PREDICTION OF METROPT APU PERFORMANCE THROUGH IOT-ENABLED PREDICTIVE MAINTENANCE WITH CNN-LSTM TECHNIQUES

2024
MASTER THESIS
COMPUTER ENGINEERING

Shahad Jameel Farhan ALSAID

Thesis Advisor
Prof. Dr. Oğuz FINDIK

PREDICTION OF METROPT APU PERFORMANCE THROUGH IOT-ENABLED PREDICTIVE MAINTENANCE WITH CNN-LSTM TECHNIQUES

Shahad Jameel Farhan ALSAID

Thesis Advisor
Prof. Dr. Oğuz FINDIK

T.C.
Karabuk University
Institute of Graduate Programs
Department of Computer Engineering

Prepared as
Master Thesis

KARABÜK
January 2024

I certify that in my opinion the thesis submitted by Shahad Jameel Farhan ALSAID titled "PREDICTION OF METROPT APU PERFORMANCE THROUGH IOT-ENABLED PREDICTIVE MAINTENANCE WITH CNN-LSTM TECHNIQUES" is fully adequate in scope and in quality as a thesis for the degree of Master of Science.

Prof. Dr. Oğuz FINDIK ....

Thesis Advisor, Department of Computer Engineering

This thesis is accepted by the examining committee with a unanimous vote in the Department of Computer Engineering as a Master of Science thesis. 17.1.2024

Examining Committee Members (Institutions)		Signature
Chairman	: Prof. Dr. Emre ÇOMAK (MAKÜ)	....
Member	: Prof. Dr. Oğuz FINDIK. (KBU)	....
Member	: Assist. Prof. Dr. Kürşat M. KARAOĞLAN (KBU)	....

The degree of Master of Science by the thesis submitted is approved by the Administrative Board of the Institute of Graduate Programs, Karabük University.

Assoc. Prof. Dr. Zeynep ÖZCAN ....

Director of the Institute of Graduate Programs

"I declare that all the information within this thesis has been gathered and presented in accordance with academic regulations and ethical principles, and I have, according to the requirements of these regulations and principles, cited all those which do not originate in this work as well."

Shahad Jameel Farhan ALSAID

ABSTRACT

M. Sc. Thesis

PREDICTION OF METROPT APU PERFORMANCE THROUGH IOT-ENABLED PREDICTIVE MAINTENANCE WITH CNN-LSTM TECHNIQUES

Shahad Jameel Farhan ALSAID

Karabük University
Institute of Graduate Programs
The Department of Computer Engineering

Thesis Advisor:
Prof. Dr. Oğuz FINDIK
January 2024, 80 pages

This study represents a pioneering leap in predictive maintenance by harnessing the power of deep learning. Focused on the MetroPT dataset and its intricate APU (Air Production Unit) metrics from a train vehicle, the research meticulously processed and engineered features for binary and multi-class analysis. The centerpiece of this work is the groundbreaking CNN-LSTM algorithm, meticulously crafted to excel in both classification paradigms. The empirical findings are nothing short of exceptional: an impressive 92% accuracy for binary classification and an outstanding 99.5% accuracy for multi-class prediction. Beyond its immediate impact on predictive maintenance, this research serves as a beacon, showcasing the transformative potential of deep learning methodologies in fortifying the reliability and efficiency of critical infrastructure maintenance systems, marking a substantial stride in the fusion of artificial intelligence and industrial upkeep.

**Key Words : ** CNN, Deep Learning, LSTM, Predictive Maintenance, MetroPT.
Science Code : 92432

ÖZET

Yüksek Lisans Tezi

METROPT APU PERFORMANSININ CNN-LSTM TEKNİKLERİYLE IOT-ETKİN ÖNGÖRÜCÜ BAKIM YOLUYLA TAHMİNİ

Shahad Jameel Farhan ALSAID

Karabük Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı:
Prof. Dr. Oğuz FINDIK

Ocak 2024, 80 sayfa

Bu çalışma, derin öğrenmenin gücünü kullanarak tahminsel bakım konusunda çığır açan bir adımı temsil ediyor. MetroPT veri kümesine odaklanan ve tren aracındaki karmaşık Hava Üretim Ünitesi (APU) ölçümlerini inceleyen araştırma, özellikleri ikili ve çoklu sınıflı analiz için özenle işlendi ve mühendislikle şekillendirmiştir. Bu çalışmanın odağında, her iki sınıflandırma paradigmasında da başarılı olacak şekilde özenle oluşturulmuş çığır açan CNN-LSTM algoritması bulunmaktadır. Deneysel bulgular sonucunda ikili sınıflandırma için %92 doğruluk ve çoklu sınıf tahminleri için %99.5 doğruluk elde edilmiştir. Bu araştırmanın tahminsel bakım üzerindeki doğrudan etkisinin ötesinde, bu çalışma, derin öğrenme metodolojilerinin kritik altyapı bakım sistemlerinin güvenilirliğini ve verimliliğini sağlamlaştırmada yapabileceği dönüştürücü potansiyeli sergileyen bir işaret olarak hizmet ediyor, yapay zeka ile endüstriyel bakımın birleşiminde önemli bir adımı temsil ediyor.

Anahtar Sözcükler: CNN, Derin Öğrenme, LSTM, Tahminsel Bakım, MetroPT.
Bilim Kodu : 92432

ACKNOWLEDGMENT

My profound gratitude to my advisor: “Prof. Dr. Oğuz FINDIK”, for his invaluable guidance and unwavering support throughout the entire process of preparing this thesis. I am sincerely thankful for his enduring interest and assistance.

Much heartfelt gratitude to my father: “Jameel FARHAN”, and my mother: “Sumyah MOHAMMED” for their unwavering support and encouragement have been instrumental in the successful completion of this thesis.

I would also like to extend my appreciation to Dr. Baktygul Assanova, Dr. Saltanat Sharmukhanbet, and Nurgul Baitemirova for their invaluable contributions.

CONTENTS

Page

<u>**Page**</u>

LIST OF FIGURES

Page

LIST OF TABLES

Page

INDEX OF SYMBOLS AND ABBREVIATIONS

ABBREVIATIONS

ADCNN	: Attention Dense Convolutional Neural Network
AI	: Artificial Intelligence
APU	: Air Production Unit
BPNN	: Back Propagation Neural Network
CBM	: Condition-Based Maintenance
CNN	: Convolutional Neural Networks
CPS	: Cyber-Physical System
CrM	: Corrective Maintenance
CWT	: Continuous Wavelet Transform
DBN	: Deep Belief Networks
DL	: Deep Learning
DNN	: Deep Neural Networks
ENN	: Elman Neural Network
FNN	: Fuzzy Neural Network
GA	: Genetic Algorithms
GBDT	: Gradient Boosted Decision Tree
GRU	: Gated Recurrent Unit
ICT	: Information And Communication Technology
IIoT	: Industrial Internet of Things
IoT	: Internet of Things
IoV	: The Internet of Vehicles
LSTM	: Long Short-Term Memory
MCSA	: Motor Current Signature Analysis
ML	: Machine Learning
PdM	: Predictive Maintenance
PHM	: Prognostics and Health Management

PNN	: Probabilistic Neural Network
PSO	: Particle Swarm Optimization
PvM	: Preventive Maintenance
RF	: Random Forest
RNN	: Recurrent Neural Networks
RVM	: Relevance Vector Machine
SCIM	: Squirrel Cage Induction Motors
SVM	: Support Vector Machine
WNN	: Wavelet Neural Network
FCNN	: Fully Connected Neural Networks
CWRU	: Case Western Reserve University

PART 1

INTRODUCTION

1.1. BACKGROUND

The Internet of Vehicles (IoV) is a beacon of transformative technology in transportation. IoV data enhances traffic management and road safety through machine learning and deep learning. Challenges exist in centralized machine learning methods, limiting their scalability in IoT deployment. IoV's intelligent vehicles, connected to sensors, establish wireless links with infrastructure, cars, and devices, ensuring safer roads [1]. Integrating the IoV within the Cyber-Physical System (CPS) framework represents a groundbreaking evolution in transportation technology. Integrating sensors, computation, control, and networking into vehicles and infrastructure, IoV facilitates intelligent decision-making through analytical algorithms, enhancing safety, traffic management, and cost efficiency [2]. This amalgamation optimizes delivery routes and vehicle maintenance for companies and revolutionizes the passenger experience, providing in-car entertainment, information access, and Internet connectivity. Moreover, IoV holds the potential to significantly reduce emissions and environmental impact by fostering more efficient transportation systems. Concurrently, the rise of the Internet of Things (IoT) has revolutionized interconnectivity across diverse fields, such as manufacturing and transportation [3], [4]. However, deploying Industrial Internet of Things (IIoT) frameworks presents multifaceted challenges, encompassing cybersecurity, scalability, interoperability, and energy efficiency [5,6].

In the contemporary industrial landscape, manufacturing systems have transformed remarkably, integrating capabilities to monitor, control, and communicate within their environment. This evolution has facilitated seamless.

Machine-to-machine and human-to-machine communication while also fortifying industrial machinery against potential failures. Through these advancements, a paradigm of intelligent manufacturing has emerged, significantly driven by machine learning models [7]. One particularly noteworthy application in this domain is predictive maintenance, a strategic approach widely embraced across diverse industries. Despite challenges in integrating Predictive Maintenance (PdM) with the IIoT, its adoption remains prevalent due to its effectiveness in mitigating uncertainties within industrial settings, enhancing operational efficiency, and reducing downtime [8].

The primary objective of PdM is to systematically monitor machinery conditions to avert costly breakdowns and execute maintenance only when genuinely warranted. The evolution of PdM traces its roots to the era of manual visual inspections of machines. [9], [10]. Typically, IoT-enabled PdM systems encompass several integral stages, including data collection, pre-processing, and constructing models for fault diagnosis and prognosis, culminating in providing decision support for maintaining industrial machinery. The imperative for implementing predictive maintenance models in the industry is now fundamentally underscored [11]. PdM, positioned as a forward-looking strategy to identify machinery failures preemptively, holds substantial potential in curtailing industrial expenditures and extending the operational lifespan of equipment. As a proactive diagnostic approach, PdM encourages diverse industrial facilities to monitor their systems in real-time actively, thereby augmenting overall operational efficiency [12]. Concurrently, diagnostic methods play a cornerstone in discerning fault types through comprehensive assessments of machinery's current status [13]. Furthermore, PdM assumes a crucial role in elevating prediction accuracy, aligning with the precision of predictions to enhance efficiency in industry-specific applications [14]. In a concerted effort toward sustainable asset preservation, industries adopt diverse machine-learning techniques to achieve precise predictions for safeguarding industrial assets [15].

Deep Learning (DL), a subset of artificial intelligence, has emerged as a powerful instrument for crafting intelligent algorithms across diverse applications. Rooted in its inspiration from the human nervous system and brain structure, DL can effectively manage high-dimensional and multivariate datasets, rendering it an appealing methodology for practitioners in PdM applications. The augmentation of layers and neurons within DL models enhances their proficiency in unsupervised learning, particularly in tackling more intricate problems. Prominent examples of DL algorithms encompass Deep Neural Networks (DNN), Convolutional Neural Networks (CNN), Deep Belief Networks (DBN), and Recurrent Neural Networks (RNN). However, it is imperative to note that the efficacy of DL algorithms hinges on the judicious selection of the appropriate DL technique tailored to the intricacies of a given problem [16], [17].

The predictive capacity to anticipate machinery failures confers substantial benefits by mitigating costs linked to unscheduled maintenance and minimizing consequential downtime, exerting a noteworthy financial impact on businesses. Consequently, researchers and developers have a dedicated commitment to augment prediction models, aiming to elevate accuracy and performance in anticipation of device errors prior to manifestation. This proactive approach facilitates preemptive replacement or repair, thereby ensuring continuous operations and minimizing disruptions.

1.2. PROBLEM STATEMENT

The industrial and transportation services sector, particularly vehicle fleets, grapples with significant challenges in maintaining its equipment and assets. These difficulties arise from the high costs and disruptions caused by unforeseen breakdowns, leading to increased downtime and subsequent material and human losses. The conventional preventive and corrective maintenance approaches pose financial strains on companies, diminishing productivity. Leveraging the advancements in technology and the growing influence of the IoT, PdM emerges as an enticing solution for businesses, industrial entities, and vehicle fleets. This research aims to improve maintenance efficiency while reducing costs by developing accurate predictive models to detect equipment and asset malfunctions early. These models empower companies to perform timely maintenance, optimizing the utilization of their valuable assets.

1.3. THESIS OBJECTIVE

The pursuit of PdM in the anticipation of failures within industrial vehicles and equipment is underscored by a set of overarching objectives:

- Cost Reduction and Enhanced Efficiency: The foremost objective resides in reducing expenses associated with preventive and corrective maintenance through the attenuation of scheduled and unwarranted equipment maintenance. By proactively forecasting potential breakdowns, companies can optimize the allocation of maintenance resources, thereby yielding cost savings and diminishing the risk exposure for maintenance personnel.
- Increased Productivity and Enhanced Production Efficiency: Through the minimization of unplanned downtime attributed to breakdowns, companies can fortify productivity and enhance the operational efficiency of their products, thereby ensuring alignment with customer demands.
- Enhanced Safety: The capacity to preemptively detect malfunctions constitutes a pivotal factor in enhancing safety within work environments, concurrently diminishing the risks of accidents associated with industrial assets and equipment. Additionally, this capability extends to ensuring occupant safety in the context of vehicular operations.
- Reputation Enhancement: By avoiding unscheduled disruptions and delays, companies can uphold their reputation and cultivate heightened customer trust, particularly in enterprises heavily reliant on vehicle fleets for service delivery.
- Advancing Sustainability: The prevention of breakdowns and interruptions assumes a pivotal role in the diminution of waste and the extension of asset lifespans. This, in turn, contributes substantially to heightened sustainability and a concomitant reduction in environmental footprint.

1.4. THESIS SIGNIFICANCE

In the context of PdM leveraging deep learning models, this study makes several notable contributions:

- Meticulous analysis and pre-processing steps applied to the MetroPT dataset [18].
- Adept execution of feature extraction and selection procedures is demonstrated.
- Involves the construction of a deep learning model employing the CNN-LSTM algorithm.
- The research attains commendable outcomes, exhibiting a level of satisfaction in comparison to findings by preceding researchers.

1.5. THESIS SCOPE

The present study intricately explores the application of deep learning algorithms in predictive maintenance, concentrating specifically on the MetroPT dataset utilized within the context of an Air Production Unit (APU) in a train vehicle. The methodological framework centers on the application of feature engineering to construct binary-class and multi-class categories, employing the CNN-LSTM algorithm for comprehensive analysis and prediction.

1.6. THESIS OUTLINES

This thesis is organized as follows: **PART 2.** provides an exhaustive review of PdM, elucidating its implications of the Industrial Revolution and delineating the various stages of its development. **PART 3.** explains the proposed model and its associated algorithms, concurrently exploring the dataset employed and detailing the methodologies applied for its analysis and processing. In **PART 4**, insights into the hyperparameters used are provided, accompanied by the presentation of experimental results based on the dataset, featuring a comparative analysis against prior research. This section also underscores the current study's limitations and outlines potential directions for future research. Finally, **PART 5.** encapsulates a comprehensive summary of the user data analysis, outlines future perspectives, draws conclusive findings, and furnishes a concluding overview of the thesis.

PART 2

LITERATURE REVIEW

2.1. INDUSTRY 4.0 AND PREDICTIVE MAINTENANCE

Industrial enterprises have undergone transformative phases across various industrial revolutions, from the advent of steam engines in the 18th century to the current era of digitalization technology. The inauguration of steam engines heralded the onset of "Industry 1.0," also known as the mechanical revolution, entailing the transition from manual production to the adoption of steam and water-powered machinery. Subsequently, "Industry 2.0," or the electric process, emerged in the 19th century, coinciding with the discovery of electricity and the introduction of assembly lines. The Third Industrial Revolution, denoted as "Industry 3.0" or the "Digital Revolution," transformed mechanical and analog systems into digital frameworks. This era witnessed notable advancements in computers, microprocessors, digital cellular phones, and the Internet, ultimately automating production processes without human intervention. Industrial production is at the threshold of a new revolution, often called "Industry 4.0." This revolution integrates Internet technologies into industrial manufacturing processes, management, and strategies. Industry 4.0 introduces flexibility and adaptability to manufacturing systems, departing from traditional production methods. Fundamental concepts such as the IIoT play a base core in industrial environments, enhancing process performance, safety, reliability, and efficiency. This involves the collection of sensor data and its transformation into actionable information through the utilization of cost-effective big data analytics tools [19], [20].

Moreover, Industry 4.0 integrates CPS to oversee and regulate operations through feedback loops. The architecture of CPS comprises five distinct levels, encompassing intelligent communication, the conversion of data to information, the Internet, Perception, and formation [21]. Cloud computing assumes a pivotal role in the ongoing evolution of Industry 4.0 by facilitating data storage and access through the Internet [22]. Industry 4.0 extends numerous advantages to manufacturing enterprises, encompassing cost savings on initial Information and Communication Technology (ICT) infrastructure, accelerated application speeds, enhanced management capabilities, diminished maintenance requirements, and heightened adaptability of ICT resources. Figure 2.1 presents the evolution of the industry [23] .

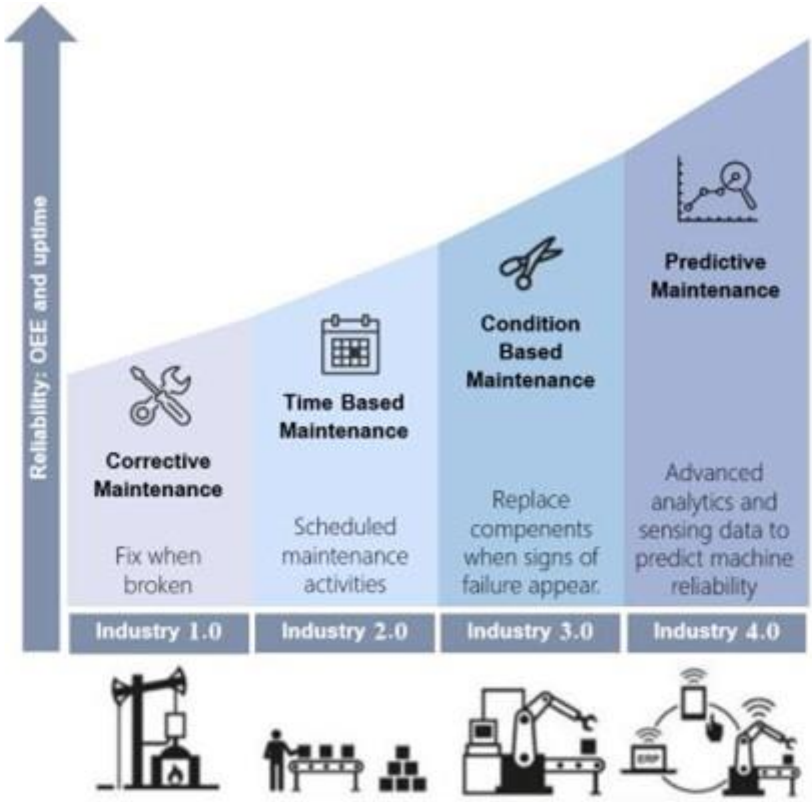

Figure 2.1. Evolution of the maintenance paradigm within the context of industrial revolutions [23].

In addition to traditional concepts, Artificial Intelligence (AI) techniques, including Machine Learning (ML) and Deep Learning (DL), are employed to enable computers to learn from data patterns, eliminating the need for explicit programming. These techniques enhance various aspects of manufacturing processes, such as maintenance, scheduling, and quality control, improving decision support and forecasting methods. Particularly impactful in Condition-Based Maintenance (CBM), these AI methods utilize real-time sensor data to recommend maintenance actions only when performance degradation or impending failures are detected, departing from predetermined maintenance intervals. Amid recent technological advancements, manufacturing systems have evolved through four industrial revolutions, bringing about changes in maintenance functions. Maintenance, as a strategic approach, involves technical, administrative, and organizational procedures to preserve or restore an item's intended function throughout its lifecycle [24]. Maintenance procedures are designed not only to mitigate breakdowns and prolong the lifespan of components but also to decrease operating costs associated with maintenance, allocate resources efficiently, and minimize downtime. Maintenance strategies are typically categorized into three distinct types [25]:

- Corrective Maintenance (CrM): is commonly applied solely in equipment malfunction or failure, to restore the equipment to a functional state. It is implemented after the breakdown of the entire device or any components. Representing the most straightforward maintenance strategy, it is well-suited for non-critical assets. This strategy may be employed in scenarios where unplanned equipment malfunctions result in additional costs and production delays.
- Preventive Maintenance (PvM): is a scheduled or pre-planned maintenance strategy designed to prevent equipment failures proactively. While generally more cost-effective than corrective maintenance, it may sometimes lead to unnecessary maintenance or part replacements, incurring additional costs for the plant.
- Predictive Maintenance (PdM): The proliferation of sensor-generated data and advancements in the industrial sector have rendered machine and deep learning algorithms valuable tools for analyzing extensive datasets and identifying hidden patterns. Research on condition-based predictive maintenance has gained prominence in recent years, with machine and deep learning models extensively utilized in the PdM and demonstrating satisfactory performance [26]. Using predictive tools, precisely machine, and deep learning techniques, this strategy anticipates optimal maintenance timing by leveraging historical equipment or component data [27]. Precise failure predictions mitigate challenges such as wasted time and additional costs, ensuring operational safety and minimizing unexpected downtimes.

PdM stands as one of the most potent and widely adopted maintenance strategies. Operating on a condition-based paradigm, it relies on assessing the operational status of production machinery. In industrial settings, PdM has been the preferred maintenance approach in 89% of cases, with other time-based maintenance policies accounting for only 11% of patients [26]. The emergence of Industry 4.0 has ushered in a wave of advanced technologies, including sophisticated sensors, computing advancements, the Internet of Things, and data-driven modeling. These technologies have facilitated the identification of equipment degradation and impending failures, resulting in a decreased necessity for routine maintenance procedures such as periodic and preventive maintenance [29]. In crafting a PdM model for a multi-component production system, carefully considering established condition and degradation thresholds for each component is paramount [27]. Figure 2.2 illustrates the overarching maintenance strategies and objectives [28].

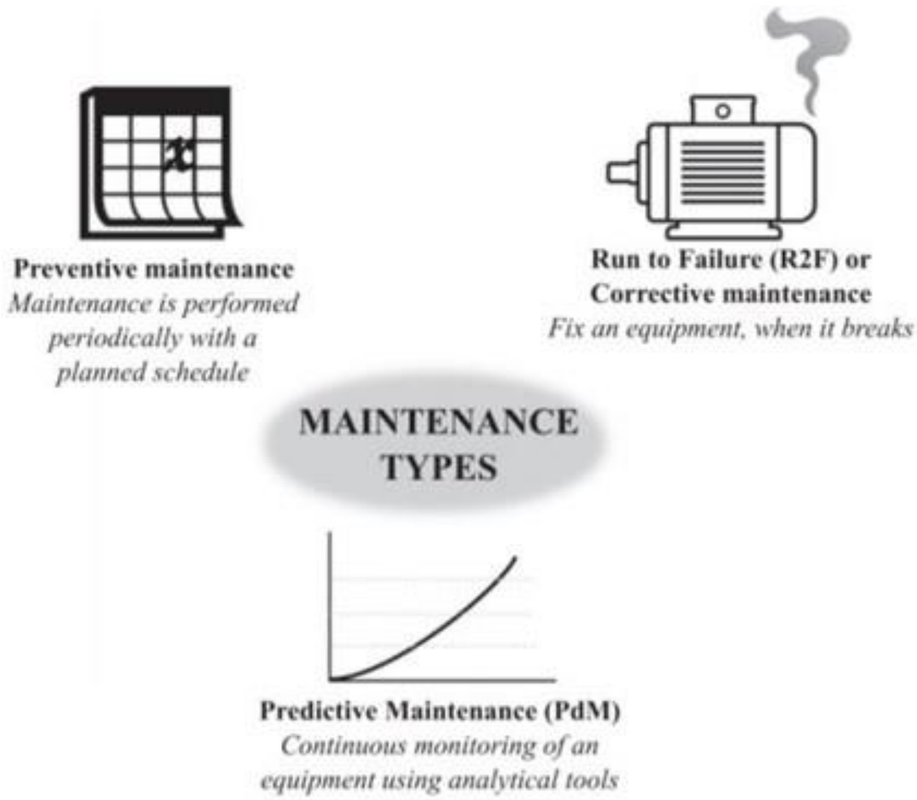

Figure 2.2. Illustrates maintenance strategies and their functions [28].

Each maintenance strategy serves a crucial role in the context of manufacturing operations. The maintenance process is of paramount significance to industries, and a well-crafted strategy should yield notable benefits, encompassing improved equipment operating conditions, reduced failures, minimized maintenance costs, and prolonged equipment life [29].

2.2. DATA-DRIVEN APPROACHES IN PREDICTIVE MAINTENANCE

The data-driven approach harnesses data collected from Industrial Internet of Things sources to analyze the damage characteristics of industrial equipment and formulate models for predicting future trends [30]. This method entails the daily analysis of condition monitoring data acquired from device metrics. It employs machine learning, deep learning, and pattern analysis techniques to predict potential malfunctions based on the collected data [31]. Data-driven approaches encompass neural networks, Random Forest (RF), Support Vector Machine (SVM), Relevance Vector Machine (RVM), Bayesian methods, fuzzy logic, regression analysis, and Genetic Algorithms (GA). The defining characteristics of data-driven technology lie in its rapidity and ease of implementation, coupled with the ability to discern previously unnoticed relationships [32].

Nonetheless, data-driven approaches necessitate substantial datasets and a balanced approach to mitigate the risk of overfitting and overgeneralization. Various machine learning and deep learning methods have been employed, relying on data to monitor machine states and predict failures. Notably, the application of deep learning techniques, recognized for their proficiency in handling high-dimensional, non-linear, and diverse data without manual intervention, has demonstrated considerable success in predictive maintenance. Researchers have proposed models based on RNN to address prognostic challenges. However, it is essential to note that RNNs face challenges such as vanishing gradient and exploding gradient issues when processing long sequences, limiting their capacity to retain prior information [33]. Figure 2.3 illustrates a detailed description of the data-driven predictive maintenance implementation process [34].

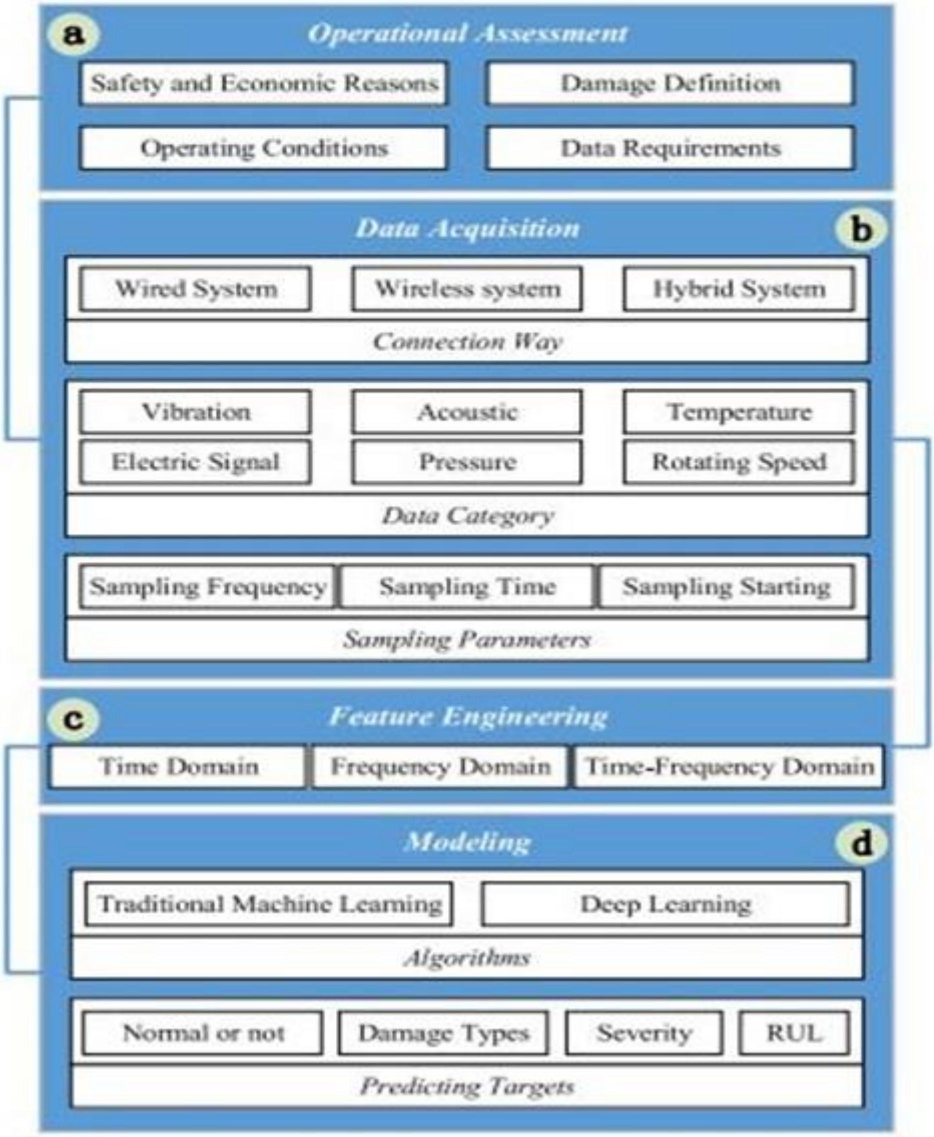

Figure 2.3. Depicts the implementation process of data-driven predictive maintenance [34].

To surmount these challenges, an enhanced version of RNN, known as LSTM, has been applied to formulate diverse predictive models for anticipating future failures [35]. LSTM has proven effective in predicting the failure location of aircraft engines. Moreover, an innovative Bidirectional-LSTM (BLSTM) model has been introduced, demonstrating the capability to simultaneously capture long-term information in both forward and backward contexts of input sequences. This BLSTM model has been employed for predicting system performance degradation. It is noteworthy that recurrent networks can impose higher computational demands. Additionally, CNNs, initially designed for image processing, have found application in predicting failures in time series data by extracting pertinent features and patterns. CNN architectures, with weight-sharing filters, have led to significant enhancements in prediction accuracy. Notably, in the literature, hybrid deep neural network models have been devised to leverage the strengths of different algorithms concurrently, enabling the extraction of both temporal and spatial features for precise failure prediction [36].

2.3. DEEP LEARNING IN PREDICTIVE MAINTENANCE

Deep learning, a subfield of machine learning, is characterized by its ability to automatically unearth concealed patterns within data by utilizing multiple layers of non-linear processing [37]. The literature has undertaken a comparative analysis between deep learning and traditional machine learning concerning the modeling process. This scrutiny reveals that data preprocessing is crucial for deep and traditional machine learning. The fundamental distinction lies in the feature processing approach. Traditional machine learning methods require significant effort in feature extraction and selection, whereas deep learning excels indirectly and automatically learning these hidden patterns. This characteristic renders deep learning methods more efficient and effective for the modeling process. However, a notable drawback is apparent—deep learning is often perceived as a 'black box,' wherein the abstracted features are not readily understandable or interpretable [38].

Deep learning techniques have been effectively employed in analyzing data obtained through continuous industrial equipment monitoring using smart electrical sensors. These methods aim to predict the equipment's health status by analyzing the collected data and extracting relevant features and patterns. In response to the escalating demand for minimizing downtime and mitigating economic losses associated with equipment failures, researchers have been dedicated to developing models for equipment condition prediction and proactive maintenance before failures occur. Deep learning networks, in particular, have demonstrated heightened accuracy when applied to systems such as aircraft maintenance systems [39],[40], delta robots [41], wind turbines [42], building management systems [43], air compressor systems[44], and wind generators [45]. Autoencoders and LSTMs are particularly effective when handling streaming data from manufacturing processes. LSTMs and recurrent neural networks are well-suited for detecting errors stemming from sensors in the automotive industry [46],[47]. In Ref [48], the A2-LSTM model was employed, utilizing a series of electrical records as input. Features were extracted from the data and incorporated into a feature attention network, where each part was automatically adjusted based on its significance. The model introduced time dependence into the manufacturing system by integrating these re-weighted features into the health prediction component. The proposed model demonstrated its efficacy in guiding equipment maintenance efforts, and the A2-LSTM model exhibited promising results compared to real-world cases.

Research has demonstrated the effectiveness of multi-layer network architectures across diverse data types [49]. For instance, Convolutional Neural Networks excel at processing image data and prove valuable in identifying anomalies in structural analyses of roads or railways [50]. Multilayer neural networks, encompassing Elman Neural Network (ENN), Back Propagation Neural Network (BPNN), Probabilistic Neural Network (PNN), Fuzzy Neural Network (FNN), and Wavelet Neural Network (WNN), have demonstrated successful diagnoses of mechanical faults in industrial equipment.

In Ref [51], A Predictive Maintenance model was introduced that harnessed the IoT in tandem with deep neural networks, specifically Long Short-Term Memory and Recurrent Neural Networks. This model was specifically designed for predicting light bulb failures. The study's results illustrated the effectiveness of their hybrid model, revealing a minimal error rate of 0.79%. This development holds relevance in maintenance planning within cyber-physical production systems; At the same time, Ref [52], LSTM autoencoders were employed to facilitate automatic decoding to classify the real-world condition of machines based on sensor data. The model exhibited an impressive average accuracy rate of 94.2%, evaluated using data collected from a steelmaking production process. Furthermore, Ref [53] introduced a hybrid model that integrated GUR-ELM (Extreme Learning Machine) for predicting bearing failures, with a specific emphasis on vibration signals using locomotive bearing datasets. The model achieved an impressive accuracy rate of 94%. To gauge the performance of their proposed model, it was benchmarked against CNN, Denoising Autoencoders (DAE), and Deep Belief Networks (DBN). Ref [54], A hybrid model was devised to predict bearing failures based on vibration signals, amalgamating Adversarial Conditional Generative Adversarial Networks (ACGAN) with CNN. The outcomes revealed a notable accuracy rate of 98%.

2.4. CONVOLUTION NEURAL NETWORKS AND LONG SHORT–TERM MEMORY

In computer vision, CNNs are foundational techniques for classification and regression. Specifically designed to process network-like structured data, such as images, CNNs share similarities with feed-forward neural networks. They utilize neurons to adjust weights through a learning process guided by a loss function. CNNs have found extensive application in pattern recognition tasks, primarily about image data, although not exclusively. The history of CNNs dates to the 1990s, when they were initially explored for tasks like speech recognition and document reading. However, it was the introduction of ImageNet and its deep CNN architecture in 2012 that propelled CNNs to the forefront of computer vision. They showcased their effectiveness by adeptly processing extensive image datasets containing over a thousand categories [55]. Recently, CNN variants, such as ResNet-50 [56] and VGG 16 [57], have demonstrated impressive performance and widespread adoption. The significant ascent of CNNs can be attributed to the substantial computational demands of ANNs. Color images, typically represented by an MxNx3 matrix, necessitate ANNs to possess MxNx3 weights, each of which must be updated in every learning iteration. This process becomes impractical due to computational constraints and extended training times. Additionally, ANNs often grapple with overfitting due to their fully connected nature. To process network-like structured data in Fully Connected Neural Networks (FCNN), data must be flattened into a one-dimensional format, resulting in the loss of specific data patterns. In contrast, CNNs employ convolution operations to process 2D data, providing advantages such as sparse interaction, parameter sharing, and consistent representation. Sparse interaction implies lower connectivity in the convolutional layer than fully connected layers of the same neuron size. Parameter sharing allows a kernel in a convolutional layer to process data at various input positions, differing from fully connected layers that process only one input position. Consistent representation ensures that any transformation applied to the input of the convolutional layer results in a corresponding transformation in the output [58]. Pooling is another crucial component of CNNs, involving partitioning the 2D matrix into different grids during the aggregation process. Aggregation functions facilitate the downsampling of network data through summary statistics. Standard pooling functions include max pooling, which selects the maximum values in the network, and mean pooling, which computes the average values. Convolution and pooling operations work in tandem to abstract information within the data. Typically, CNNs consist of multiple convolutional and pooling layers. The large 2D array is divided into smaller 2D arrays following data abstraction and downsampling. These matrices are then normalized and fed into a fully connected layer for further processing [58].

The LSTM network is another type of deep learning architecture structured based on the principles of recurrent neural networks. LSTMs are particularly well-suited for handling time series applications due to their utilization of feedback connections. One of their key advantages is their ability to address the 'vanishing gradient' problem, enabling the gradients to flow consistently. The vanishing gradient problem is a common issue in computational solutions and arises when the eigenvalue spectrum of a matrix is less than 1. In an LSTM network, the fundamental unit is called a 'cell,' which comprises an input gate 'i,' an output gate 'o,' and a forget gate 'f.' The number of cells within each LSTM network corresponds to the number of hidden layers. The concept of Long Short-Term Memory was initially introduced by Hochreiter and Schmidhuber in 1997 [60]. They identified a significant challenge with recurrent neural networks, particularly the computational cost of backpropagation when attempting to store information over extended periods. To address this, they eliminated unnecessary scaling and introduced a constant error flow through the 'Constant Error Carrousel' (CEC). CEC maintains a constant cell state with a weight equal to 1. LSTM was initially tested with combined Rebbel rules, sequence data, and electrical signals. This approach proved capable of solving problems that were hitherto deemed unsolvable. The computational complexity of each time step and weight in LSTM is O (1).

Deep learning techniques, exemplified by the LSTM, have demonstrated their efficacy in various time series classification applications characterized by long-term dependencies facilitated by memory mechanisms. Conversely, CNN finds its primary utility in image classification tasks. CNN's notable feature lies in its multi-layer stacked architecture, which efficiently extracts and represents input data features, thus enhancing feature extraction capabilities. As a result, CNN has the advantage of capturing and extracting data features more effectively. In predictive applications for proactive maintenance, there is a growing interest in combining various deep learning methods in hybrid forms to harness the strengths of each approach. These mixed deep learning methods leverage historical sensor data and machine health information to perform predictive maintenance on production equipment.

Refs [61], [62], The authors proposed a hybrid approach combining CNN-LSTM for predictive maintenance. This CNN-LSTM model not only enhances accuracy but also reduces complexity. Evaluation against regular LSTM and Gradient Boosted Decision Tree (GBDT) methods, using a predictive maintenance dataset from Microsoft's GitHub repository, revealed the superiority of the CNN-LSTM hybrid approach, increasing the average F-Score from 93.34% to 97.48%. Additionally, they introduced the PPO-LSTM model, merging Deep Reinforcement Learning and Long-Term Memory, which demonstrated optimal decision-making in a stochastic environment. Experimental results showcased its superior performance, surpassing other DRL approaches by 53% in resource management and 65% against human participants, confirming its efficiency, adaptability, and convergence in simulation. At the same time, Ref [63] proposed an early detection method for rolling bearing faults using a multi-scale CNN and Gated Recurrent Unit network with an attention mechanism (MCNN-AGRU). This model was trained using average data. Another algorithm for multiple fault diagnosis in rotating machinery, referred to as the local-wise response CNN-based Naïve Bayes (WCNN-NB) algorithm, was introduced by [64]. The results indicated classification accuracies of 99.68%, 92.5%, and 97.5% for three datasets, with acceptable misclassification rates under the examined operational conditions.

Ref [65], proposed an HPC model for fault diagnosis that combines Hierarchical Symbolic Analysis (HSA) and Particle Swarm Optimization (PSO) with a Convolutional Neural Network (PSO-CNN). This model, evaluated on two distinct datasets, achieved a maximum classification accuracy of 98.97% and 99.09%. To identify tolerance errors, [66] introduced a model that employs CNN and gcForest. This model converts raw bearing vibration signals into time-frequency images using Continuous Wavelet Transform (CWT). The results demonstrated a high classification accuracy for bearing faults, with a fault detection rate exceeding 98% across datasets of varying sizes. Kumar [67] employed an adaptive gradient optimizer in conjunction with a deep CNN to detect bearing and rotor faults in Squirrel Cage Induction Motors (SCIM), achieving an average accuracy of 99.70%. Another Ref [68], introduced a model based on Motor Current Signature Analysis (MCSA) and a novel 2D CNN to eliminate the need for manual feature extraction. Ref [69] utilized CNNs to identify stator short turns and broken rotor bars via the axial flux signal, and deep neural networks were employed to detect stator and rotor faults. Authors in Ref [70] proposed an approach based on a CNN model with a small kernel size, adaptive gradient optimizer, and batch normalization. This CNN model, featuring a larger number of computational layers, achieved commendable classification accuracy across different health states of SCIM, exceeding 99.50% on the Case Western Reserve University (CWRU) dataset.

Ref [71] adopted a traditional feature engineering rendering approach called Dilator-CNN (D-CNN) for fault diagnosis. This approach eliminates the need for raw vibration signals. CNN was employed to create a model for diagnosing bearing faults in embedded devices using acoustic emission signals, as undertaken by authors in Ref [72]. The model exhibited a classification accuracy of up to 99.58% while maintaining lower computational costs compared to other DL-based methods. A Predictive Maintenance model utilizing CNN (PdM-CNN) has been proposed for classifying faults in rotating equipment and guiding maintenance timing, as presented by Ref [73]. Data were collected from a vibration sensor mounted on the motor drive end bearing, with classification accuracy reaching 99.58% and 97.3% when the method was applied to publicly available MaFaulDa and CWRU databases, respectively. Ref [74], a dual-path RNN-based method was presented, incorporating a wide primary kernel and a deep path CNN (RNN-WDCNN) for diagnosing rolling element bearing faults in electromechanical systems, using raw temporal signals such as vibration data. This method excels in swiftly classifying input sequences compared to traditional fast transform (FFT)--based approaches. While Ref [75] introduced a technique utilizing LSTM networks and the engine's no-load testing acoustic signal.

To evaluate the effectiveness of the suggested multi-domain features across diverse DL architectures, Ref [24] conducted sensitivity analyses on input channels, revealing that Convolutional LSTM (CLSTM) exhibited superior performance. The method's effectiveness was compared with twelve other algorithms, and the findings indicated that the proposed model achieved 100% accuracy with shorter inputs compared to other models. CNN and LSTM were employed to capture real-time current and vibration signals, facilitating a combined representation and temporal coding of raw data streams. In Ref [76], the Attention Dense Convolutional Neural Network (ADCNN) was introduced, integrating dense convolutional blocks with an attention mechanism. Simulation results showcased that the proposed method required fewer unknown learning parameters and achieved an accuracy of 99.51%. Subsequent enhancements to this model by Ref [77] further improved accuracies to 99.57% and 99.6%, respectively.

2.5. SUMMARY OF RELATED WORK

In the dynamic landscape of Prognostics and Health Management (PHM), deep learning emerges as a beacon of innovation with diverse applications in fault diagnosis and failure prediction, especially within industrial equipment and systems. With a remarkable track record in domains like computer vision and medical image analysis, deep learning techniques have proven their mettle in decoding complex condition monitoring signals, such as vibrations, acoustic emissions, and pressure. Their unparalleled ability to unravel intricate representations from raw data forms the crux of their effectiveness. This chapter unfolds a comprehensive exploration of the foundational elements for grasping deep learning, encompassing architectures tailored explicitly for this purpose. Moreover, critical challenges and data-driven intricacies are carefully addressed. In the context of Industry 4.0, where Predictive Maintenance holds paramount importance, sensor-generated data undergoes meticulous processing, facilitating informed decision-making. The recent years have witnessed the ascendancy of machine and deep learning techniques in real-time error monitoring and detection, proving highly efficient. Undeniably, deep learning has entrenched itself in the realm of industrial big data analytics, empowering decision-makers in proactive equipment maintenance, thereby averting breakdowns, minimizing losses, and elevating overall system reliability.

PART 3

RESEARCH METHODOLOGY

3.1. ARTIFICIAL INTELLIGENCE

The interdisciplinary domain of AI is positioned at the intersection of computer science, mathematics, engineering, and cognitive science. At its core, the overarching aim is the generation of intelligent machines or programs equipped with the capacity to simulate cognitive functions reminiscent of human cognition. Tasks traditionally reliant on human intelligence, such as natural language comprehension, pattern recognition, decision-making, and experiential learning, are systematically addressed and executed by engineered AI systems. Within this expansive scope, AI has manifested as a formidable tool facilitating the advancement of intelligence predictive algorithms across a myriad of applications [78]. The adeptness of artificial intelligence approaches is displayed in their proficiency in managing multidimensional and multivariable data and discerning concealed relationships within intricate and dynamic environments [79]. Inextricably entwined with big data, AI demonstrates exceptional suitability for addressing vital inquiries, rectifying deficiencies, and elucidating key procedural issues, particularly in substantial data sets' analysis and processing stages. AI is pivotal in the transition to Industry 4.0, providing a potent technological alternative to alleviate the inherent limitations and inefficiencies associated with conventional industrial techniques and practices. As a subfield of artificial intelligence, ML has evolved from its origins in pattern recognition to include the analysis of data structures and their integration into models that can be comprehended and reconstructed by end-users [80].

Furthermore, machine learning is delineated into four primary subtypes: supervised learning, unsupervised learning, reinforcement learning, and deep learning, as Illustrated in Figure 3.1, encapsulating all AI categories. Supervised and unsupervised learning find application in scenarios where the objective involves predicting or discerning the presence of labels in datasets. The presence of a dependent variable characterizes supervised learning, while unsupervised learning lacks such explicit labels. In contrast, reinforcement learning constitutes a computational paradigm wherein learning unfolds through interactions with the environment, emphasizing the facilitation of systems to execute actions within their surroundings to maximize cumulative rewards intelligently. The ultimate objective is to endow systems with the capability to adeptly perform tasks and make informed decisions grounded in data [81].

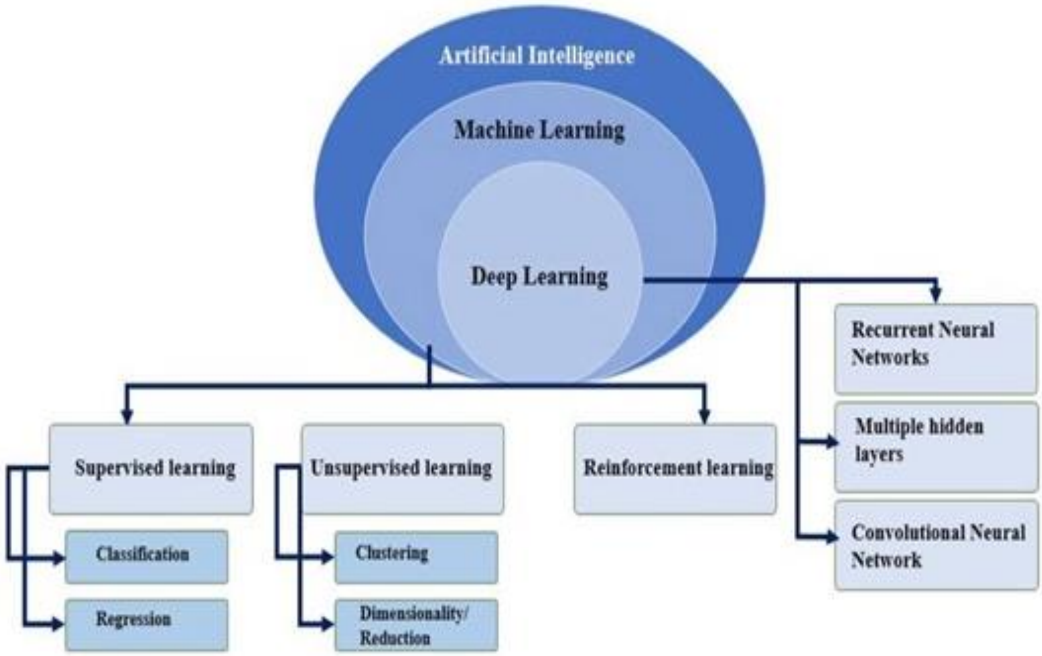

Figure 3.1. Artificial intelligence state of the art [81].

In the context of Industry 4.0, which strives to fulfill the requisites of intelligent technology-driven manufacturing systems, diverse maintenance methodologies have been devised, encompassing reactive, preventive, and predictive maintenance approaches [82]. Reactive maintenance is enacted in response to equipment failure, whereas preventive and predictive maintenance are implemented during routine machine operation. While essential, preventative maintenance may entail costs and disrupt machine uptime, given its requirement for scheduled downtime [83]. It lacks continuous machine condition monitoring and fails to construct a comprehensive machine health profile. The integration of AI-based methodologies has significantly enhanced the reliability and efficacy of PdM, incorporating machine and deep learning techniques. Machine learning approaches have been widely applied in machinery for failure prediction. More recently, various deep learning methodologies have surfaced, propelled by advancements in the field, to implement predictive maintenance for industrial equipment and augment the reliability of PdM [84]. Deep learning leverages extended chains of neural network layers [85], each nonlinearly transforming the network to achieve progressively abstract and higher-level representations, drawing inspiration from computational models of intricate real-world systems. Moreover, data-driven manufacturing, facilitated by Industry 4.0 technologies such as CPS, the IoT, and Big Data Analytics, catalyzes advancements in predictive maintenance [86]. This evolution enables continuous machine monitoring and utilizes health status data analysis for early fault detection [87]. AI is pivotal in supporting Predictive Maintenance (PdM) by scrutinizing extensive sensory data to identify patterns and anomalies, thereby preempting production disruptions [88]. In conclusion, machine and deep learning have been pivotal in Machine Health Management (MHM) [89], and artificial intelligence methodologies hold substantial significance in the context of Industry 4.0. These methodologies significantly contribute to formulating maintenance strategies, particularly in predictive maintenance for industrial equipment and machinery. The overarching objective is to implement proactive maintenance measures and forestall material and human disasters.

3.2. THESIS MODEL

3.2.1. Convolutional Neural Network

The acronym "CNN" denotes Convolutional Neural Network, a category of deep neural networks distinguished by its terminology derived from the convolutional process inherent in linear mathematics. LeCun, the progenitor, introduced this paradigm with the pioneering LeNet-5 architecture in the formative years of the 1980s. The genesis of CNN finds its roots in LeCun's seminal contributions in 1989, manifesting as a structured framework designed to adeptly analyze data through a network-centric topology, with a pronounced focus on image and time series data [90].

Conventional CNNs have garnered widespread utility in the realm of image data processing. Termed 2D CNNs due to their distinctive operation in two dimensions within the data space, these networks stand in contrast to their one-dimensional counterparts, aptly named 1D CNNs. The kernel, integral to the convolutional process, traverses the data in two dimensions in the case of 2D CNNs, emphasizing their proficiency in handling intricate spatial information inherent in images. Conversely, 1D CNNs engage in convolution along a singular dimension, rendering them particularly adept at processing time series data, where the sequential nature of information is paramount [91]. The fundamental disparity distinguishing 1D from 2D CNNs lies in the representation of kernels and feature maps. While 1D CNNs employ one-dimensional matrices for these components, 2D CNNs utilize two-dimensional matrices. The computational complexity associated with 1D and 2D convolutions exhibits a noteworthy contrast. When an image of dimensions $M \times M$ undergoes convolution with a kernel of size $T \times T$, the computational cost for a 2D CNN is denoted as $O(M^2T^2)$. In contrast, a 1D convolution applied to data with analogous dimensions (M and T) incurs a computational cost of $O(MT)$. This discrepancy underscores the substantially reduced computational burden imposed by 1D CNNs relative to their 2D counterparts under comparable circumstances, rendering them a preferred choice for researchers across diverse domains, particularly in signal processing, owing to their commendable performance. Furthermore, the versatility of one-dimensional CNNs extends to their capacity to adeptly handle multimodal data, encompassing images, audio, and video domains.

The hierarchical configuration inherent in deep CNNs is distinguished by its aptitude for learning intricate representations across diverse levels of abstraction. The pivotal divergence between CNNs and shallower architectures lies in the strategic employment of parameter sharing, allowing the network to discern specific features across varied spatial locations. A prototypical CNN architecture typically manifests as amalgamating three foundational neural layers or building blocks, wherein convolutional layers interleave with pooling layers, culminating in fully connected layers. The augmentation of these architectures is accomplished through the incorporation of assorted organizational modules, such as batch normalization and dropout, thereby fortifying the overall performance of CNNs. The strategic arrangement of CNN layers assumes paramount significance, exerting a tangible influence on generating novel architectures and consequent enhancements in performance. In the ensuing sections, a concise overview and discourse on the functionality and role of each layer will be presented [92].

3.2.1.1. Architecture

A conventional CNN architecture comprises a sequential arrangement of convolutional and pooling layers, succeeded by fully connected layers. To gain a deeper understanding of this architecture, Figure 3.2 serves as a visual representation, elucidating the inherent structure of the CNN:

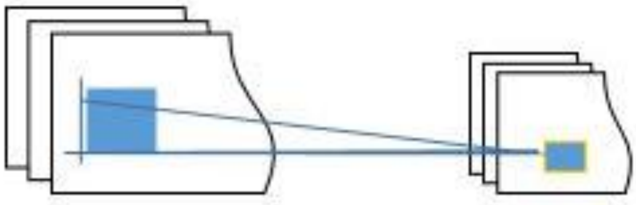

Figure 3.2. Convolutional layer structure.

- The Convolutional Layer employs trainable filters, commonly referred to as kernels, to process input data through a continuous or windowed sliding technique. The resultant output consists of a set of feature maps, each delineating the activations generated by corresponding filters. This mechanism enables the network to discern and highlight distinctive patterns and features within the input data [93]. Every filter within a convolutional layer provides a distinct perspective on the input data, thereby rendering the selection and quantity of filters pivotal determinants for network performance. Each convolutional layer plays a significant role in the feature extraction process, thereby introducing a successive level of abstraction to the input data representation. The cumulative effect of these layers is to progressively distill and emphasize salient features essential for the network's overall understanding and subsequent decision-making processes [89]. In the standard configuration, a convolutional layer operates linearly, and an activation function is subsequently applied to yield a non-linear output. The fundamental essence of the CNN architecture resides in the convolution operation, applicable to one-dimensional or two-dimensional data. This process is executed by deploying sliding convolutional filters that traverse vertically and horizontally, capturing crucial input data features [94]. Figure 3.3 visually illustrates the structural intricacies of the convolutional layer within the CNN architecture,

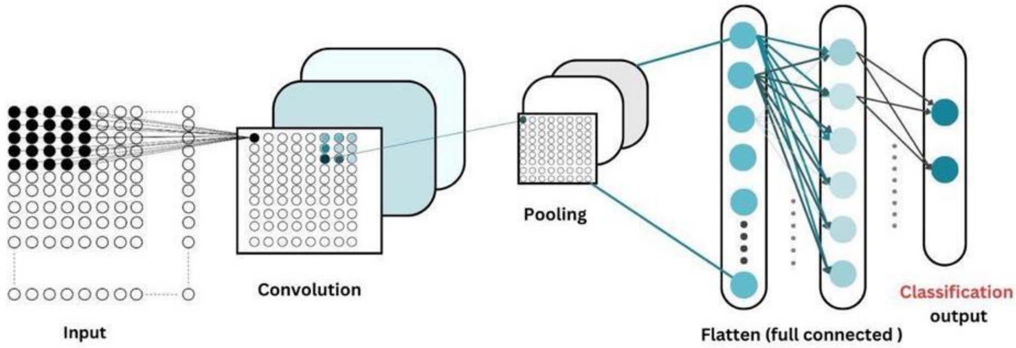

Figure 3.3. CNN structure.

Convolution within a neural network encompasses three notable properties. Firstly, weight sharing is employed, entailing the utilization of the same weights across multiple locations. This practice effectively diminishes the overall number of parameters, contributing to a more efficient and streamlined model. Secondly, the concept of sparse interactions or connectivity comes into play. By incorporating sparsity in weights, the network adeptly learns associations between adjacent pixels, thereby reducing the demand for parameter storage and computational resources. Lastly, convolutional operations exhibit the property of invariant representations, wherein the output remains steadfast and consistent despite variations in the input. These properties collectively enhance the network's capacity for effective feature extraction and contribute to the overall efficiency of the convolutional layer. Figure 3.4 presents the pseudocode of CNN,

Algorithm 1: CNN model

Input: x input features vector, F filter with size k x d

Output: c' output features vector

For i = 1 to N

    w_i = [x_i, x_i+1, …, x_i+k-1]

    c_i = ReLU(w_i ⊙ F)

End

c' = MaxPool(c)

Figure 3.4. CNN pseudocode.

- The Pooling Layer, strategically situated following the convolutional layer, consolidates the feature maps generated by the preceding convolutional layer into singular values. Its function involves discerning the most pivotal features extracted by each filter, irrespective of their relevance in other filters. This crucial process mitigates the feature maps' dimensionality and concurrently reduces the number of parameters, thereby fortifying the model's resilience against noise. Analogous to convolutional layers, pooling layers actively contribute to cultivating displacement-invariant features by incorporating neighboring pixels in their computations. Their operational essence is summarizing information within the receptive field, ultimately outputting the predominant response within that local region. This mechanism enhances the network's proficiency in capturing essential features while promoting computational efficiency [95].
- Fully Connected Layers (FC), a customary inclusion in most CNN architectures, bear semblance to traditional Multilayer Perceptrons (MLP) layers. FC layers 1Qfunction as a traditional neural network that is sequentially positioned after the convolutional and pooling layers. Their primary objective lies in converting multi-dimensional features into a singular one-dimensional (1D) feature vector, subsequently utilized by a classifier or predictor. This strategic transformation facilitates the extraction of comprehensive high-level representations from the preceding layers. Conclusively, a softmax layer is frequently integrated as the probabilistic classifier, adding a final layer of refinement to the CNN

architecture. This collective arrangement effectively utilizes learned features for accurate predictions and classifications.

- Dropout: is incorporated into the network to impart regularization, a mechanism essential for preventing overfitting and improving overall model performance. A predetermined dropout probability is applied in this process, leading to the random deactivation of specific neurons or connections during the training phase. By introducing this stochastic element, dropout effectively diversifies the network's learning process, reducing reliance on particular pathways and features. This regularization technique fosters a more robust and generalizable model by preventing overemphasizing individual neurons or connections, thus enhancing the network's capacity to generalize well to unseen data [91].

The CNN architecture undergoes fine-tuning to minimize error margins and optimize performance by applying a backpropagation algorithm. This iterative approach systematically adjusts the learning weights of the network, facilitating continuous refinement and adaptation of the model throughout the training process. As a result, CNN has proven to be a formidable tool for automated pattern recognition, adept at extracting global features from images. This proficiency obviates the necessity for manual feature engineering, underscoring the network's capacity to autonomously discern and leverage relevant patterns in the data, thus contributing to its efficacy in diverse image-processing applications [96]. CNNs have showcased their effectiveness across various domains, extending their utility to analyzing and classifying time series data. Their notable impact is especially pronounced within the purview of Industry 4.0, where they excel in proactively predicting failures in industrial equipment and machinery. Leveraging their adept feature extraction capabilities, CNNs facilitate the timely identification of potential issues, enabling proactive maintenance interventions. This forward-looking strategy unequivocally contributes to cost savings in maintenance operations and concurrently minimizes downtime, thereby solidifying CNN's pivotal role in enhancing operational efficiency within industrial contexts [97].

3.2.2. Long Short-Term Memory

LSTM, an abbreviation for Long Short-Term Memory, is a member of the deep learning models associated with the RNN family. Tailored for analyzing and processing sequential data, LSTM is specifically designed to excel in tasks involving time series data, with a primary emphasis on predictive modeling. Its notable suitability for time series applications stems from incorporating feedback mechanisms, enabling it to retain and recall information over prolonged periods effectively. This aligns seamlessly with the inherent sequential nature of sensor data, making LSTM a particularly potent and well-adapted model for tasks demanding temporal dependencies and nuanced contextual understanding [98]. LSTM manifests as a recurrent neural network algorithm recognized for its aptitude in assimilating extensive temporal dependencies. In contrast to conventional recursive algorithms, including traditional neural networks, wherein the persistence of prolonged information and the resolution of dependency concerns across temporal epochs prove challenging, LSTM introduces a pioneering solution. Demonstrating proficiency in mitigating the predicament of vanishing gradients, it orchestrates an adept intervention to sustain a seamless progression of gradient flows. The ubiquitous challenge of the vanishing gradient problem arises in mathematical solutions, mainly when the singular value spectrum of the matrix falls below unity. As the complexity of the learning network escalates throughout the training regimen and the inverse weight values of the network approach diminutive or proximate-to-zero magnitudes, the viability of updates diminishes, potentially leading to the cessation of the training process.

LSTM strategically confronts the vanishing gradient obstacle by integrating a memory cell into its architectural framework. This inclusion alleviates the complexities associated with long-range dependencies, fostering the seamless amalgamation of antecedent information with contemporaneously acquired data. Data The inception of LSTM dates back to the seminal work of Hochreiter and Schmidhuber in 1997 [99]. Their discerning perspective was rooted in recognizing that backpropagation in recurrent neural networks, intended to retain information across prolonged intervals, incurred substantial computational costs, predominantly stemming from the inadequate backpropagation of error. In response to this challenge, they introduced the innovative concept of maintaining a constant error flow through Constant Error Carousels (CEC). The CEC encapsulates an unvarying cell state featuring a stable weight set at 1. LSTM's inaugural assessments encompassed succinct Rebbel rules, sequential data, and electrical signals. Significantly, it showcased an unprecedented capability to surmount previously intractable problems. Notably, LSTM emerged as an exemplary learning algorithm, proficient in prediction tasks and excelling in scenarios characterized by extensive datasets exhibiting sequential dependencies.

An LSTM network comprises units denominated as cells, each featuring four intricately interconnected gates, meticulously crafted to address the challenges inherent in managing long-range dependencies and seamlessly integrating prior information with novel data. These four gates, namely the input gate, output gate, forget gate, and memory unit, collectively facilitate the retention and transmission of information from preceding time steps to subsequent ones. The values encapsulated within these units persistently endure, remaining impervious to external influences, thereby ensuring their enduring relevance.

The quantity of cells within each LSTM aligns with the count of hidden layers, and the computational complexity at each time step is consistently maintained at O (1). With its resilient architectural design, LSTM substantiates heightened accuracy and superior decision-making capabilities, empowering the predictive modeling of future values predicated upon historical time series data. Remarkably, LSTM distinguishes itself through its remarkable efficacy. Within the illustrated LSTM module structure, the symbol X(t) corresponds to the current input value, h(t-1) signifies the preceding hidden state, and c(t-1) represents the antecedent memory state. The output denoted as h(t) gives rise to the present latent state, while c(t) mirrors the contemporary memory state [100].

3.2.2.1. Architecture

An LSTM network comprises several vital components collaborating to process time series data and maintain information over time. The main components of the LSTM network are as follows [101]:

- Cell Unit: At the core of the LSTM network, the cell unit serves as the foundational entity responsible for storing and processing information across temporal intervals. Comprising indispensable components such as inner and outer gates and a memory module, the cell unit plays a pivotal role in orchestrating the intricate mechanisms that underlie the network's ability to capture, retain, and manipulate information over time.
- Gates: Integral to the functioning of LSTM, gates meticulously govern the ingress and egress of information within the cell unit. The three principal gates in LSTM are as follows:

    1. Input Gate: This gate discerns the degree to which new information is permitted to enter the cell, playing a crucial role in regulating the assimilation of fresh data into the network.
    2. Output Gate: Responsible for modulating the extent to which information residing within the cell can emanate and impact the ultimate output, the output gate is a critical determinant of the network's external influence.
    3. Forget Gate: Functioning as a pivotal component, the forget gate determines the extent to which prior information within the cell is subject to abandonment or removal, crucial for managing the retention and discard of historical data within the LSTM framework.

- Memory Unit: As a crucial repository, the memory unit adeptly stores information across temporal epochs. Its functions encompass not only the storage of data but also its continual updating, thereby facilitating the perpetuation of the network's internal state.
- Memory Gate: Within the LSTM architecture, the memory gate assumes a pivotal role in facilitating updates to the memory, dynamically incorporating both new and past information. This gate operates as a critical mechanism for regulating data flow into the memory unit, ensuring that the network can adaptively integrate relevant details while preserving the continuity of its internal representation.

Figure 3.5 intricately elucidates the architectural framework of the LSTM, offering a visual representation of its integral components and their interplay in effecting temporal information retention and processing.

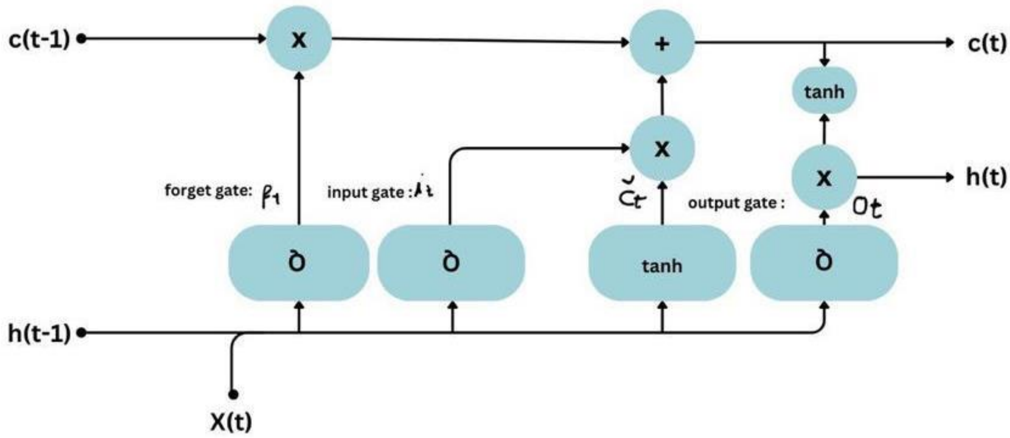

Figure 3.5. LSTM structure.

The inherent capability of the LSTM network to adeptly navigate long-term dependency challenges and regulate information flow substantiates its considerable value, particularly in the analysis of sequential data, exemplified by time series datasets. Its proficiency resides in its ability to process sequential data methodically, thereby proving instrumental in tasks such as failure prediction, where the nuanced understanding and retention of temporal dependencies are paramount for accurate and reliable predictions [102]. Figure 3.6 depicts the pseudocode delineating the operational logic of the LSTM algorithm.

```
Algorithm 2: LSTM model
# Input: features vector c'
# Output: h vector
For j = 1 to t
    i_j = σ(W_i * [h_t-1 + b_i])
    f_j = σ(W_f * [h_t-1, x_t] + b_f)
    q_j = tanh(W_q * [h_t-1, x_t] + b_q)
    o_j = σ(W_o * [h_t-1, x_t] + b_o)
    c_j = f_j ⊙ c_j-1 + i_j ⊙ q_j
    h_j = tanh(c_j)
End
```

Figure 3.6. LSTM pseudocode.

3.2.3. Hybrid CNN-LSTM Model

Colloquially called fusion-based technology, hybrid-based technology represents an amalgamation of model-based and data-driven technologies. This innovative paradigm harnesses the power of data to acquire insights into model parameters while concurrently integrating knowledge of underlying physical processes. By doing so, it discerns the optimal type of regression analysis—linear, polynomial, exponential, etc. This approach finds application in predicting machine failures and executing proactive equipment maintenance. The incorporation of data-driven methods in this hybrid framework facilitates the seamless integration of multiple techniques, culminating in precise and reliable forecasting results [103]. The hybrid strategy strategically capitalizes on the inherent strengths of model-based and data-driven approaches, seamlessly integrating physics principles when data is unavailable. This adaptability renders the hybrid method incredibly potent in scenarios characterized by incomplete system knowledge, effectively addressing the limitations of individual methodologies. Through this symbiotic fusion, the hybrid technique emerges as a robust solution capable of providing accurate insights and predictions even in circumstances where one approach alone may fall short.

The suggested hybrid model capitalizes on the collective capabilities of CNN and LSTM deep learning algorithms, seamlessly integrating their distinct strengths. CNN is esteemed for its adeptness in feature extraction, while LSTM excels in handling time series data. This algorithm has been meticulously employed to craft a specialized hybrid model designed for PdM, with a principal emphasis on predicting failures. Notably, the model leverages the MetroPT dataset as a foundation for training and validation [104].

The architectural configuration of the hybrid CNN-LSTM model is elucidated in Figure 3.7 (A), and Figure 3.7 (B) detailing the model structure.

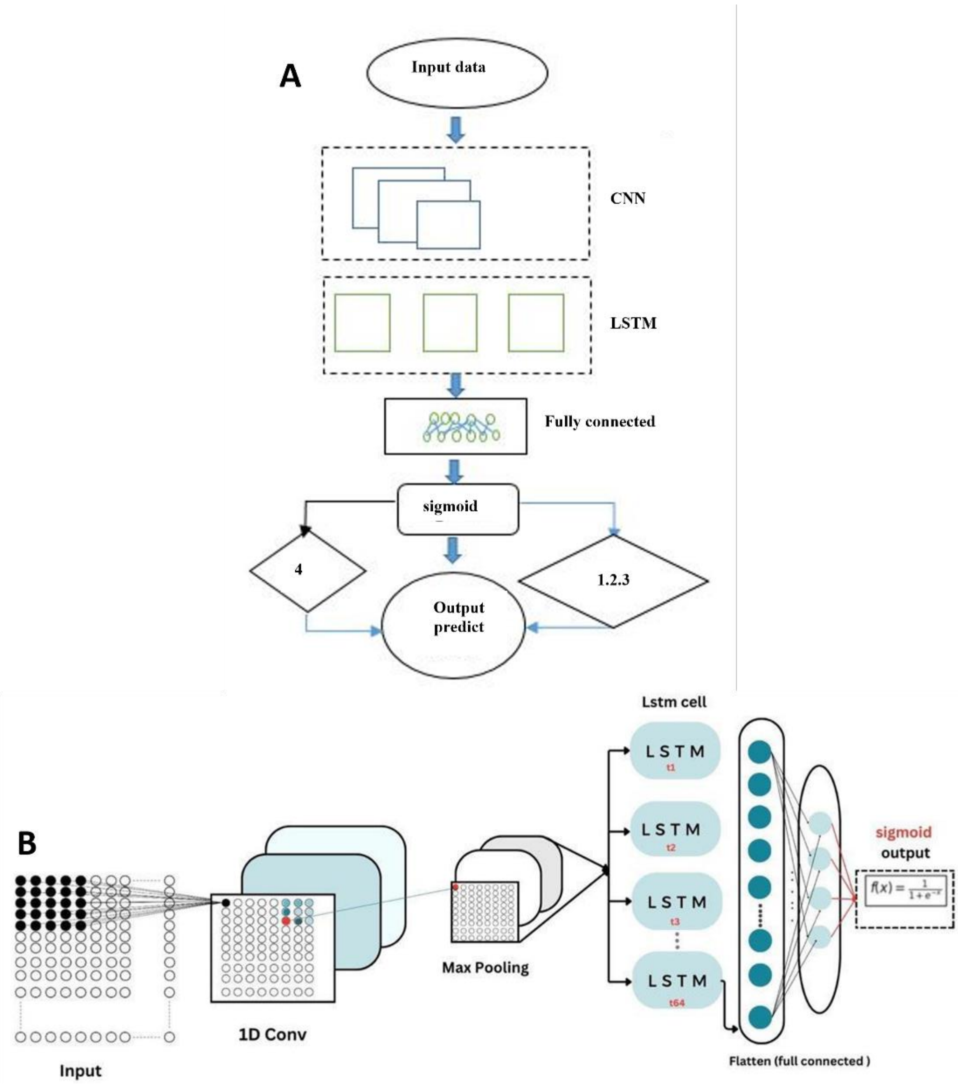

Figure 3.7. A) present CNN-LSTM structure, B) CNN-LSTM model.

The algorithm employed in this study, briefly described in Algorithm 3, embodies the fusion of CNN and LSTM methodologies. Specifically designed for the prediction of the failure state of an APU component, the model accommodates a total of 13 input and output resolution features. These features are systematically organized to predict distinct targets, thereby encapsulating a comprehensive approach to failure prediction within the specified domain.

The failure prediction within the proposed model is categorized into two distinct groups. The first category involves a binary rating system, where '1' indicates component failure and '0' signifies non-failure. The second category extends to multiple ratings, delineating specific failure types (Failure 1 - air leak on the customer, Failure 2 - air leak in the air dryer, Failure 3 - oil leak on the customer's compressor, and 4 indicating no failure). Table 3.1 comprehensively outlines the classified failures within these categories.

Table 3.1. Failure properties and classified.

No.	Failure Type	Component	Start	End
1	Air leak	Clients	28-02-2022 21:53	01-03-2022 02:00
2	Air leak	Air Dryer	23-03-2022 14:54	23-03-2022 15:24
3	Oil leak	Compressor	30-05-2022 12:00	30-05-2022 12:00

The overarching architecture of the proposed model is founded upon the CNN-LSTM framework, strategically designed to ensure the precise prediction of component failures. The structural representation of the model is visually depicted in the model chart presented in Figure 3.5.

```
Algorithm3: Hybrid CNN-LSTM model
# Input: X (a set of input features)
# Output: Y (predictions)
# Initialize an empty list to store predictions
Y = []
# For each x in X
For each x in X:
    # Apply CNN to feature x and store the result in Cx
    Cx = CNN(x)
    # Apply LSTM to Cx and store the result in Ox
    Ox = LSTM(Cx)
    # Apply sigmoid activation to Ox and store the result in Yx
    Yx = sigmoid(Ox)
    # Append Yx to the list of predictions Y
    Append Yx to Y
# End of loop
# Return the list of predictions Y
Return Y
```

Figure 3.8. Pseudocode of the CNN-LSTM model.

3.3. DATASET EXPLORATION

Compiled to position itself as a benchmark for predictive maintenance in 2022, the MetroPT dataset constitutes a crucial element within the Explainable Predictive Maintenance (XPM) initiative. Tailored specifically for an urban metro transmission line located in Porto, Portugal [104], and accessible on Zenodo [105], the dataset encompasses observations procured from Air Production units installed on the rooftops of metro vehicles, each serving diverse functionalities. Among these units, the secondary suspension system is responsible for sustaining the car's level, irrespective of the passenger load. The Air Production units play indispensable roles in the functionality of these vehicles, particularly during daytime operations.

The MetroPT dataset stands as a real-world repository with documented ground truth anomalies extracted from the maintenance reports of the relevant company. Its primary objective is to serve as a definitive reference dataset for predictive maintenance, facilitating impartial comparisons among diverse machine and deep learning algorithms employed to identify faults and anomalies. This, in turn, fosters the implementation of preventative maintenance strategies based on continuously monitoring sensor data streams. In recent years, many studies have concentrated on predictive maintenance, harnessing the advancements in machine and deep learning methodologies. PdM's overarching aim is to promptly forecast evolving and unforeseen failures through sustained monitoring of equipment conditions. Dynamic scheduling of maintenance plans is executed to minimize unplanned downtime and the associated financial ramifications. Moreover, identifying affected components and assessing failure severity contribute to the formulation of more efficient recovery plans. The absence of redundancy often precipitates an immediate withdrawal of trains for repair, with failures of this nature eluding traditional condition-based maintenance criteria defined by rigid thresholds [106].

A myriad of scholarly endeavors has delved into the realm of predictive maintenance, leveraging sophisticated deep learning methodologies. Recent literature encompasses a comprehensive survey elucidating critical facets of data-driven public distribution management within the domain of predictive maintenance. Furthermore, another survey illuminates advancements in the application of both machine learning and deep learning techniques for holistic traffic management within the railway industry [107].

A third manuscript delineates three pivotal research trajectories within the PdM domain: namely, failure prediction, Remaining Useful Life (RUL) estimation, and Root Cause Analysis (RCA). Operationally, the principal objective of predictive maintenance is to ameliorate operational challenges, curtail unforeseen interruptions and downtimes, and transition the maintenance paradigm from reactive to predictive. Within this framework, the early identification of such challenges holds the potential to avert trip cancellations and service disruptions, thereby conferring substantial advantages to both the operating company and passengers. To realize this objective, a monitoring system has been deployed within the APU, capturing both analog and digital signals and providing precise location coordinates for track and train waiting areas. These collected signals are transmitted to a remote server at five-minute intervals via the GSM network, operating at a frequency of 1 Hz. Over the period from January to June 2022, an average of 26 trips per day were recorded. This comprehensive dataset encompasses 21 features, encompassing analog sensor readings (pressure, temperature, and current consumption), as detailed in Table 3.2, which presents analog sensor data on the APU. Additionally, digital signals (control and discrete signals) are outlined in Table 3.3, representing digital sensor data on the APU. Furthermore, GPS information, including latitude, longitude, and speed, is presented in Table 3.4, encapsulating GPS data on the APU.

Table 3.2. MetroPT analog sensors.

Analog Sensors (These sensors measure pressure, temperature, and electrical current related to the APU)				
Num.	**Sensor**	**Symbol**	**Description**	**Units**
1	Compressor Pressure	TP2	Measures the pressure in the compressor	bar
2	Pneumatic Plate Pressure	TP3	Records the pressure generated at the pneumatic plate	bar
3	Command Pressure Switch Valve	H1	This valve is activated when the pressure reading exceeds the operational threshold of 10.2 bar	bar
4	Air Dryer Pressure Drop	DV_pressure	Detects the pressure drop resulting from the air dryer towers discharging water. When the reading is zero, it indicates that the compressor is operating under load.	bar
5	Reservoirs	Reservoirs	Records the pressure inside the air tanks installed on the trains	bar
6	Compressor Oil Temperature	Oil_Temperature	Measures the temperature of the oil in the compressor	°C
7	Air Flow	Flowmeter	Sensor calculates the airflow at the pneumatic control panel	m3/h
8	Motor Current	Motor _Current	Monitors the motor's current, with expected values of (i) close to 0A when the compressor is running, (ii) close to 4A when the compressor is operating, and (iii) close to 7A when the compressor is under load	A

Table 3.3. MetroPT Digital sensors.

Digital Sensors				
The APU incorporates eight digital sensors that provide binary data, indicating either a value of zero when they are inactive or one when a specific event activates them.				
Num.	**Sensor**	**Symbol**	**Description**	**Units**
1	Compressor Air Intake Valve	COMP	Generates an electrical signal representing the air intake valve's status on the compressor. It registers a value of one when there is no air entering the compressor	Binary
2	Compressor Outlet Valve	DV_electric	Controls the electrical signal for the compressor's outlet valve. When active, it indicates that the compressor is operating under load, while inactivity suggests that the compressor is loaded or overloaded.	Binary
3	Towers	Towers	Signal identifies which tower is currently drying the air and which one is engaged in draining the moisture extracted from the air. An active signal indicates that the second tower is in operation, while inactivity indicates that the first tower is functioning	Binary
4	Intake Valve Activation	MPG	Responsible for triggering the intake valve to initiate compressor operation under load when the APU's pressure falls below 8.2 bar. Consequently, it activates the COMP sensor, functioning similarly to the MPG sensor.	Binary
5	Low Pressure Signal	LPS	Signal activates when the pressure within the APU falls below 7 bar.	Binary
6	Pressure Switch	Pressure Switch	Activates when pressure is detected on the pilot control valve.	Binary
7	Oil Level	Oil Level	Indicates the oil level in the compressor. A reading of one signifies that the oil level is below the expected values.	Binary
8	Caudal_impulses	Caudal_impulses	Signal is generated by the altimeter and indicates the presence of airflow per second.	Binary

Table 3.4. MetroPT GPS Signals sensors.

GPS Signals				
The train is outfitted with a secondary GPS antenna designed to capture data pertaining to signal strength, speed, latitude, and longitude. When the train enters a tunnel and loses satellite connectivity, the acquisition system resets the GPS signal to zero.				
Num.	**Sensor**	**Symbol**	**Description**	**Units**
1	Longitude Position	gpsLong	This feature provides the longitude position in degrees	°
2	Latitude Position	gpsLat	It offers the latitude position in degrees	°
3	Speed	GPSSpeed	This feature records the speed in kilometers per hour	(km/h)
4	Signal Quality	GPSQuality	It signifies the quality of the GPS signal	

The APU of the train integrates a signal capture system, with the acquisition system adhering rigorously to established standards and guidelines governing railway equipment usage. The selection of sensors was meticulously grounded in the principles of Failure Mode and Effects Analysis (FMEA) and Failure Mode and Effects and Criticality Analysis (FMECA) specific to the APU. Notably, the dataset under consideration comprises an extensive total of 10,979,547 data points, each devoid of any missing values. Over a duration of six months, three instances of catastrophic failures were discerned, with two attributable to air leakage within the system and the third associated with an oil leak.

A stringent framework of protocols and standards has been meticulously devised to safeguard the safety, reliability, and efficiency of systems and equipment integral to metro transportation. These protocols and standards constitute an exhaustive compendium of regulations and requisites that the signal acquisition system incorporated into the APU of the train is obliged to adhere to. These encompass:

- TS EN 45545 - Railway applications - Fire protection in railway vehicles: These standard addresses fire protection measures in railway vehicles to safeguard passengers and railway assets.
- EN 50121 - Railway applications - Electromagnetic compatibility: This standard focuses on ensuring electromagnetic compatibility in railway

applications to prevent interference and ensure proper functioning of electronic systems.

- EN 50125 - Railway applications - Environmental conditions of equipment: This standard specifies environmental conditions that railway equipment must withstand, including temperature, humidity, and vibration.
- EN 50128 - Railway applications - Communications, signaling, and processing systems - Railway Software Control and protection systems: This standard deals with software used in railway control and protection systems, emphasizing safety and reliability.
- EN 50129 – Railway applications – Communication, signaling, and processing systems – Safety-related electronic systems for signals: This standard addresses the safety-related electronic systems used in railway signaling, ensuring integrity and safety.
- EN 50153 - Rolling Equipment - Protection provisions relating to electrical risks: This standard is concerned with safeguarding electrical systems in rolling stock.
- EN 50155 – Railway applications – Electronic equipment used in railway vehicles: It outlines requirements for electronic equipment used in railway vehicles to ensure durability and safety.
- EN 60529 - Specifications for degrees of protection provided by enclosures (IP code): This standard defines IP (Ingress Protection) codes, indicating the level of protection provided by enclosures against solid objects, dust, water, and other environmental factors.
- EN 61373 - Railway applications - Rolling stock - Shock and vibration tests: This standard provides testing procedures to evaluate the ability of railway equipment to withstand shocks and vibrations.
- IEC 60068 – Environmental tests: This standard outline testing method for assessing the durability of electronic equipment under various environmental conditions.
- IEC 60571 - Electronic equipment used in railway vehicles: It addresses electronic equipment used in rail vehicles, focusing on safety and reliability.
- IEC 61375-1 - Railway electronic equipment - Train communications network (TCN) - Part 1: General architecture: This standard defines the general

architecture of train communication networks, ensuring interoperability and efficiency.

- IEC 61375-2-1 - Electronic railway equipment - Train communications network (TCN) - Part 2-1: Wired train bus (WTB): This standard pertains to wired train bus systems, a critical part of train communication networks.
- IEC 61375-3-1 - Electronic railway equipment - Train communications network (TCN) - Part 3-1: Multi-function vehicle bus (MVB): This standard deals with multi-function vehicle bus systems within train communication networks, enabling multiple functions and data exchange.

Derived from maintenance reports and substantiated by the company's provided ground truth, the dataset encompasses three instances of catastrophic failures. Among these, two were precipitated by a system air leak, while the third was attributed to an oil leak.

- Air Leak on Air Dryer: This malfunction is ascribed to a flaw in the air pilot valve responsible for unsealing the drainpipes during compressor operation.
- Air Leakage on Customers: This complication arose due to air leakage in the pipe that supplies various system customers, encompassing spacers, suspension components, and others.
- Oil Leakage on the Compressor: Owing to the equipment's design, the absence of an oil signaling system to alert the train driver led to this issue. The resultant oil leak caused substantial damage to the compressor engine. With the compressor rendered inoperative, a decline in air pressure ensued, necessitating the suspension of the train.

As per the entity tasked with curating this dataset, two principal objectives are served: firstly, Failure Prediction, and secondly, the Identification of Components Associated with Failures. Concerning the inaugural undertaking of failure prediction, the aim is to discern an impending failure a minimum of two hours prior to the cessation of train operation, thereby facilitating its secure extraction from the tracks and enabling proactive maintenance measures. This necessitates the precise delineation of the failure type and the localization of the specific component implicated in the occurrence of the failure. Recent scholarly endeavors have leveraged the MetroPT dataset to proffer methodologies addressing the challenge of failure prediction. In the initial scholarly endeavor, a comprehensive analysis was undertaken to elucidate the intricacies of this predictive framework [108], A rule-based system was devised to orchestrate the generation of alerts concerning the compressor's status. In the subsequent investigation outlined in the second work [109], a deep learning paradigm centered on autoencoders was employed for the purpose of alert generation. Although both methodologies have yielded commendable outcomes, a notable space for enhancement persists, particularly in the realms of accuracy and interpretability.

- True Positive (TP): This occurs when the predicted failure interval overlaps with the observed failure interval.
- False Positive (FP): This happens when the expected failure interval does not overlap with the observed failure interval.
- True Negative (TN): This is the case when there is neither an expected failure nor an observed failure.
- False Negative (FN): This takes place when there is no expected failure, but an observed failure occurs.

The paramount objective is the reduction of both False Alarms (FP) and Missed Failures (FN) to mitigate the occurrence of failures during operational activities and circumvent unwarranted maintenance procedures. Furthermore, an essential facet involves the computation of the remaining useful life of components, facilitating informed decision-making by the management team regarding train removal without incurring service interruptions. Figure 3.9 illustrates the envisaged evaluation protocol for the anticipation of failures.

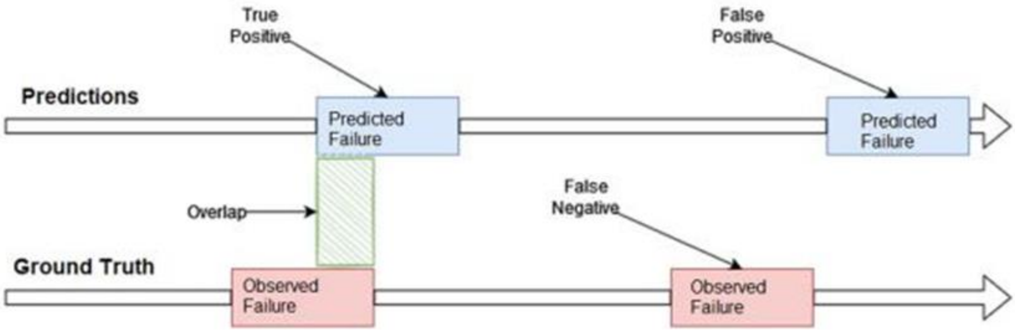

Figure 3.9. Evaluation protocol for predicting failures.

3.4. DATASET VISUALIZATION

The MetroPT dataset underwent a sequence of data preprocessing procedures. Initially, a comprehensive visualization and descriptive analysis were employed to adeptly portray its inherent characteristics.

An in-depth examination of the data structure yielded valuable insights, encompassing the enumeration and categorization of features into integer (INT), floating-point (float), or object data types. Memory consumption was scrutinized, and the presence of any missing values within the data frame was ascertained through the utilization of the info () function. The outcomes revealed that the 'timestamp' feature was categorized as 'object,' while the remaining attributes comprised ten 'int64' and ten 'float64' features. Notably, the dataset occupied a memory footprint of 1.7 gigabytes. Figure 3.10 present info details.

```
[6]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10773588 entries, 0 to 10773587
Data columns (total 21 columns):
 #   Column           Dtype
---  ------           -----
 0   timestamp        object
 1   TP2              float64
 2   TP3              float64
 3   H1               float64
 4   DV_pressure      float64
 5   Reservoirs       float64
 6   Oil_temperature  float64
 7   Flowmeter        float64
 8   Motor_current    float64
 9   COMP             int64
 10  DV_eletric       int64
 11  Towers           int64
 12  MPG              int64
 13  LPS              int64
 14  Pressure_switch  int64
 15  Oil_level        int64
 16  Caudal_impulses  int64
 17  gpsLong          float64
 18  gpsLat           float64
 19  gpsSpeed         int64
 20  gpsQuality       int64
dtypes: float64(10), int64(10), object(1)
memory usage: 1.7+ GB

descibe data frame
```

Figure 3.10. MetroPT features info details.

Building upon a foundational comprehension of the dataset, a meticulous statistical analysis was conducted utilizing the describe () function. This approach facilitated an in-depth portrayal of the data through essential statistical metrics, including maximum and minimum values, mean, and standard deviation. Additionally, quartile values were computed, partitioning the data into 25% increments, delineating the first and third quarters, and the median representing the midpoint. These statistical insights are indispensable for elucidating the distribution of the data and fostering exploratory data analysis. Moreover, they play a pivotal role in discerning the range between maximum and minimum values, providing valuable perspectives on the variance among data points. Figure 3.11. presents feature descriptions.

```
[7]: df.describe().transpose()
```

[7]:

	count	mean	std	min	25%	50%	75%	max
TP2	10773588.0	1.152184e+00	3.075296	-0.030000	-0.00800	-0.008000	-0.006000	10.876000
TP3	10773588.0	8.974608e+00	0.700696	0.006000	8.48400	8.984000	9.492000	10.408000
H1	10773588.0	7.751421e+00	3.051447	-0.034000	8.23200	8.746000	9.290000	10.414000
DV_pressure	10773588.0	-2.454095e-02	0.148657	-0.038000	-0.03200	-0.028000	-0.026000	8.326000
Reservoirs	10773588.0	1.565051e+00	0.090163	1.350000	1.47000	1.590000	1.638000	2.054000
Oil_temperature	10773588.0	6.730720e+01	5.383852	13.875000	63.67500	68.325000	71.075000	97.900000
Flowmeter	10773588.0	2.039515e+01	3.743607	18.834719	19.01225	19.040281	19.255188	43.072406
Motor_current	10773588.0	2.383179e+00	2.193381	-0.012500	0.00250	3.705000	3.837500	9.685000
COMP	10773588.0	8.698926e-01	0.336422	0.000000	1.00000	1.000000	1.000000	1.000000
DV_eletric	10773588.0	1.301137e-01	0.336428	0.000000	0.00000	0.000000	0.000000	1.000000
Towers	10773588.0	9.347704e-01	0.246931	0.000000	1.00000	1.000000	1.000000	1.000000
MPG	10773588.0	8.698921e-01	0.336422	0.000000	1.00000	1.000000	1.000000	1.000000
LPS	10773588.0	6.281380e-03	0.079006	0.000000	0.00000	0.000000	0.000000	1.000000
Pressure_switch	10773588.0	0.000000e+00	0.000000	0.000000	0.00000	0.000000	0.000000	0.000000
Oil_level	10773588.0	2.784588e-07	0.000528	0.000000	0.00000	0.000000	0.000000	1.000000
Caudal_impulses	10773588.0	1.488919e-03	0.038558	0.000000	0.00000	0.000000	0.000000	1.000000
gpsLong	10773588.0	-4.384534e+00	4.317794	-9.130040	-8.65891	-8.542650	0.000000	0.000000
gpsLat	10773588.0	2.091144e+01	20.592543	0.000000	0.00000	41.151900	41.188200	41.949000
gpsSpeed	10773588.0	4.913657e+00	11.518220	0.000000	0.00000	0.000000	0.000000	323.000000
gpsQuality	10773588.0	5.076832e-01	0.499941	0.000000	0.00000	1.000000	1.000000	1.000000

Figure 3.11. MetroPT features Describe () details.

Following this, the existence of null and missing values within each feature of the data frame was discerned through the implementation of the isnull () function, disclosing that the dataset was devoid of any vacant or missing values. To gauge the diversity inherent in each feature, the unique () function was employed to ascertain the count of distinct values within each feature. Figure 3.12 A and B present features missing and unique values.

```
print("\nMissing value count:")
print(df.isnull().sum())
```

A

```
Missing value count:
timestamp          0
TP2                0
TP3                0
H1                 0
DV_pressure        0
Reservoirs         0
Oil_temperature    0
Flowmeter          0
Motor_current      0
COMP               0
DV_eletric         0
Towers             0
MPG                0
LPS                0
dtype: int64
```

```
print("\nUnique value count:")
print(df.nunique())
```

B

```
Unique value count:
timestamp          10773588
TP2                    5440
TP3                    5173
H1                     5198
DV_pressure            3712
Reservoirs              317
Oil_temperature        3165
Flowmeter              1911
Motor_current          3196
COMP                      2
DV_eletric                2
Towers                    2
MPG                       2
LPS                       2
dtype: int64
```

covert object to data time

Figure 3.12. A) MetroPT features isnull () details. B) MetroPT features Unique () details.

```
df['timestamp'] = pd.to_datetime(df['timestamp'])

df.dtypes
```

A

```
timestamp          datetime64[ns]
TP2                       float64
TP3                       float64
H1                        float64
DV_pressure               float64
Reservoirs                float64
Oil_temperature           float64
Flowmeter                 float64
Motor_current             float64
COMP                        int64
DV_eletric                  int64
Towers                      int64
MPG                         int64
LPS                         int64
dtype: object
```

```
df.info()
```

B

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10773588 entries, 0 to 10773587
Data columns (total 14 columns):
 #   Column           Dtype
---  ------           -----
 0   timestamp        datetime64[ns]
 1   TP2              float64
 2   TP3              float64
 3   H1               float64
 4   DV_pressure      float64
 5   Reservoirs       float64
 6   Oil_temperature  float64
 7   Flowmeter        float64
 8   Motor_current    float64
 9   COMP             int64
 10  DV_eletric       int64
 11  Towers           int64
 12  MPG              int64
 13  LPS              int64
dtypes: datetime64[ns](1), float64(8), int64(5)
memory usage: 1.1 GB
```

Figure 3.13. A) MetroPT features “timestamp” type details. B) MetroPT features “timestamp” type details.

Upon confirming the feature types, it was observed that the 'timestamp' feature was categorized as 'object.' Subsequently, a prudent adjustment was made by converting its type to 'DateTime,' thereby facilitating the transformation of string representations into Date Time objects. This conversion was duly validated and corroborated through the inspection conducted via the info() function. Figure 3.13 A and B illustrate timestamp type.

Following a comprehensive phase of data analysis and assimilation, data visualization emerged as a pivotal undertaking, aimed at extracting insights and adeptly portraying the inherent characteristics of the dataset. This crucial step in data preparation was instrumental in optimizing the dataset for subsequent model utilization, thereby ensuring enhanced model accuracy and performance. In the initial stages, a strategic classification of features into two distinct groups, namely analog signals and digital signals, was executed. Signals specific to geographical coordinates, namely longitude and latitude, were systematically excluded. Leveraging the Matplotlib library for data visualization, it was discerned that certain signals exhibited a constant pattern across all vehicle operating states. Consequently, these unvarying signals were judiciously eliminated from consideration. The features expunged from the data frame encompassed ["gpsLong," "gpsLat," "gpsSpeed," "gpsQuality," "Pressure_switch," "Oil_level," "Caudal_impulses"]. Figure 3.14 presents drop signals, and Figure 3.15. presents analogue signals visualization, while Figure 3.16. presents digital signals visualization.

```
df= df.drop(["gpsLong","gpsLat","gpsSpeed","gpsQuality","Pressure_switch","Oil_level","Caudal_impulses"],axis=1)
```

```
df.head(5)
```

	timestamp	TP2	TP3	H1	DV_pressure	Reservoirs	Oil_temperature	Flowmeter	Motor_current	COMP	DV_eletric	Towers	MPG	LPS
0	2022-01-01 06:00:00	-0.012	9.758	9.760	-0.028	1.576	63.350	19.049625	3.9550	1	0	1	1	0
1	2022-01-01 06:00:01	-0.012	9.760	9.760	-0.028	1.578	63.250	19.049625	4.0275	1	0	1	1	0
2	2022-01-01 06:00:02	-0.010	9.760	9.760	-0.028	1.578	63.325	19.040281	3.9450	1	0	1	1	0
3	2022-01-01 06:00:03	-0.012	9.756	9.756	-0.030	1.576	63.200	19.040281	3.9300	1	0	1	1	0
4	2022-01-01 06:00:04	-0.012	9.756	9.756	-0.030	1.578	63.150	19.049625	3.9950	1	0	1	1	0

Figure 3.14. MetroPT drop signals.

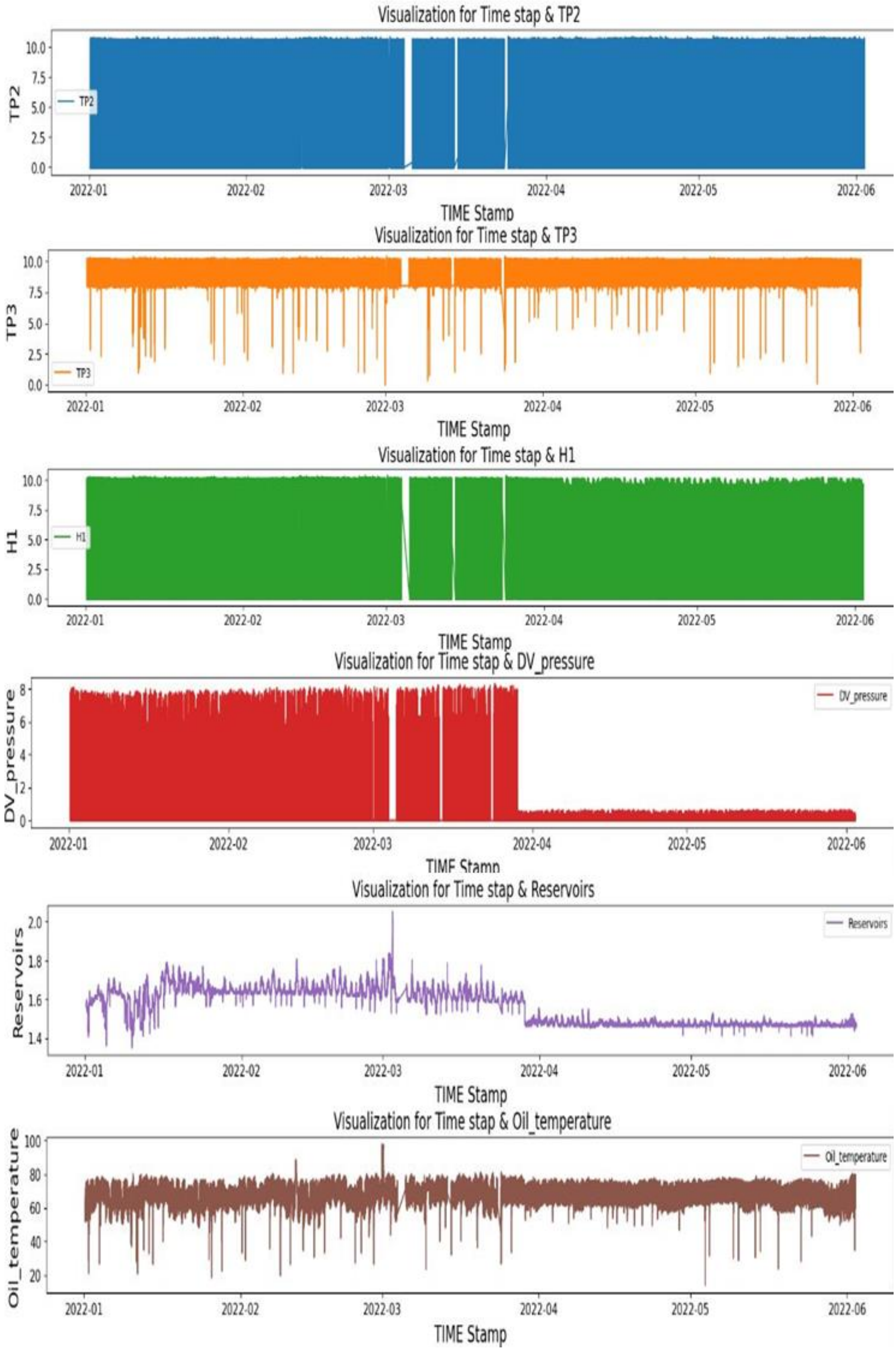

Figure 3.15. MetroPT analogue signals visualization.

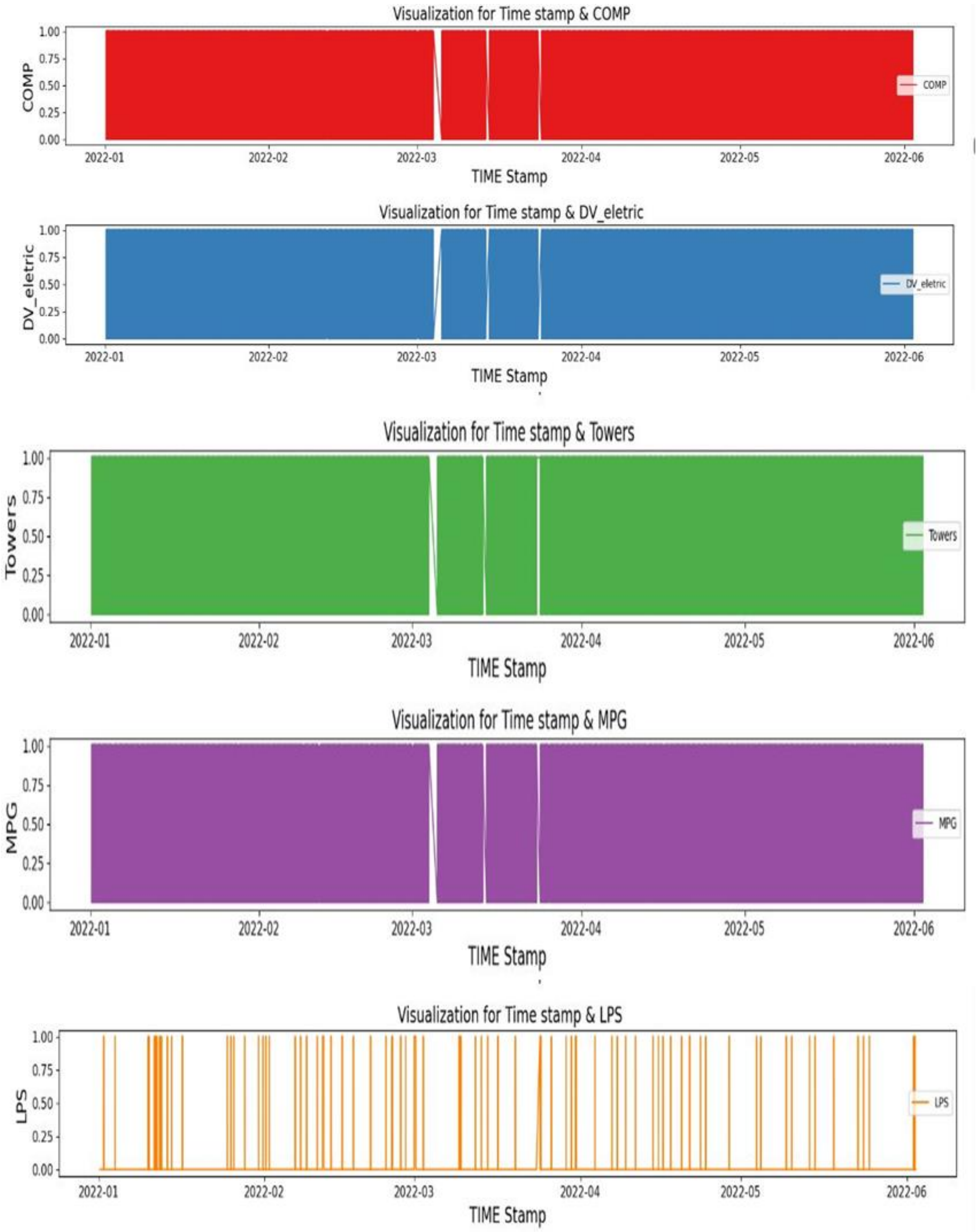

Figure 3.16. MetroPT digital signals visualization.

3.5. FEATURE ENGINEERING

Subsequent to the visualization and descriptive analysis of the data, a conspicuous observation surfaced: the dataset lacked the requisite labels essential for classification [104]. In the inaugural phase of label assignment, a binary classification schema (0,1) was implemented. Instances aligned with the three documented failures were assigned a label of "1," denoting the occurrence of failure in the air production unit. Conversely, instances falling outside the designated time frames, indicative of the absence of failure, were designated the label "0,". In formulating the second set of labels, failures were systematically categorized into three distinct classes. The initial class was designated a label of "1" corresponding to the first recorded failure, the subsequent class was assigned a label of "2" for the second documented failure, and the third class received the label "3" denoting the third failure. A fourth class was introduced with the label "4," encapsulating instances characterized by the absence of failures. This classification was contingent upon the precise timestamps associated with each recorded failure as shown in Table 3.1.

Upon the generation of these labels, an observation surfaced regarding the imbalanced distribution of values. To rectify this issue, the under-sampling technique was applied to both the first and second sets of labels, ensuring a balanced representation of classes and thereby rendering the data conducive to modeling and analysis.

Following label balancing, normalization was implemented on the analog signals characterized by continuous values, thereby scaling the data to conform to a predetermined range, typically (0, 1). Post-normalization, a stacking technique was invoked to resample the data, further optimizing its suitability for subsequent processing.

Subsequently, a meticulous organization, sorting, and partitioning of the data ensued, resulting in three distinct subsets: an 80% training set, a 10% validation set, and a 10% test set. These delineated subsets served as the foundational components for the training of the proposed model and the subsequent evaluation of its performance in predicting failures.

PART 4

EXPERIMENT RESULT AND DISCUSSION

4.1. HYPERPARAMETER

A multitude of fields and domains witness the extensive application of DL algorithms. The optimization of these algorithms' performance is inherently linked to the meticulous selection of optimal parameters, a process deemed fundamental as it profoundly influences both model performance and accuracy. Consequently, the challenge inherent in the process of parameter selection assumes significance, given its direct impact on the augmentation of model performance [110]. Therefore, parameters are intricately selected and fine-tuned to align with the characteristics of the specific dataset and its dimensions, along with the algorithms seamlessly integrated into the model.

In the pursuit of this research endeavor, inspiration for parameter selection was drawn from prior study models, iterative experimentation, and the progress achieved by scholars in the realm of optimal parameter selection. Initially, a spectrum of distinct values for each hyperparameter, possessing the capacity to exert influence on model performance, was systematically examined. These hyperparameters encapsulated pivotal facets such as the learning rate, network depth (expressed in terms of layer count), units allocated per layer, types of activation functions, batch size, and other relevant parameters. The thoroughness of this expansive exploration facilitated a comprehensive analysis of the repercussions of diverse parameter values on model performance.

This dedicated effort reached its pinnacle with the discerning selection of pertinent hyperparameters for our model, culminating in a substantial augmentation of its performance and accuracy. The meticulously tuned hyperparameters of the envisaged

CNN-LSTM model are exhaustively delineated in Table 4.1, revealing noteworthy outcomes and outstanding predictive proficiency. These results underscore the authentic efficacy of deploying hyperparameter optimization techniques to achieve elevated model performance, thereby fostering the advancement of more exacting and efficacious predictive maintenance applications [111].

Table 4.1. Hyperparameter utilized in the CNN-LSTM model.

Hyper parameter	**Value**
Activation function	Relu
Dropout	0.2
Loss function	binary_crossentropy
Epoch	10
Batch size	12
optimizer	Adam

4.2. IMPLICATION OF THE FINDINGS

The hybrid CNN-LSTM model, conceived for the anticipation of failures in metro air production unit components, undergoes a training regimen utilizing the Metro PT suite. The execution of this research is conducted within the flexible Jupyter Notebook platform, harnessing the capabilities of the Python 3.7 programming language for the intricate construction of the model. All experiments are conscientiously executed on a computing system outfitted with an Intel Core i7 CPU and a robust 8 GB of RAM. This configuration guarantees substantial computational prowess, thereby ensuring the adequacy of computational resources for our research pursuits.

To address this, we have taken on the task of creating two distinct categories. The first pertains to binary classification, while the second is tailored for multi-class classification. This dual classification framework underpins our model's approach, systematically addressing the unique challenges presented by each label distinction. This approach is implemented in the proposed hybrid CNN-LSTM model for predicting failures contained within the data set, and which has been registered by the organization responsible for preparing the data.

4.2.1. Binary Classification

This section of the chapter presents the models employed in analyzing the Metro PT dataset for predicting APU component failure through binary classification (where "1" denotes component failure and "0" denotes non-failure). Following data preparation and processing, the correlation matrix is utilized to assess the relationships among the features, visually depicted in Figure 4.1. Subsequently, these features undergo further preparation before integration into the model, involving the removal of seven features. This results in a total of 13 inputs and two binary outputs (1 and 0) being fed into the models.

The dataset is partitioned into three groups: an 80% training group, a 10% validation group, and another 10% designated as the test group. Consequently, the features are primed for binary classification using various models (CNN, LSTM, CNN-LSTM), and the outcomes of each model are compared independently.

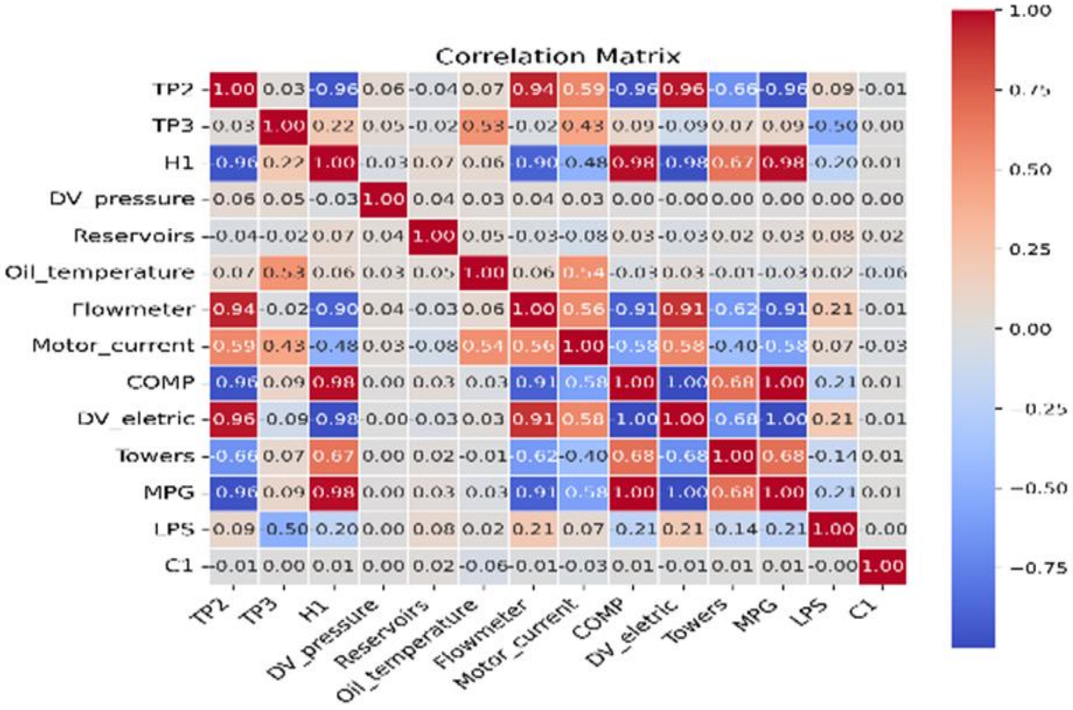

Figure 4.1. Correlation matrix of MetroPT features.

4.2.1.1. CNN Binary Classification

Upon analyzing the correlation matrix to unveil inter-feature relationships, the specified features are removed from the initial dataset. This dataset is subsequently divided into three distinct groups. The objective is to ready it for a CNN designed for forecasting component failure. The CNN model is configured with 13 input and output features for binary classification, where "1" represents component failure and "0" indicates no failure. Its customization aims at predicting the likelihood of failure for an APU component.

The CNN model architecture comprises three key layers: a one-dimensional convolutional layer, a pooling layer, and a flattening layer. The convolutional layer incorporates 64 filters and a kernel size of 3, utilizing the Rectified Linear Unit (ReLU) activation function. This is followed by a one-dimensional max pooling layer with a pool size of 2. The flattening layer includes 50 nodes activated by ReLU, and a dropout layer (dropout rate = 0.2) is added to prevent potential overfitting. Another layer, node 1, utilizes the "sigmoid" activation function to make the final classification decision based on the selected features. For optimization, the model employs the "binary_crossentropy" loss function and the "Adam" optimizer. Evaluating the model's performance and accuracy utilizes accuracy metrics, visually represented by Figure 4.2 showcasing the CNN model's architecture for binary classification. Additionally, Table 4.2 outlines the achieved results of the CNN model, using the specified hyperparameters in Table 4.1.

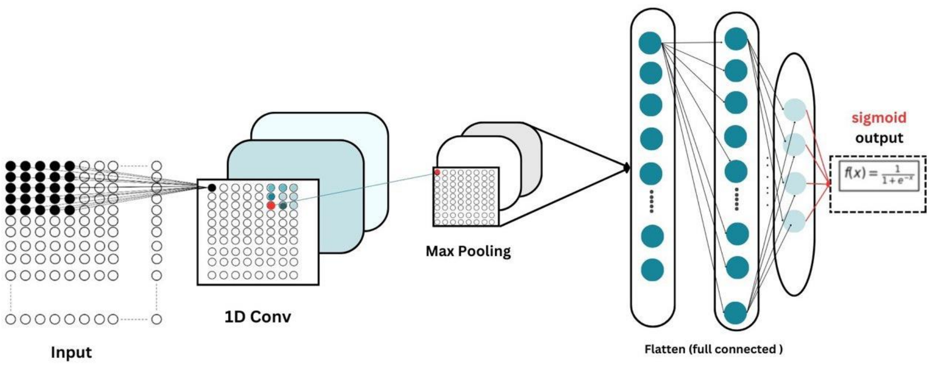

Figure 4.2. CNN model for binary class.

Table 4.2. CNN model result.

Method	The proposed (Binary class) CNN
Accuracy	**90.64%**

4.2.1.2. LSTM Binary classification

After employing the correlation matrix to discern feature relationships and subsequently eliminating them from the initial dataset, the data undergoes partitioning into three distinct groups. This segregation precedes the preparation for the LSTM network model, aimed at forecasting component failure. Configured with 13 input and output features for binary classification (where "1" signifies component failure and "0" indicates non-failure), the LSTM model is specialized to predict the probability of failure for an APU component.

The LSTM model architecture consists of four layers. The initial three layers each contain 50 units, while the fourth and final layer introduces a dense layer with a single node using the "sigmoid" activation function to determine the classification based on chosen features. Utilizing the 'binary_crossentropy' as the loss function and the 'Adam' optimizer, the model's structure is represented visually in Figure 4.3, showcasing its design for binary classification. Moreover, Table 4.3 details the outcomes attained by the LSTM model, employing the hyperparameters specified in Table 4.1.

Table 4.3. LSTM model result.

Method	The proposed (Binary class) LSTM
Accuracy	**94.13%**

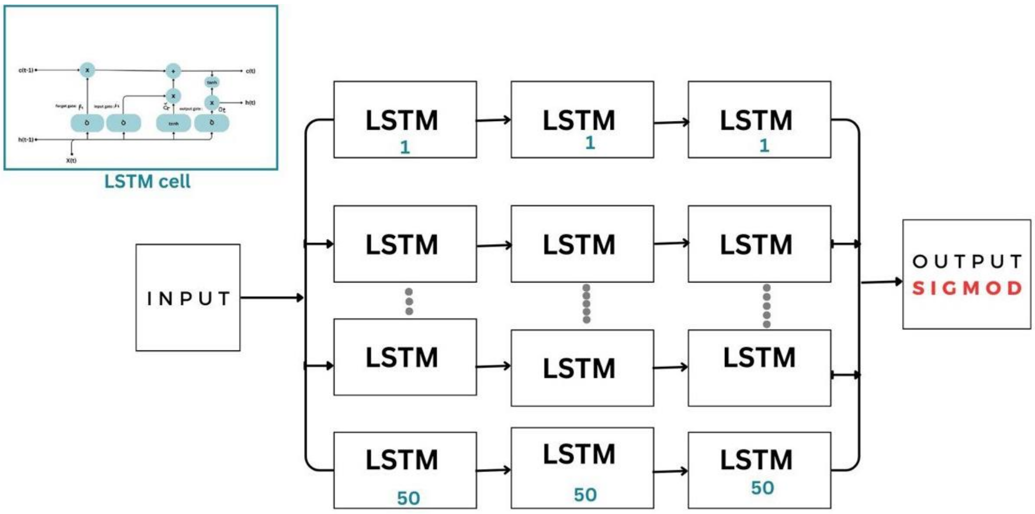

Figure 4.3. LSTM model for binary class.

4.2.1.3. CNN-LSTM Binary Classification

A correlation matrix was employed to establish a connection between the features and the binary classification target, visually represented in Figure 4.1. Subsequently, the hybrid model underwent execution both prior to and subsequent to the removal of seven features, with results indicating nominal disparities. Following this phase, the data underwent meticulous partitioning and preparation for input into the model, involving the allocation of an 80% training set, a 10% validation set, and a 10% test set.

The envisaged hybrid CNN-LSTM model accommodates 13 input and output features, specifically tailored for predicting the likelihood of failure in an APU component, adhering to a binary classification paradigm (where '1' signifies component failure and '0' denotes non-failure). The model's architectural framework predominantly leverages CNN-LSTM for precise failure prediction. The CNN component integrates a one-dimensional convolutional layer featuring 64 filters and a kernel size of 3, employing the Rectified Linear Unit (ReLU) as the activation function. Subsequently, a one-dimensional max-pooling layer with a pool size of 2 is introduced, accompanied by a dropout layer (dropout rate = 0.2) designed to mitigate potential overfitting. The output of this layer seamlessly transitions into the LSTM model, characterized by 64 units. The LSTM outputs are further directed to a flat layer equipped with 64 units, functioning as a fully connected layer adept at converting multidimensional features into one-dimensional data. ReLU is employed as the activation function for this layer, and a dropout layer (dropout rate = 0.2) is judiciously incorporated to address overfitting concerns. Lastly, the model integrates a dense output layer housing 1 unit, responsible for delivering the ultimate classification decision based on the selected features. The "sigmoid" activation function governs this layer, with "binary_crossentropy" serving as the loss function and the "Adam" optimizer. Model performance and accuracy undergo evaluation through accuracy metrics, and Figure 4.4 vividly illustrates the architectural structure of the model tailored for binary classification.

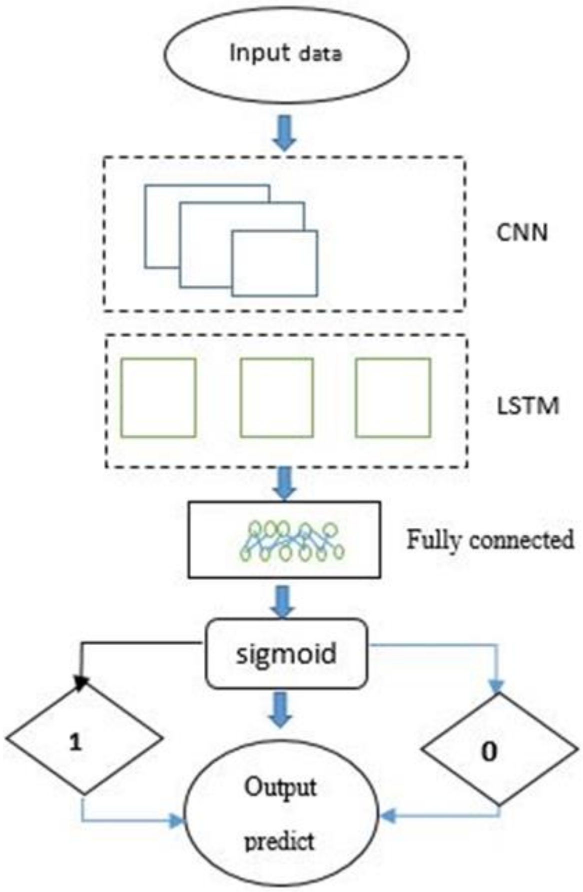

Figure 4.4. CNN-LSTM model structure for binary classification.

The hybrid model undergoes training using an 80% training set, with its performance meticulously assessed through a 10% validation set. Following the training and validation phases, the model undergoes a rigorous evaluation utilizing a dedicated 10% test dataset. Key evaluation metrics, including accuracy and F-Score, are systematically computed to provide a comprehensive assessment of the model's performance and accuracy. The outcomes of the Predictive Maintenance (PdM) CNN-LSTM model for binary classification notably showcase its efficacy in predicting failure scenarios with a commendable level of accuracy. This encompasses the model's adeptness in identifying components prone to failure, facilitating proactive maintenance measures. Furthermore, the model demonstrates efficiency in scenarios where components operate without any failure, thereby reducing unnecessary maintenance efforts. A comparative analysis of the performance results of the proposed model against relevant work utilizing a Random Forest machine learning model on the MetroPT dataset [112], reveals that our hybrid CNN-LSTM model attains superior predictive accuracy. This suggests enhanced speed and accuracy in error detection and diagnosis, surpassing the efficacy of previous models. The comparative metrics results are meticulously presented in Table 4.4.

Table 4.4. Overall proposed study results for binary-class.

Method	Previous study for Binary class Random Forest [112]	CNN	LSTM	The proposed hybrid **(Binary class) CNN-LSTM**
F-score	85%	X	X	**92%**
Accuracy	84%	90.64%	94.13%	**92.84%**
Recall	X	X	X	**0.92**
Precision	X	X	X	**0.93**

The accuracy and performance of the model undergo additional scrutiny through the utilization of a confusion matrix. As illustrated in Figure 4.5, the matrix delineates the following values: TP (True Positives) = 7697, FP (False Positives) = 550, FN (False Negatives) = 625, and TN (True Negatives) = 7545. The discernment drawn from these values underscores the commendable performance of the model.

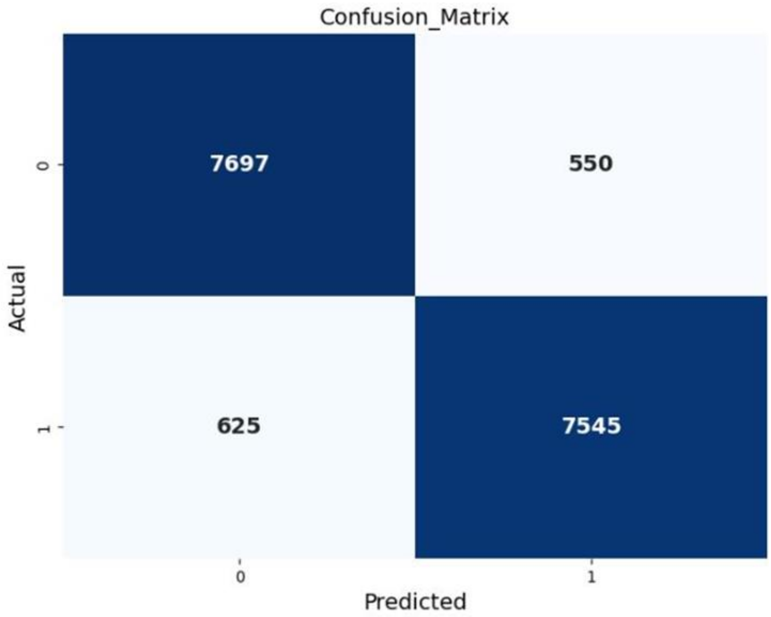

Figure 4.5. Confusion matrix for binary classification.

The model achieves an impressive accuracy rate, reaching approximately 92.7%, demonstrating exceptional proficiency in identifying true positive cases with a recall rate of about 92.5%. Furthermore, the model accurately identifies positive cases with a high prevalence, registering approximately 93.3%. The F-1 factor serves as a noteworthy metric, highlighting the delicate equilibrium between precision and recall in the model's performance[113].

Accuracy: This metric gauge the overall accuracy of the model and can be calculated using the formula (1):

$$\text{Precision} = TP / (TP + FP) \tag{4.1}$$

Recall: Recall assesses the completeness of the model and is computed as indicated in (2):

$$\text{Recall} = TP / (TP + FN) \tag{4.2}$$

F-score: The F-score represents the weighted average of recall and precision and is particularly valuable when dealing with imbalanced training data. The F-score is determined using the formula (3):

$$F - score = (2 * Precision * Recall)/ (Precision) + Recall)) \tag{4.3}$$

The model's accuracy exhibits improvement with a progressive increase in the number of epochs, as visually depicted in Figure 4.6. The graphical representation elucidates the discernible enhancement in accuracy and concurrent reduction in loss with respect to the increment in epoch size.

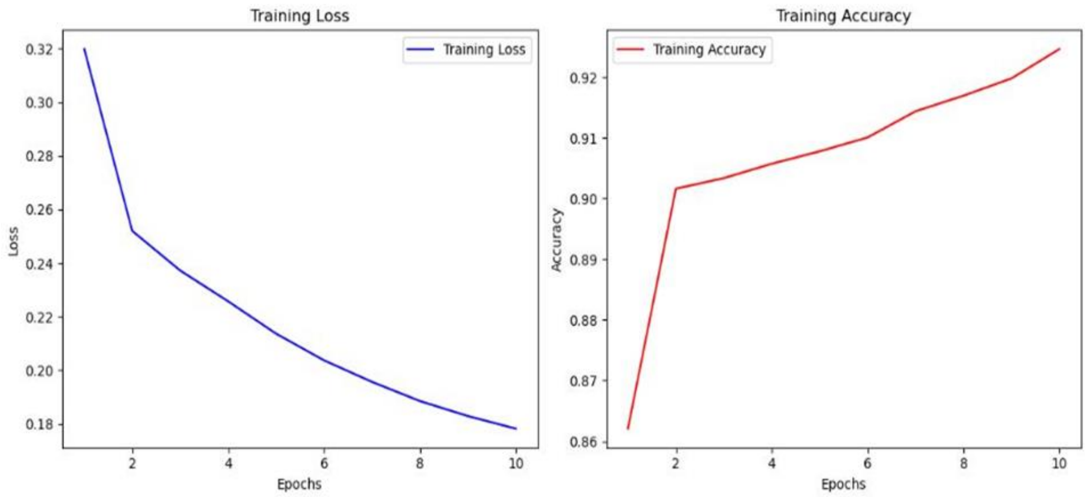

Figure 4.6. Training loss & Training Accuracy for binary classification.

4.2.2. Multi Classification

This section introduces a multi-classification methodology designed for categorizing failures within a dataset utilizing various models (CNN-LSTM, CNN, LSTM). Failures are classified into four distinct categories: the first category is labeled as "1" for the initial recorded failure, the second category is designated as "2" for the second recorded failure, the third category is assigned "3" for the third failure, and the fourth category is represented by the number "0," indicating No failure, as illustrated in Table 4.5.

Table 4.5. Multi-class failures types.

Multi-Class	
Failure	**Non Failure**
Air leak =1	4
Air leak =2	
oil leak =3	

Following the classification of failures into these four categories, the dataset was partitioned into three distinct sets: an 80% training set, a 10% validation set, and a 10% test set. Hyperparameters were systematically applied to the model, with detailed specifications provided in Table 4.1. Thirteen input features after deleting seven features and four output features were prepared, and the failure state of the APU component was predicted based on the classified failure states (as per Table 4.5). The proposed models were tested separately for accuracy.

4.2.2.1. CNN Multi Classification

This section introduces the CNN model tailored for multiple classification. After categorizing failures into four distinct classes based on the recorded multi-class target failures, as outlined in Table 4.5, the CNN model's architecture is designed to handle 13 input features and predict the failure state of the APU component according to the classified failure states (refer to Table 4.5).

The CNN component encompasses a one-dimensional convolutional layer with a filter size of 64 and a kernel size of 3, employing the rectified linear unit (ReLU) activation function. This is followed by a one-dimensional maximum pooling layer with a pooling size of 2 and a flattening layer consisting of 50 nodes. The ReLU activation function is applied to this layer, and a dropout layer with a rate of 0.2 is incorporated. The model is concluded with a dense output layer comprising 4 units, utilizing a "sigmoid" activation function to organize the outputs (1, 2, 3, 4). The loss function is denoted by "binary_crossentropy," optimized through the "Adam" optimizer. Model parameters align with those specified in Table 4.6. Figure 4.7 visually depicts the model structure for multiple classification.

Table 4.6. CNN model result.

Method	The proposed (Multi class) CNN
Accuracy	**99.34%**

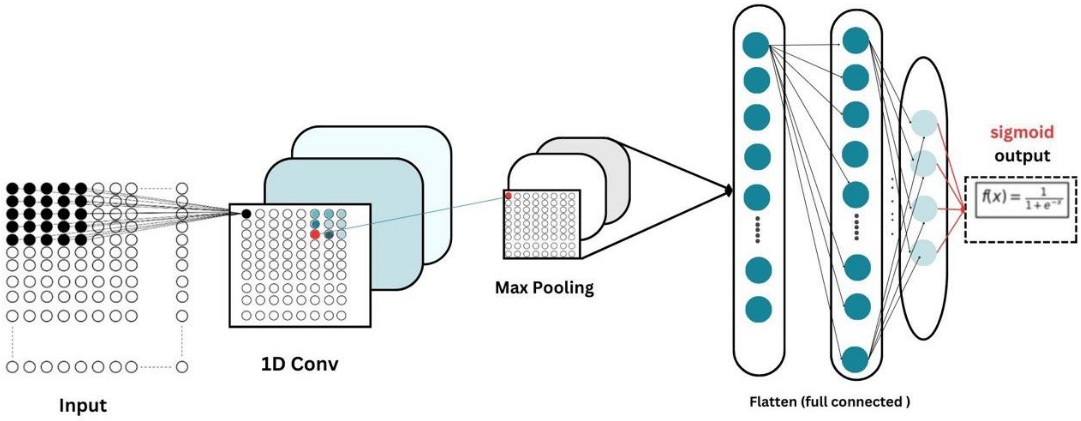

Figure 4.7. CNN model architecture for multi-class.

4.2.2.2. LSTM Multi Classification

This section introduces the LSTM model tailored for multiple classification. Following the categorization of failures into four distinct classes based on the recorded multi-class target, as outlined in Table 4.5, the LSTM model's structure is designed to accommodate 13 input features and predict the failure state of the APU component based on the classified failure states (refer to Table 4.5).

The LSTM model is composed of four layers, with the first, second, and third layers each containing 50 units. The fourth and final layer involves the addition of a dense layer with a single node and a "sigmoid" activation function, responsible for delivering the final classification decision based on the selected features. The loss function is denoted by "binary_crossentropy," and the "Adam" optimizer is employed.

Model parameters align with those specified in Table 4.1. Figure 4.8 provides a visual representation of the model structure for multiple classification. Standard performance evaluation involves accuracy metrics, encompassing training on an 80% training set and validation on a 10% validation set. Subsequently, a dedicated 10% test dataset is utilized to assess the model, employing metrics such as accuracy. The Table 4.7 presents the detailed results of the model.

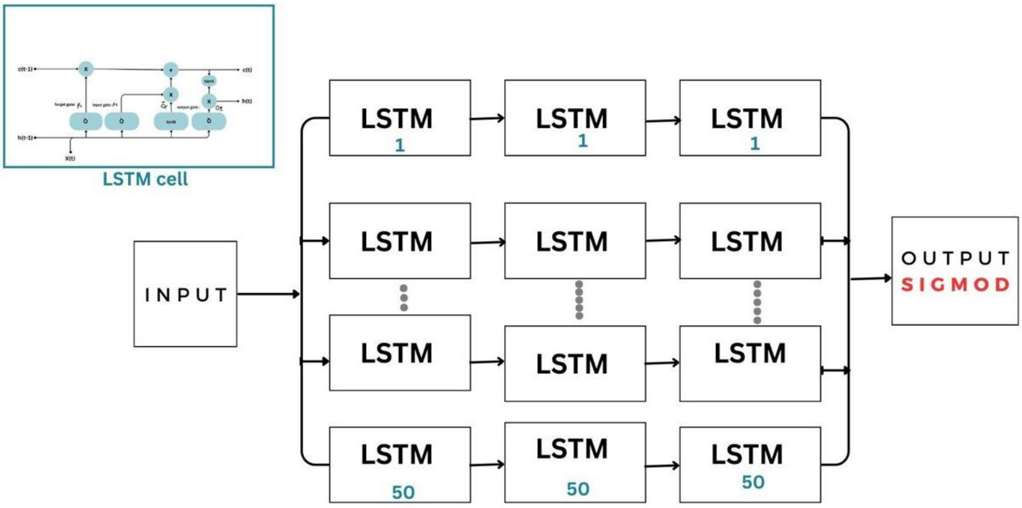

Figure 4.8. LSTM model architecture for multi-class.

Table 4.7. LSTM model result.

Method	The proposed (Multi class) LSTM
Accuracy	**98.26%**

4.2.2.3. CNN-LSTM Multi Classification

This section introduces a multi-classification methodology for categorizing failures within a dataset, employing a hybrid CNN-LSTM model [111]. Failures are classified into four distinct categories: the first category is denoted by a "1" for the initial recorded failure, the second category by a "2" for the second recorded failure, the third category by a "3" for the third failure, and the fourth category is represented by a "0" signifying no failure, as elucidated in Table 4.3.

After the categorization of failures into four classes, the dataset underwent a partitioning into three distinct sets: an 80% training set, a 10% validation set, and a 10% test set. Hyper-parameterization was systematically applied to the model, with detailed specifications outlined in Table 4.1. The architecture of the hybrid CNN-LSTM model is designed to accommodate 13 input and output features, predicting the failure state of the APU component based on the classified failure states (as per Table 4.5). The CNN component encompasses a one-dimensional convolutional layer (filter size: 64, kernel size: 3) with Rectified Linear Unit (ReLU) activation, succeeded by a one-dimensional maximum pooling layer (pooling size: 2) and a dropout layer (dropout rate: 0.2) implemented for managing overfitting. The subsequent LSTM layer, comprised of 64 units, processes the output from the CNN component. The LSTM outputs are then directed into a dense layer (64 units) that functions as a fully connected layer, adept at converting multidimensional features into one-dimensional data. The ReLU activation function is applied to this layer, accompanied by a dropout layer (dropout rate: 0.2). The model concludes with a dense output layer, housing 4 units and utilizing a "sigmoid" activation function to organize the outputs (1, 2, 3, 4). The loss function is denoted by "binary_crossentropy," optimized through the "Adam" optimizer. Figure 4.9 provides a graphical representation of the model structure for multi-classification. Model performance evaluation centers around accuracy metrics, entailing training with an 80% training set and validation with a 10% validation set. Subsequently, a dedicated 10% test dataset is employed for model evaluation, utilizing metrics such as accuracy and F-Score.

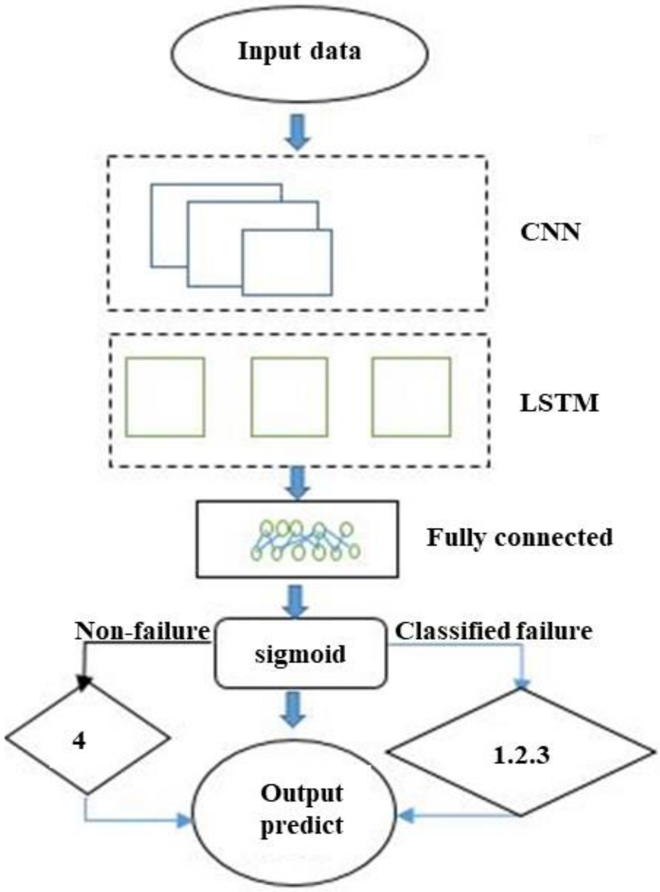

Figure 4.9. CNN-LSTM model structure for Multi-class.

The evaluation outcomes of the multi-classification CNN-LSTM model underscore its efficacy in predicting classified failure scenarios and discerning potential components prone to failure, thereby mitigating unnecessary maintenance efforts. A comparative analysis with preceding research, employing a Random Forest machine learning model on the MetroPT dataset, reveals that the deep hybrid model attains superior predictive accuracy. This advancement is presumed to contribute positively to fault detection and diagnosis. The detailed comparison metrics are presented in Table 4.8.

Table 4.8. Overall proposed study results for multi-class.

Method	Previous study for multi-class Random Forest [112]	CNN	LSTM	The proposed hybrid (Multi-class) CNN-LSTM
F-score	97%	X	X	**0.99**
Accuracy	87%	99.34%	98.26%	**99.5%**
Recall	X	X	X	**0.99**
Precision	X	X	X	**1**

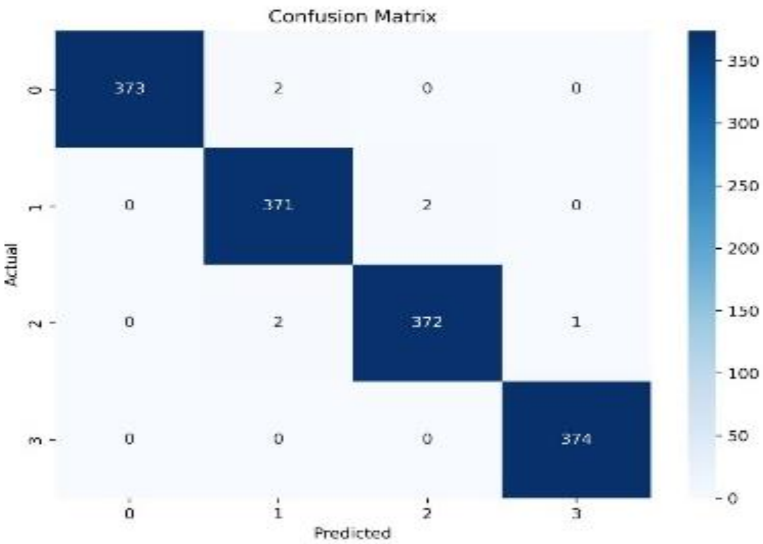

Figure 4.10. Confusion Matrix for multi class CNN- LSTM model.

The model's accuracy exhibits improvement with a progressive increase in the number of epochs, as visually depicted in Figure 4.11 The graphical representation elucidates the discernible enhancement in accuracy and concurrent reduction in loss with respect to the increment in epoch size.

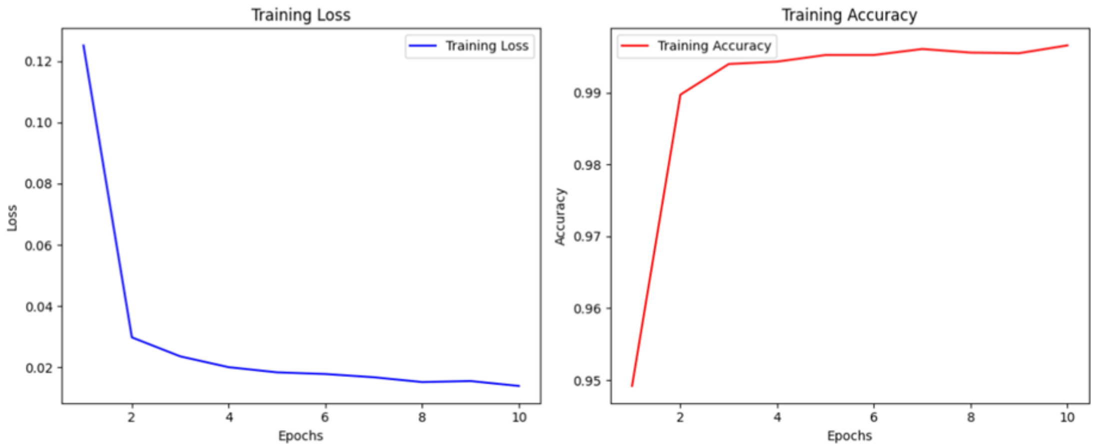

Figure 4.11. Training loss & Training Accuracy for Multi-classification.

4.3. LIMITATIONS OF THE STUDY

In every research endeavor, it is crucial to recognize that no study exists without limitations. Despite our assiduous efforts to mitigate these constraints, they persist, influencing the breadth and resilience of our research. Acknowledging and addressing these limitations becomes imperative, not only to highlight their presence but also to provide context to our findings and illuminate pathways for subsequent investigation and improvement. Various factors and issues have the capacity to influence the outcomes and conclusions of this study. This section will delve into the inherent limitations of our research, their ramifications on the interpretation of our findings, and strategies aimed at their amelioration.

A paramount limitation in this study revolves around the process of data acquisition and compilation. The existing datasets (CAMPSS, predictive maintenance, and motor data) within the domain of predictive maintenance have undergone thorough scrutiny and refinement by researchers and practitioners. In our investigation, the reliance on the Metro PT dataset served as a test case for implementing our proposed hybrid model. Nevertheless, the dataset posed challenges, particularly in the absence of labels, making label extraction a complex and challenging task.

Another notable limitation stems from the heterogeneous nature of the data, marked by the coexistence of both digital and analog signals. Managing this diversity, which encompasses both continuous and intermittent signals, introduces complexities that necessitate careful navigation. Additionally, our research grapples with limitations associated with the selection of hyperparameters. Decisions pertaining to the number of layers, dropout values, and batch sizes—crucial for optimizing the model's performance and accuracy—present challenges in their own right. Resource constraints, encompassing both financial and temporal aspects, have imposed restrictions on the depth and breadth of our research. These limitations have inevitably influenced the extent to which we could delve into the subject matter.

4.4. FUTURE RESEARCH DIRECTION

Potential avenues for future research are delineated in this section, presenting captivating opportunities to enrich our knowledge base and address the limitations inherent in this study. These directions possess the potential to engender more comprehensive insights, thereby contributing substantively to the continual progression of this field. The promising results demonstrated by our proposed framework are underscored, as evidenced by the attainment of an approximate 92% accuracy rate for the binary classifier and a notable 99% accuracy rate for multiple classifications. These achievements were realized through the application of hybrid deep learning algorithms on the MetroPT dataset. Therefore, prospects on the horizon suggest forthcoming opportunities to augment the accuracy of the binary classifier.

An additional promising avenue lies in the anticipation of component failures, constituting a pivotal tool to empower decision-makers in the proactive detection and mitigation of malfunctioning components, thereby minimizing downtime. This proactive approach not only mitigates material and time-related losses but also facilitates the implementation of more streamlined and efficient maintenance practices.

Furthermore, opportunities abound for the refinement of this study and the provision of valuable assistance to researchers through the prediction of the remaining useful life of these components. This approach harbors considerable potential, particularly within the domain of metro vehicles, as it stands to augment the overall reliability of the metro system, enhance passenger safety, elevate the quality of the passenger experience, and preemptively address potential safety concerns before they escalate into severe issues. The capacity to forecast when maintenance is requisite and make judicious decisions regarding asset repair or replacement additionally contributes to the reduction of disruptions for passengers. The incorporation of machine learning and deep learning techniques holds the promise of unveiling deeper insights into intricate data patterns and predictive modeling, thereby ushering in novel horizons for research in this domain.

PART 5

CONCLUSION

The research initiative aims to predict equipment failures for the proactive implementation of maintenance strategies, leveraging the MetroPT dataset. This dataset encompasses three distinct signal types, namely digital, analog, and GPS, sourced from sensors strategically positioned within the air production unit. In an imperative initial phase, the dataset underwent meticulous scrutiny and preprocessing. Deliberate exclusions were made for GPS signals and signals manifesting fixed patterns. Following this, two distinct labels were introduced to address the unclassified nature of the data, aligning classification criteria with the failure cases documented by the company. The first label corresponds to binary classification (0, 1), while the second label accommodates multi-class classification (1, 2, 3, 4).

To prepare the data for model input, a diverse set of techniques was deployed. Under-sampling was employed to rectify imbalances in the dataset, normalization methods were applied to ensure uniformity for continuous signals, and a resampling technique utilizing stacking was executed prior to data integration. Significantly, the research introduced a meticulously crafted hybrid model that harnesses the strengths of deep learning methodologies, specifically integrating a CNN and a LSTM network. This hybrid model was purposefully designed to forecast component failures within the air production unit situated atop the metro system.

The outcomes derived from this study are notably promising. The proposed hybrid model has exhibited a commendable level of accuracy, attaining 92% accuracy for binary classification and an impressive 99% accuracy for multi-class classification. These results unequivocally substantiate the efficacy of the hybrid model in the domain of predictive maintenance, surpassing the performance benchmarks established by antecedent research.

In summary, this research effectively showcased the potential of deep learning techniques in the sphere of predictive maintenance, utilizing the MetroPT dataset. The hybrid model, amalgamating components from CNN and LSTM, demonstrated exceptional accuracy in predicting faults within the metro's air production unit. These findings provide invaluable insights into the practical applications of predictive maintenance, emphasizing the paramount significance of proactive strategies in mitigating operational disruptions and extending equipment longevity. The contributions of this research are poised to catalyze significant advancements in maintenance practices, ultimately establishing a benchmark for both future research endeavors and real-world applications.

REFERENCES

[1] F. J. e. al., "Petroblock: A Blockchain-Based Payment Mechanism for fueling Smart Vehicles," vol. Appl. Sci., vol. 11, no. 7p. 3055., 2021.

[2] R. K. e. al., "Security and Privacy in Connected Vehicle Cyber Physical system Using Zero Knowledge Succinct Non Interactive Argument Of Knowledge Over Blockchain," Appl. Sci., vol. 13, no. 3,, p. 1959, 2023.

[3] C. F. Resende, D.; Oliveira, J.; Franco, B.; Moreira, W.; Oliveira-Jr, A.; Cavaleiro, A.; Carvalho, R. . , "TIP4.0: Industrial Internet of Things Platform for Predictive Maintenance," Sensors 2021, 21, 4676. [CrossRef] [PubMed], 2021.

[4] M. Madhiarasan, "Design and development of IoT based solar powered versatile moving robot for military application," Int. J. Syst. Assur. Eng. Manag, vol. 437–450. [CrossRef], 2021, 12.

[5] P. F.-C. Fraga-Lamas, T.M.; Castedo, L. , "Towards the Internet of Smart Trains: A Review on Industrial IoTConnected Railways," Sensors 2017, 17, 1457, vol. [CrossRef] [PubMed].

[6] Y. B. A. Zikria, R.; Afzal, M.K.; Kim, S.W. , "Next-Generation Internet of Things (IoT): Opportunities, Challenges, and Solutions Sensors," 1174. [CrossRef], 2021, 21.

[7] M. V. Pech, J.; Bednáˇr, "Predictive Maintenance and Intelligent Sensors in Smart Factory: Review.," Sensors 2021, 21, 1470.[CrossRef].

[8] S. S. A. . Sheuly, M.U., "Machine-Learning-Based Digital Twin in Manufacturing: A Bibliometric Analysis and Evolutionary Overview.," Begum, S Appl. Sci. 2022, 12, 6512. [CrossRef].

[9] I. Z. Niyonambaza, M., "Predictive Maintenance (PdM) Structure Using Internet of Things (IoT) for Mechanical Equipment Used into Hospitals in Rwanda," Uwitonze, A. Future Internet 2020, 12, 224. [CrossRef].

[10] S. H. B. Dolatabadi, "Systematic Literature Review Predictive Maintenance Solutions for SMEs from the Last Decade," Machines 2021, 9, 191. [CrossRef].

[11] F. R. Calabrese, A.; Botti, L.; Mora, C.; Galizia, F.G. , "Unsupervised Fault Detection and Prediction of Remaining Useful Life for Online Prognostic Health Management of Mechanical Systems," Appl. Sci. 2020, 10, 4120. [CrossRef].

[12] I. Y. Ullah, F.; Khan, R.; Liu, L.; Yang, H.; Gao, B.; Sun, K. , "Predictive

Maintenance of Power Substation Equipment by Infrared Thermography Using a Machine-Learning Approach," Energies 2017, 10, 1987.

[13] L. Z. Wang, C.; Zhu, J.; Xu, F., "Fault Diagnosis of Motor Vibration Signals by Fusion of Spatiotemporal Features," Machines 2022, 10, 246. [CrossRef].

[14] M. H. M. Abidi, M.K.; Alkhalefah, H. , "Predictive Maintenance Planning for Industry 4.0 Using Machine Learning for Sustainable Manufacturing.," Sustainability 2022, 14, 3387. [CrossRef].

[15] Z. M. A. N. Çınar, A.; Zeeshan, Q.; Korhan, O.; Asmael, M.; Safaei, B. , "Machine Learning in Predictive Maintenance towards Sustainable Smart Manufacturing in Industry 4.0.," Sustainability 2020, 12, 8211. [CrossRef].

[16] A. M. A. Shrestha, "Review of deep learning algorithms and architectures," IEEE Access, vol. 7, pp. 53040-53065, 2019.

[17] X. K. X. K. Dang, Truong, H.N., Nguyen, V.C. and Pham, T.D.A., "Applying convolutional neural networks for limited-memory application," TELKOMNIKA Telecommunication, Computing, Electronics and Control, vol. vol. 19, no. 1, pp. 244-251.

[18] BrunoVeloso 1, 3 ✉, Rita P. Ribeiro2,4, JoãoGama2,3 & Pedro Mota Pereira2, "The MetroPT dataset for predictive maintenance," Data Descriptor, (2022).

[19] H. K. e. al., "Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0: Securing the Future of German Manufacturing Industry," Final Report of the Industrie 4.0 Working Group. Forschungsunion, 2013. URL: https://books.google.it/books?id=AsfOoAEACAAJ., 2013.

[20] R. M. D. e. al., "Business models for the Internet of Things," In:International Journal of Information Management 35.6 (2015), pp. 672–678.

[21] Q. Q. a. F. Tao., "Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison," In: IEEE Access 6 (2018),pp. 3585–3593.

[22] X. Xu.. "From cloud computing to cloud manufacturing," In: Robotics and computer-integrated manufacturing 28.1 (2012), pp. 75–86., 2012.

[23] N. Omri., "Knowledge extraction from SME data for the implementation of PHM process," PhD thesis. Université Bourgogne Franche-Comté, 2021.

[24] M. Jalayer, Orsenigo, C., & Vercellis, C. , "Fault Detection and Diagnosis for Rotating Machinery: A Model Based on Convolutional LSTM," Fast Fourier and Continuous Wavelet Transforms. Computers in Industry, 125, 103378, (2020).

[25] G. A. Susto, Schirru, A., Pampuri, S., McLoone, S., & Beghi, A. , "Machine learning for predictive maintenance: A multiple classifier approach.," IEEE Transactions on Industrial Informatics, 11(3), 812-820, (2014). .

[26] H. M. Hasemian, "State-of-the-art predictive maintenance techniques," IEEE Transactions on Instrumentation and measurement, 60, 1, 226-236,

[27] a. L. P. A. Van Horenbeek, "A dynamic predictive maintenance policy for complex multi-component systems," Reliability Engineering & System Safety, vol. 120, pp. 39-50, , 2013.

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

[29] K. H. L. T. Abbasi, and K. S. Yam, "Predictive maintenance of oil and gas equipment using recurrent neural network," Iop conference series: Materials science and engineering, vol. 495, no. 1, 2019, Art. no. 012067.

[30] M. Schwabacher, "A survey of data-driven prognostics," in Proceedings of the AIAA Infotech@Aerospace Conference: American Institute for Aeronautics and Astronautics, Inc., Reston, VA, 2005., 2005.

[31] Y. Peng, Dong, M., and Zuo, M. J., "Current status of machine prognostics in condition based maintenance: a review," The International Journal of Technology, vol. 50, pp. 297-313, 2010., 2010.

[32] M. a. G. Schwabacher, K., "A Survey of Artificial Intelligence for Prognostics " AAAI Fall Symposium: Artificial Intelligence for Prognostics., 2007. .

[33] a. C. I.-E. H. Chaoui, "State of Charge and State of Health Estimation for Lithium Batteries using Recurrent Neural Networks," IEEE Transactions on Vehicular Technology, vol. 66, no. 10, pp. 8773-8783, October. 2017.

[34] D. Y. Weiting Zhang , Member, IEEE, and Hongchao Wang , Member, IEEE, "Data-Driven Methods for Predictive Maintenance of Industrial Equipment: A Survey," IEEE SYSTEMS JOURNAL, VOL. 13, NO. 3, SEPTEMBER 2019, 2019.

[35] C. I.-E. H. Chaoui, and H. Gualous, "Aging Prediction and State of Charge Estimation of a LiFePO4 Battery using Input Time-Delayed Neural Networks,," Electric Power Systems Research, Elsevier, vol. 146, pp. 189-197, May. 2017., 2017.

[36] K. R. a. G. Methaq, "A hybrid of CNN and LSTM methods for securing web application against cross-site scripting attack," Indonesian Journal of Electrical Engineering and Computer Science (IJEECS), vol. 21, pp. 1022-1029, 2021.

[37] A. M. A. Shrestha, "Review of deep learning algorithms and architectures," IEEE Access, vol. 7, pp. 53040-53065, 2019, .

[38] X. Z. Y. Ran, P. Lin, Y. Wen, R. Deng, "A survey of predictive maintenance: Systems, purposes and approaches," arXiv preprint arXiv:1912.07383. , 2019.

[39] S. Ning, Sun, J., Liu, C. and Yi, Y., "Applications of deep learning in big data analytics for aircraft complex system anomaly detection," Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability, Vol. 235 No. 5, pp. 923-940, , (2021).

[40] L. Basora, Bry, P., Olive, X. and Freeman, F., "Aircraft fleet health monitoring with anomaly detection techniques," Aerospace, Vol. 8 No. 4, pp. 1-33, (2021).

[41] K. Fathi, van de Venn, H.W. and Honegger, M. , "Predictive maintenance: an autoencoder anomaly-based approach for a 3 DoF delta robot," Sensors, Vol. 21, pp. 1-23,, (2021).

[42] C. M. A. Roelofs, Lutz, M.A., Faulstich, S. and Vogt, S., "Autoencoder-based anomaly root cause analysis for wind turbines," vol. Energy and AI, Vol. 4, pp. 1-9, (2021).

[43] !!! INVALID CITATION !!! .

[44] M. Gribbestad, Hassan, M.U., Hameed, I.A. and Sundli, K., "Health monitoring of air compressors using reconstruction-based deep learning for anomaly detection with increased transparency," vol. Entropy, Vol. 23 No. 1, (2021).

[45] M. Beretta, C ardenas, J.J., Koch, C. and Cusid o, J. , "Wind fleet generator fault detection via SCADA alarms and auto encoders," vol. Applied Sciences, Vol. 10, p. 23,, (2020).

[46] L. Xiao, Liu, Z., Zhang, Y., Zheng, Y. and Cheng, C. , "Degradation assessment of bearings with trend-reconstruct-based features selection and gated recurrent unit network," vol. Measurement,Vol. 165., (2020).

[47] G. Manco, Ritacco, E., Rullo, P., Gallucci, L., Astill, W., Kimber, D. and Antonelli, M. , "Fault detection and explanation through big data analysis on sensor streams," Expert Systems with Applications, vol. Vol. 87, pp. 141-156, (2017).

[48] b. Yuchen Jiang a, Pengwen Dai c, Pengcheng Fang d, Ray Y. Zhong e, Xiaoli Zhao f, Xiaochun Cao c, "A2-LSTM for predictive maintenance of industrial equipment based on machine learing," Computers & Industrial Engineering 172 (2022) 108560, 2022.

[49] M. Huang, Liu, Z. and Tao, Y. , "Mechanical fault diagnosis and prediction in iot based on multi source sensing data fusion," Simulation Modelling Practice and Theory, vol. Vol. 102, pp. 1-19, (2020).

[50] E. V. Burnaev, "Time-series classification for industrial applications: road surface damage detection use case," Journal of Communications Technology and Electronics, vol. Vol. 65 No. 12, pp. 1491-1498, (2020).

[51] a. D. A. J. S. Rahhal, "IOT Based Predictive Maintenance Using LSTM RNN Estimator," International Conference on Electrical, Communication, and Computer Engineering (ICECCE),, vol. pp. 1-5, 2020.

[52] G. S. X. Bampoula, N. Nikolakis, K. Alexopoulos., "A Deep Learning Model for Predictive Maintenance in Cyber-Physical Production Systems Using LSTM Autoencoders," Sensors, vol. 21, no. 3, , Art. 972, 2021.

[53] K. Z. a. H. Shao., "Intelligent fault diagnosis of rolling bearing using adaptive deep gated recurrent unit," vol. Neural Process. Lett., vol. 51, no. 2,pp. 1165–1184, Oct. 2020.

[54] Q. G. e. al., "Intelligent fault diagnosis method based on full 1-D convolutional generative adversarial network,," IEEE Trans. Ind. Inform.,, vol. vol. 16, no. 3, pp. 2044–2053, Mar. 2020.

[55] Y. LECUN, BENGIO, Y. & HINTON, G., "Deep learning.," Nature, 521, 436-444., 2015.

[56] S. XIE, GIRSHICK, R., DOLLÁR, P., TU, Z. & HE, K., "Aggregated residual transformations for deep neural networks," Proceedings of the IEEE conference on computer vision and pattern recognition, vol. 1492-1500., 2017.

[57] K. Z. SIMONYAN, A. , "Very deep convolutional networks for large-scale image recognition.," arXiv preprint arXiv:1409.1556., 2014.

[58] I. GOODFELLOW, BENGIO, Y. & COURVILLE, A., "Deep learning," MIT press, 2016.

[59] V. S. D. Lee, R. Cruz, C. Yetman, and S. Diego, "Convolutional Neural Net and Bearing Fault Analysis," p. 7.

[60] S. H. a. J. Schmidhuber, "Long Short-Term Memory " Neural Computation, vol. vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[61] H. A.-K. Ahmed Nasser, "A hybrid of convolutional neural network and long short-term memory network approach to predictive maintenance," International Journal of Electrical and Computer Engineering (IJECE), vol. Vol. 12, No. 1, pp. 721~730, February 2022.

[62] M. Kevin Shen Hoong Ong , IEEE, Wenbo Wang , Member, IEEE, Dusit Niyato , Fellow, IEEE, "Deep-Reinforcement-Learning-Based Predictive Maintenance Model for Effective Resource Management in Industrial IoT," IEEE INTERNET OF THINGS JOURNAL, vol. VOL. 9, NO. 7, APRIL 1, 2022.

[63] B. Zhang, Zhang, S. and Li, W.), "Bearing performance degradation assessment using long short-term memory recurrent network," Computers in Industry, vol. Vol. 106, pp. 14-29, (2019).

[64] L. Xiao, Liu, Z., Zhang, Y., Zheng, Y. and Cheng, C. , "Degradation assessment of bearings with trend-reconstruct-based features selection and gated recurrent unit network," Measurement,Vol. 165, (2020).

[65] A. Santolamazza, Dadi, D. and Introna, V, "A data-mining approach for wind turbine fault detection based on SCADA data analysis using artificial neural networks," Energies, Vol. 14, p. 7., (2021).

[66] L. Xiao, Liu, Z., Zhang, Y., Zheng, Y. and Cheng, C, "Degradation assessment of bearings with trend-reconstruct-based features selection and gated recurrent unit network," Measurement, Vol. 165, (2020).

[67] P. Kumar, & Hati, A.S. , "Deep Convolutional Neural Network Based on Adaptive Gradient Optimizer for Fault Detection in SCIM," ISA Transactions. , (2020).

[68] M. Azamfar, Singh, J., Bravo-Imaz, I., & Lee, J. . . , . "Multisensor Data Fusion for Gearbox Fault Diagnosis Using 2-D Convolutional Neural Network and Motor Current Signature Analysis," Mechanical Systems and Signal Processing, vol. 144, 106861, (2020).

[69] M. Skowron, "Application of Deep Learning Neural Networks for the Diagnosis of Electrical Damage to the Induction Motor Using the Axial Flux," Bulletin of the Polish Academy of Sciences: Technical Sciences, vol. 68(5),1031-1038. , (2020).

[70] P. Kumar, & Hati, A.S., "Convolutional Neural Network with Batch Normalisation for Fault Detection in Squirrel Cage Induction Motor.," IET Electric Power Applications, vol. 15(1), 39-50, (2021).

[71] M. A. Khan, Kim, Y.H., & Choo, J. .. , "Intelligent Fault Detection Using Raw Vibration Signals via Dilated Convolutional Neural Networks," The Journal of Supercomputing, 76, 8086-8100., (2018).

[72] M. T. Pham, Kim, J.M., & Kim, C.H., "Deep Learning-Based Bearing Fault Diagnosis Method for Embedded Systems," Sensors, 20(23), 6886, (2020)

[73] R. M. Souza, Nascimento, E.G.S., Miranda, U.A., Silva, W.J.D., & Lepikson, H.A. , "Deep Learning for Diagnosis and Classification of Faults in Industrial Rotating Machinery," Computers & Industrial Engineering, vol. 153,107060., (2020).

[74] A. Shenfield, & Howarth, M. , "A Novel Deep Learning Model for the Detection and Identification of Rolling Element-Bearing Faults," Sensors, 20(18), 5112, (2020).

[75] H. Nakamura, Asano, K., Usuda, S., & Mizuno, Y. . . "A Diagnosis Method of Bearing and Stator Fault in Motor Using Rotating Sound Based on Deep Learning," Energies, 14(5), 1319. , (2021).

[76] S. Plakias, & Boutalis, Y.S. , "Fault Detection and Identification of Rolling Element Bearings with Attentive Dense CNN," Neurocomputing, 405, 208-217, (2020).

[77] Y. L. Karnavas, Plakias, S., Chasiotis, I.D. , "A Multi-Scale Deep Learning Attention-based Feature Method for Rolling Elements Bearing Fault Detection in Industrial Motor Drives.," Proceedings of the 10th International Conference on Circuits and Systems Technologies (MOCAST). Thessaloniki, Greece, (2021b).

[78] I. M. Daniyan, K.; Oyesola, M.; Ramatsetse, B.; Adeodu, A., "Artificial intelligence for predictive maintenance in the railcar learning factories.," Procedia Manuf. 2020, 45, 13–18. [CrossRef].

[79] T. P. S. Carvalho, F.A.A.M.N.; Vita, R.; Francisco, R.D.P.; Basto, J.P.; Alcalá, S.G.S., "A systematic literature review of machine learning methods applied to predictive maintenance.," Comput. Ind. Eng. 2019, 137, 106024. [CrossRef].

[80] S. A.-P. Barja-Martinez, M.; Munné-Collado, Í.; Lloret-Gallego, P.; Bullich-Massagué, E.; Villafafila-Robles, R., "Artificial intelligence techniques for enabling Big Data services in distribution networks: A review. Renew. Sustain. Energy Rev.," 2021, 150, 111459. [CrossRef].

[81] M. F. Nacchia, F.; Lambiase, A.; Bruton, K., "A Systematic Mapping of the Advancing Use of Machine Learning Techniques for Predictive Maintenance in the Manufacturing Sector.," Appl. Sci. 2021, 11, 2546. [CrossRef].

[82] A. G. Frank, Dalenogare, L.S., & Ayala, N.F. , "Industry 4.0 technologies: implementation patterns in manufacturing companies. ," International Journal of Production Economics. 210, 15-26. , (2019). .

[83] J. Lee, Lapira, E., Yang, S., & Kao, A. , "Predictive Manufacturing System - Trends of Next-Generation Production Systems.," IFAC Proceedings Volumes, 46(7), 150-156., (2013). .

[84] F. Tao, Qi, Q., Liu, A., & Kusiak, A. , "Data-driven smart manufacturing. Journal of Manufacturing Systems," vol. 48, 157-169, (2018)

[85] R. Zhao, Yan, R., Chen, Z., Mao, K., Wang, P., & Gao, R.X., "Deep learning and its applications to machine health monitoring. Mechanical Systems and Signal Processing," 115, 213-237., (2019).

[86] W. Ji, & Wang, L. , "Big data analytics based fault prediction for shop floors scheduling," Journal of Manufacturing Systems, 43, 187-194, (2017).

[87] J. H. Lee, Pack, J.H., & Lee, I.S. , "Fault Diagnosis of Induction Motor Using Convolutional Neural Network," Applied Sciences, 9(15), 2950., (2019).

[88] J. R. Ruiz-Sarmiento, Monroy, J., Moreno, F.A., Galindo, C., Bonelo, J.M., & Gonzalez-Jimenez, J. , " A predictive model for the maintenance of industrial machinery in the context of industry 4.0. Engineering Applications of Artificial Intelligence,," 87, 103289., (2020).

[89] J. Wang, Fu, P., Zhang, L., Gao, R.X., & Zhao, R., "Multi-Level Information Fusion for Induction Motor Fault Diagnosis. ," IEEE/ASME Transactions on Mechatronics, 24(5), 2139-2150, (2019).

[90] Y. B. Yann LeCun, and Geoffrey Hinton. , "“Deep learning”," In: na ture 521.7553 , pp. 436–444., (2015).

[91] Y. B. Ian Goodfellow, and Aaron Courville.. , "Deep learning," MIT press, , 2016.

[92] A. K. e. al., "“A survey of the recent architectures of deep convolutional neural networks”," In: Artificial intelligence review 53.8 pp. 5455–5516, (2020).

[93] W. Sun, Zhao, R., Yan, R., Shao, S., & Chen, X., "Convolutional discriminative feature learning for induction motor fault diagnosis," IEEE Transactions on Industrial Informatics, 13(3), 1350-1359, (2017).

[94] O. S. e. al., "“Deep learning models for predictive maintenance: a survey, comparison, challenges and prospects”," In: Applied Intelligence pp. 1–31. , (2022).

[95] A. M. Dominik Scherer, and Sven Behnke. , "“Evaluation of pooling operations in convolutional architectures for object recognition”," In: Interna tional conference on artificial neural networks. Springer. 2010, pp. 92–101 2010.

[96] Y. Z. L. Dong, C. Wen, and H. Wu, "“Camera anomaly detection based on morphological analysis and deep learning,," IEEE International Conference on Digital Signal Processing (DSP). Beijing, China: IEEE, Oct. 2016, pp. 266–270. [Online]. , in 2016

[97] V. S. D. Lee, R. Cruz, C. Yetman, and S. Diego, "Convolutional Neural Net and Bearing Fault Analysis," p. 7.

[98] K. R. S. Zheng, A. Farahat, and C. Gupta, "Long short-term memory network for remaining useful life estimation," IEEE international conference on prognostics and health management (ICPHM), 2017, pp. 88-95, 2017

[99] S. H. a. J. Schmidhuber, "“Long Short-Term Memory "," Neural Computation,vol. 9, no. 8, pp. 1735–1780 Nov.1997.

[100] K. L. C. Hwang, and H. Jung, "Improving data quality using a deep learning network," Indonesian Journal of Electrical Engineering and Computer Science (IJEECS), vol. 20, no. 1, pp. 306-312.

[101] H. A.-K. Ahmed Nasser, "A hybrid of convolutional neural network and long short-term memory network approach to predictive maintenance," International Journal of Electrical and Computer Engineering (IJECE), February 2022.

[102] D. Xiao, Huang, Y., Zhang, X., Shi, H., Liu, C., & Li, Y. (2018). . "Fault Diagnosis of Asynchronous Motors Based on LSTM Neural Network," Prognostics and System Health Management Conference (PHM-Chongqing). Chongqing, China. , (2018).

[103] D. Y. a. H. W. Weiting Zhang, "“Data-driven methods for predictive maintenance of industrial equipment: A survey”," In: IEEE Systems Journal 13.3 (2019), pp. 2213–2227., (2019).

[104] BrunoVeloso 1, 3 ✉, Rita P. Ribeiro2,4, JoãoGama2,3 & Pedro Mota Pereira, "The MetroPT dataset for predictive maintenance," (2022).

[105] R. R. J. G. B. Veloso, and P. Pereira, "MetroPT: A Benchmark dataset for predictive maintenance," Zenodo, Jul. 18, 2022.

[106] J. Gama, Ribeiro, R. P. & Veloso, B., "Data-driven predictive maintenance," IEEE Intelligent Systems 1–2 (2022).

[107] N. e. a. S. Davari, 5739 (2021), "A survey on data-driven predictive maintenance for the railway industry," Sensors 21, 5739 (2021).

[108] M. Barros, Veloso, B., Pereira, P. M., Ribeiro, R. P. & Gama, J. . In IoT and E. Streams for Data-Driven Predictive Maintenance and IoT, and Mobile for Embedded Machine Learning, 61–74 (Springer, 2020), "Failure detection of an air production unit in operational context," In IoT Streams for Data-Driven Predictive Maintenance and IoT, Edge, and Mobile for Embedded Machine Learning, 61–74 (Springer, 2020), 2020.

[109] N. Davari, Veloso, B., Ribeiro, R. P., Pereira, P. M. & Gama, J. , "Predictive maintenance based on anomaly detection using deep learning for air production unit in the railway industry.," In 2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA), 1–10 (IEEE, 2021). 2021.

[110] A. D. Anjir Ahmed Chowdhury1, Khadija Kubra Shahjalal Hoque3 , and Debajyoti Karmaker, "A Comparative Study of Hyperparameter Optimization Techniques for Deep Learning," January 2022.

[111] BrunoVeloso 1, 3 ✉, Rita P. Ribeiro2,4, JoãoGama2,3 & Pedro Mota Pereira2, "The MetroPT dataset for predictive maintenance," 2022.

[112] H. I. A. Ayat Najjar, and Ahmad Hasasneh, "Predictive Maintenance of Urban Metro Vehicles: Classification of Air Production Unit Failures Using Machine Learning," 2023.

[113] H. A.-K. Ahmed Nasser, "A hybrid of convolutional neural network and long short-term memory network approach to predictive maintenance," nternational Journal of Electrical and Computer Engineering (IJECE) ·, February 2022.

RESUME

Shahad Jameel Farhan ALSAID, graduated in 2016 with a bachelor's degree in information engineering in department international network engineering in Al-Nahrain University and enrolled in a master's degree in 2021 in the Department of computer engineering. She worked as a junior engineer in the field of computer engineering in industry Ministry (Iraq) in department of statistics and informatics in planning.