., Felicetti, A., Mancini, A., Frontoni, E., & Loncarski, J. (2018, July). Machine learning approach for predictive maintenance in industry 4.0. In 2018 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA) (pp. 1-6). IEEE.

[16] Li, Y., & Wang, K. (2020). Modified convolutional neural network with global average pooling for intelligent fault diagnosis of industrial gearbox. Eksploatacja i Niezawodność, 22(1), 63-72.

[17] RodRigues, J., Costa, I., Farinha, J. T., Mendes, M., & Margalho, L. (2020). Predicting motor oil condition using artificial neural networks and principal component analysis. Eksploatacja i Niezawodność, 22(3).

[18] Microsoft. Predictive Maintenance Modelling Guide. 2018. Available online:https://notebooks.azure.com/Microsoft/projects/ PredictiveMaintenance (accessed on 1 May 2020).

[19] Heidarpour, F., Ciccolella, A., & Uva, A. E. (2022, September). Design and Development of an IoT Enabled Device for Remote Monitoring of Hydraulic Hammers. In Advances on Mechanics, Design Engineering and Manufacturing IV: Proceedings of the International Joint

Conference on Mechanics, Design Engineering & Advanced Manufacturing, JCM 2022, June 1-3, 2022, Ischia, Italy (pp. 390-398). Cham: Springer International Publishing.

[20] Hofmann, E., & Rüsch, M. (2017). Industry 4.0 and the current status as well as future prospects on logistics. Computers in industry, 89, 23-34.

[21] Kagermann, H., Wahlster, W., & Helbig, J. (2013). Recommendations for implementing the strategy initiative Industrie 4.0. Securing the Future of German Manufacturing Industry [Final Report of the Industrie 4.0. Working Group]. Acatech–National Academy of Science and Engineering, 4-7.

[22] Lasi, H., Fettke, P., Kemper, H. G., Feld, T., & Hoffmann, M. (2014). Industry 4.0. Business & information systems engineering, 6, 239-242.

[23] Sassanelli, C., Pezzotta, G., Rossi, M., Terzi, S., & Cavalieri, S. (2015). Towards a Lean Product Service Systems (PSS) Design: state of the art, opportunities and challenges. Procedia CIRP, 30, 191-196.

[24] Kim, S., Son, C., Yoon, B., & Park, Y. (2015). Development of an innovation model based on a service-oriented product service system (PSS). Sustainability, 7(11), 14427-14449.

[25] Andersson, H., Simonsson, K., Hilding, D., Schill, M., Sigfridsson, E., & Leidermark, D. (2019). Validation of a co-simulation approach for hydraulic percussion units applied to a hydraulic hammer. Advances in Engineering Software, 131, 102-115.

[26] Gorodilov, L. V., Kudryavtsev, V. G., & Pashina, O. A. (2011). Experimental research stand and procedure for hydraulic percussion systems. Journal of Mining Science, 47, 778-786.

[27] Alimov, O. D., Basov, S. A., & Lavendel, Ė. Ė. (1990). Gidravlicheskie vibroudarnye sistemy. Nauka.

[28] Ashavskiy, A. M. (1966). Osnovy proektirovaniya optimal’nykh parametrov zaboinykh burovykh mashin. Design Engineering for Optimal Parameters of Heading Boring Machines), Moscow: Nedra.

[29] Gorodilov, L. V., Kudryavtsev, V. G., & Pashina, O. A. (2011). Experimental research stand and procedure for hydraulic percussion systems. Journal of Mining Science, 47, 778-786.

[30] Gorbunov, V. F., Lazutkin, A. G., & Ushakov, L. S. (1986). Impulse Hydraulic Drive for Mining Machines. Novosibirsk: Science, 1986.-197 P.

[31] AyalaCarcedo, F. (2017). Drilling and blasting of rocks. Routledge.

[32] Gorodilov, L. V. (2020). Basic properties of one-way action hydraulic percussion system with two piston arresters. Journal of Mining Science, 56(6), 972-981.

[33] Melamed, Y., Kiselev, A., Gelfgat, M., Dreesen, D., & Blacic, J. (2000). Hydraulic hammer drilling technology: developments and capabilities. J. Energy Resour. Technol., 122(1), 1-7.

[34] Lee, S., Tewolde, G., & Kwon, J. (2014, March). Design and implementation of vehicle tracking system using GPS/GSM/GPRS technology and smartphone application. In 2014 IEEE world forum on internet of things (WF-IoT) (pp. 353-358). IEEE.

[35] da Silva, A. F., Ohta, R. L., dos Santos, M. N., & Binotto, A. P. (2016). A cloud-based architecture for the internet of things targeting industrial devices remote monitoring and control. Ifac-Papersonline, 49(30), 108-113.

[36] Gao, H., Qin, X., Barroso, R. J. D., Hussain, W., Xu, Y., & Yin, Y. (2020). Collaborative learning-based industrial IoT API recommendation for software-defined devices: the implicit knowledge discovery perspective. IEEE Transactions on Emerging Topics in Computational Intelligence, 6(1), 66-76.

[37] Gong, W., Zhang, W., Bilal, M., Chen, Y., Xu, X., & Wang, W. (2021). Efficient web APIs recommendation with privacy-preservation for mobile app development in industry 4.0. IEEE Transactions on Industrial Informatics, 18(9), 6379-6387.

[38] Carnero, M. (2006). An evaluation system of the setting up of predictive maintenance programmes. Reliability Engineering & System Safety, 91(8), 945-963.

[39] Hogxia, W., Xiaohui, Y., & Ming, Y. (2016). Study on predictive maintenance strategy. International Journal of u-and e-Service, Science and Technology, 9(4), 295-300.

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

[41] Aydın, İ., Karaköse, M., & Akın, E. (2015). Combined intelligent methods based on wireless sensor networks for condition monitoring and fault diagnosis. Journal of Intelligent Manufacturing, 26, 717-729.

[42] Zhang, W., Jia, M. P., Zhu, L., & Yan, X. A. (2017). Comprehensive overview on computational intelligence techniques for machinery condition monitoring and fault diagnosis. Chinese Journal of Mechanical Engineering, 30, 782-795.

[43] Wikipedia contributors. (2023). Bathtub curve. Wikipedia. https://en.wikipedia.org /wiki/ Bathtub_curve

[44] Scheffer, C., & Girdhar, P. (2004). Practical machinery vibration analysis and predictive maintenance. Elsevier.

[45] Liao, L., & Köttig, F. (2016). A hybrid framework combining data-driven and model-based methods for system remaining useful life prediction. Applied Soft Computing, 44, 191-199.

[46] Blancke, O., Combette, A., Amyot, N., Komljenovic, D., Lévesque, M., Hudon, C., & Zerhouni, N. (2018, July). A predictive maintenance approach for complex equipment based on petri net failure mechanism propagation model. In PHM Society European Conference (Vol. 4, No. 1).

[47] Li, L., Liu, M., Shen, W., & Cheng, G. (2017). An expert knowledge-based dynamic maintenance task assignment model using discrete stress–strength interference theory. Knowledge-Based Systems, 131, 135-148.

[48] Li, Y., Kurfess, T. R., & Liang, S. Y. (2000). Stochastic prognostics for rolling element bearings. Mechanical Systems and Signal Processing, 14(5), 747-762.

[49] Venkatasubramanian, V., Rengaswamy, R., Yin, K., & Kavuri, S. N. (2003). A review of process fault detection and diagnosis: Part I: Quantitative model-based methods. Computers & chemical engineering, 27(3), 293-311.

[50] Baptista, M., Sankararaman, S., de Medeiros, I. P., Nascimento Jr, C., Prendinger, H., & Henriques, E. M. (2018). Forecasting fault events for predictive maintenance using data-driven techniques and ARMA modeling. Computers & Industrial Engineering, 115, 41-53.

[51] Park, J., Kim, S., Choi, J. H., & Lee, S. H. (2021). Frequency energy shift method for bearing fault prognosis using microphone sensor. Mechanical Systems and Signal Processing, 147, 107068.

[52] Xiao, D., Huang, Y., Zhang, X., Shi, H., Liu, C., & Li, Y. (2018, October). Fault diagnosis of asynchronous motors based on LSTM neural network. In 2018 prognostics and system health management conference (PHM-Chongqing) (pp. 540-545). IEEE.

[53] Zhao, F., Tian, Z., Liang, X., & Xie, M. (2018). An integrated prognostics method for failure time prediction of gears subject to the surface wear failure mode. IEEE Transactions on Reliability, 67(1), 316-327.

[54] Canizo, M., Onieva, E., Conde, A., Charramendieta, S., & Trujillo, S. (2017, June). Real-time predictive maintenance for wind turbines using Big Data frameworks. In 2017 ieee international conference on prognostics and health management (icphm) (pp. 70-77). IEEE.

[55] Gao, Z., Cecati, C., & Ding, S. X. (2015). A survey of fault diagnosis and fault-tolerant techniques—Part I: Fault diagnosis with model-based and signal-based approaches. IEEE transactions on industrial electronics, 62(6), 3757-3767.

[56] Zhang, W., Yang, D., & Wang, H. (2019). Data-driven methods for predictive maintenance of industrial equipment: A survey. IEEE Systems Journal, 13(3), 2213-2227.

[57] Lee, J., Kao, H. A., & Yang, S. (2014). Service innovation and smart analytics for industry 4.0 and big data environment. Procedia cirp, 16, 3-8.

[58] Bagheri, B., Yang, S., Kao, H. A., & Lee, J. (2015). Cyber-physical systems architecture for self-aware machines in industry 4.0 environment. IFAC-PapersOnLine, 48(3), 1622-1627.

[59] Sreedharan, V. R., & Unnikrishnan, A. (2017). Moving towards industry 4.0: a systematic review. International Journal of Pure and Applied Mathematics, 117(20), 929-936.

[60] Nunes, P., Santos, J., & Rocha, E. (2023). Challenges in predictive maintenance–A review. CIRP Journal of Manufacturing Science and Technology, 40, 53-67.

[61] Géron, A. (2022). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow. " O'Reilly Media, Inc.".

[62] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12, 2825-2830.

[63] Kanawaday, A., & Sane, A. (2017, November). Machine learning for predictive maintenance of industrial machines using IoT sensor data. In 2017 8th IEEE international conference on software engineering and service science (ICSESS) (pp. 87-90). IEEE.

[64] Seiffert, C., Khoshgoftaar, T. M., Van Hulse, J., & Napolitano, A. (2008, December). RUSBoost: Improving classification performance when training data is skewed. In 2008 19th international conference on pattern recognition (pp. 1-4). IEEE.

[65] Wu, Y., Ding, Y., & Feng, J. (2020). SMOTE-Boost-based sparse Bayesian model for flood prediction. EURASIP Journal on Wireless Communications and Networking, 2020(1), 1-12.

[66] McCarthy, J., Minsky, M. L., Rochester, N., & Shannon, C. E. (2006). A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955. AI magazine, 27(4), 12-12.

[67] Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding machine learning: From theory to algorithms. Cambridge university press.

[68] Diez-Olivan, A., Del Ser, J., Galar, D., & Sierra, B. Data fusion and machine learning for industrial prognosis: Trends and perspectives towards Industry 4.0. Information Fusion, 50, 92-111, 2019.

[69] Diez-Olivan, A., Del Ser, J., Galar, D., & Sierra, B. (2019). Data fusion and machine learning for industrial prognosis: Trends and perspectives towards Industry 4.0. Information Fusion, 50, 92-111.

[70] Panda, S. K., & Jana, P. K. (2015). Efficient task scheduling algorithms for heterogeneous multi-cloud environment. The Journal of Supercomputing, 71, 1505-1533.

[71] Patro, S. G. O. P. A. L., & Sahu, K. K. Patro, S. G. O. P. A. L., & Sahu, K. K. (2015). Normalization: A preprocessing stage. arXiv preprint arXiv:1503.06462.

[72] Pal, M. (2005). Random forest classifier for remote sensing classification. International journal of remote sensing, 26(1), 217-222.

[73] Kumar, M., & Kumar, M. (2021). XGBoost: 2D-object recognition using shape descriptors and extreme gradient boosting classifier. In Computational Methods and Data Engineering: Proceedings of ICMDE 2020, Volume 1 (pp. 207-222). Springer Singapore.

[74] Maalouf, M. (2011). Logistic regression in data analysis: an overview. International Journal of Data Analysis Techniques and Strategies, 3(3), 281-299.

[75] Abhishek, L. (2020, June). Optical character recognition using ensemble of SVM, MLP and extra trees classifier. In 2020 International Conference for Emerging Technology (INCET) (pp. 1-4). IEEE.

[76] Skurichina, M., & Duin, R. P. (2002). Bagging, boosting and the random subspace method for linear classifiers. Pattern Analysis & Applications, 5, 121-135.

[77] Alam, S., Sonbhadra, S. K., Agarwal, S., & Nagabhushan, P. (2020). One-class support vector classifiers: A survey. Knowledge-Based Systems, 196, 105754.

[78] Ghosh, S., Dasgupta, A., & Swetapadma, A. (2019, February). A study on support vector machine based linear and non-linear pattern classification. In 2019 International Conference on Intelligent Sustainable Systems (ICISS) (pp. 24-28). IEEE.

[79] Sesmero, M. P., Ledezma, A. I., & Sanchis, A. (2015). Generating ensembles of heterogeneous classifiers using stacked generalization. Wiley interdisciplinary reviews: data mining and knowledge discovery, 5(1), 21-34.

[80] Rojas, R. (2009). AdaBoost and the super bowl of classifiers a tutorial introduction to adaptive boosting. Freie University, Berlin, Tech. Rep.

[81] Kingsford, C., & Salzberg, S. L. (2008). What are decision trees?. Nature biotechnology, 26(9), 1011-1013.

[82] Kattenborn, T., Leitloff, J., Schiefer, F., & Hinz, S. (2021). Review on Convolutional Neural Networks (CNN) in vegetation remote sensing. ISPRS journal of photogrammetry and remote sensing, 173, 24-49.

[83] Ossher, H., Harrison, W., & Tarr, P. (2000, May). Software engineering tools and environments: a roadmap. In Proceedings of the Conference on the Future of Software Engineering (pp. 261-277).

[84] Prado, F. F., & Digiampietri, L. A. (2020, November). A systematic review of automated feature engineering solutions in machine learning problems. In XVI Brazilian Symposium on Information Systems (pp. 1-7).

[85] Halder, A., Dey, D., & Sadhu, A. K. (2020). Lung nodule detection from feature engineering to deep learning in thoracic CT images: a comprehensive review. Journal of digital imaging, 33(3), 655-677.

[86] Isabelle, G., Steve, G., Massoud, N., & Lotfi, Z. (2006). Feature Extraction: Foundations and Applications. Pattern Recognition.

[87] Ren, L., Sun, Y., Wang, H., & Zhang, L. (2018). Prediction of bearing remaining useful life with deep convolution neural network. IEEE access, 6, 13041-13049.

[88] Schmidhuber, J., & Hochreiter, S. (1997). Long short-term memory. Neural Comput, 9(8), 1735-1780.

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

[90] Poosapati, V., Katneni, V., Manda, V. K., & Ramesh, T. L. V. (2019). Enabling cognitive predictive maintenance using machine learning: Approaches and design methodologies. In Soft Computing and Signal Processing: Proceedings of ICSCSP 2018, Volume 2 (pp. 37-45). Springer Singapore.

Acknowledgements

I would like to express my deepest gratitude to my advisors, Professor Antonio E. Uva, Professor Artūras Serackis and Alessandro Ciccolella, for their invaluable guidance, support, and encouragement throughout my PhD journey. Without their patience, expertise, and unwavering commitment, this work would not have been possible.

I would also like to extend my thanks to the members of my thesis committee, for their constructive feedback, insightful comments, and helpful suggestions, which have significantly improved the quality of my research.

Finally, I would like to thank my family, for their unconditional love, unwavering support, and understanding, which have sustained me throughout this challenging journey.

Thank you all for your contributions to my academic and personal growth.