People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

University of Echahid Hamma Lakhdar El Oued

Faculty of Exact Sciences

End-of-Study Dissertation

Academic Master

Field : Computer Science

Specialty : Internet of Things and Network Security

Topic

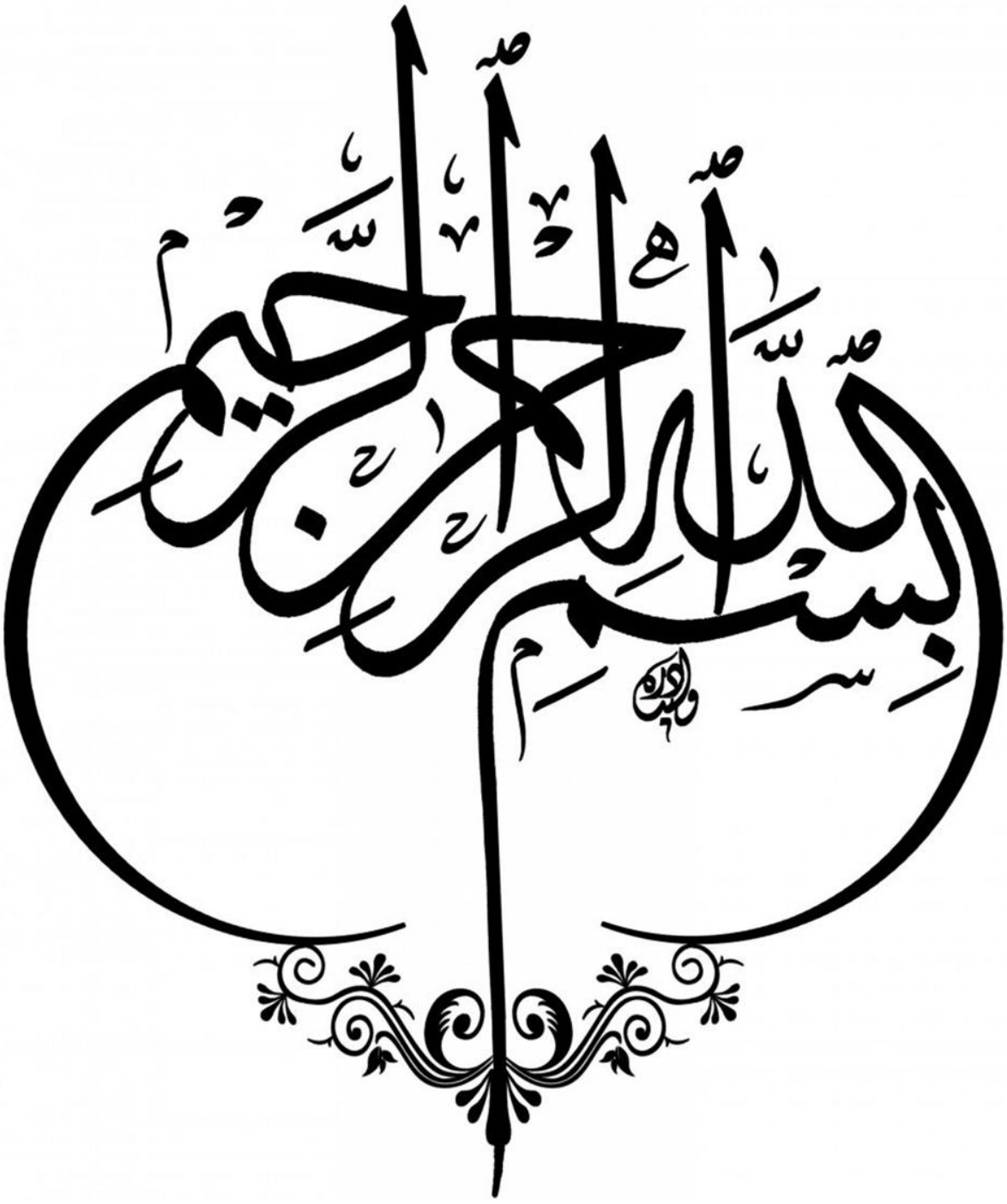
An IoT-Driven Framework for Predictive Maintenance of Induction Motors Using Machine Learning

Presented by : Hafsi Ahmed
Chenguel Mohammed Laid

Publicly defended on :07-06-2026, before the jury composed of :

Maamra Oum El Hana	**Pr**	**Univ. of El-Oued**	**President**
Benbraika mohamed kamel	**Pr**	**Univ. of El-Oued**	**Supervisor**
Outhmani Samir	**MCA**	**Univ. of El-Oued**	**Examiner**

Academic Year : 2025 – 2026.


بسم الله الرحمن الرحيم



الشكر والتقدير



الحمد لله أولاً وآخراً على نعمه التي لا تُحصى، والذي وفقني ومنحني القوة والصبر لإتمام هذا العمل.

أتقدم بخالص الشكر والتقدير إلى الأستاذ المشرف على هذه المذكرة، لما قدمه من توجيهات ونصائح قيمة ومتابعة مستمرة طوال فترة إنجاز هذا العمل.

كما أتوجه بالشكر إلى جميع أساتذة قسم الإعلام الآلي على ما قدموه من علم ومعرفة طيلة مشواري الدراسي.

ولا يفوتني أن أتقدم بجزيل الشكر والعرفان إلى عائلتي الكريمة على دعمهم المتواصل وتشجيعهم الدائم وصبرهم طوال فترة الدراسة.

وفي الأخير، أتوجه بالشكر لكل من ساهم من قريب أو بعيد في إنجاز هذه المذكرة.



الاهداء



أهدي هذا العمل المتواضع إلى والديّ الكريمين،

اللذين كانا سندي ودعمي الدائم، وإلى كل من

غمرني بدعائه وتشجيعه طوال مسيرتي الدراسية.

إلى عائلتي وأصدقائي، الذين كانوا دائماً مصدر تحفيز

ودعم.

إلى أساتذتي الأفاضل الذين ساهموا في تعليمي

وتوجيهي.

وإلى كل من آمن بقدرتي على النجاح

وساعدني في الوصول إلى هذه المرحلة.

Abstract

This work proposes an IoT-driven predictive maintenance framework combined with Machine Learning to monitor induction motors and estimate their Remaining Useful Life (RUL). The methodology integrates a conceptual IoT architecture for data telemetry with ML-driven prognostic modeling. To evaluate the framework under varying complexities, a MATLAB/Simulink simulation was developed to generate progressive Broken Rotor Bar (BRB) fault trajectories, complemented by the real-world XJTU-SY bearing dataset. Following rigorous preprocessing and feature extraction, two regression models were investigated: Random Forest (RF) and Long Short-Term Memory (LSTM) networks. Experimental results demonstrated that while RF offers stable predictions using statistical time-domain features, LSTM excels in capturing sequential temporal degradation trends. The findings demonstrate that coupling simulated fault benchmarks with real-world failure datasets provides a reliable approach for evaluating automated RUL estimation models under industrial noise, establishing a scalable foundation for future edge hardware deployments.

Keywords: Predictive Maintenance, Internet of Things, Remaining Useful Life, Induction Motor, Machine Learning, LSTM, Random Forest.

الملخص

يقترح هذا العمل إطار عمل للصيانة التنبؤية يعتمد على إنترنت الأشياء (IoT) مدمجاً بتقنيات تعلم الآلة لمراقبة المحركات الحثية وتقدير عمرها التشغيلي المتبقي .(RUL) تدمج المنهجية المتبعة بين بنية مفاهيمية لإنترنت الأشياء ومحاكاة تدفق البيانات، وبين نمذجة التنبؤ القائمة على الذكاء الاصطناعي. ولتقييم النظام تحت تعقيدات مختلفة، تم تطوير نموذج محاكاة ببرنامج MATLAB/Simulink لتوليد مسارات تدهور تدريجية لأعطال قضبان الدوار المكسورة(BRB) ، تعززها قاعدة بيانات المحامل الحقيقية .(XJTU–SY)وبعد عمليات معالجة مسبقة واستخلاص صارم للميزات، تم فحص نموذجين للتنبؤ: خوارزمية الغابة العشوائية (Random Forest) وشبكات الذاكرة طويلة قصيرة المدى .(LSTM) وأظهرت النتائج التجريبية أنه بينما توفر الغابة العشوائية تنبؤات مستقرة باستخدام الميزات الإحصائية الزمنية، تفوقت شبكات LSTM في التقاط اتجاهات التدهور الزمني المتسلسل. وتؤكد النتائج أن دمج نماذج الأعطال المحاكاة مع بيانات الفشل الواقعية يوفر نهجاً موثوقاً لتقييم نماذج التقدير الآلي لعمر المعدات تحت تأثير الضوضاء الصناعية، مما يضع أساساً قابلاً للتطوير لتطبيقات الحوسبة الطرفية (Edge Computing) مستقبلاً.

الكلمات المفتاحية: الصيانة التنبؤية، إنترنت الأشياء، العمر التشغيلي المتبقي(RUL) ، المحرك الحثي، تعلم الآلة، LSTM، الغابة العشوائية.

Résumé

Ce travail propose un cadre de maintenance prédictive basé sur l'IoT, combiné à l'Apprentissage Automatique (Machine Learning) pour surveiller les moteurs à induction et estimer leur durée de vie utile restante (RUL). La méthodologie intègre une architecture IoT conceptuelle pour l'émulation des données avec une modélisation pronostique par l'IA. Pour évaluer le système sous différentes complexités, une simulation MATLAB/Simulink a été développée afin de générer des trajectoires de dégradation progressives pour les défauts de barres de rotor cassées (BRB), complétée par la base de données réelle de roulements XJTU-SY. Suite à un prétraitement rigoureux et à l'extraction de descripteurs, deux modèles de régression ont été examinés : Random Forest (RF) et les réseaux Long Short-Term Memory (LSTM). Les résultats expérimentaux ont démontré que, tandis que RF offre des prédictions stables via des descripteurs statistiques temporels, LSTM excelle dans la capture des tendances de dégradation temporelle séquentielle. Ces résultats prouvent que le couplage de repères de défauts simulés avec des données de défaillance réelles fournit une approche fiable pour évaluer les modèles d'estimation automatisée de la RUL sous bruit industriel, établissant ainsi une base évolutive pour de futurs déploiements matériels embarqués (Edge Computing).

Mots-clés: Maintenance Prédictive, Internet des Objets, Durée de Vie Utile Restante (RUL), Moteur à Induction, Apprentissage Automatique, LSTM, Random Forest.

TABLE OF CONTENT

التكر والتقدير I

الاهداء II

ABSTRACT III

الملخص IV

RÉSUMÉ V

GENERAL INTRODUCTION

CHAPTER I :GENERAL OVERVIEW OF IOT-BASED PREDICTIVE MAINTENANCE AND ML

I.1 INTRODUCTION 5

I.2 INDUSTRY 4.0 5

I.2.1 Advantages of Industry 4.0 ..... *6*

I.2.2 Challenges and Limitations of Industry 4.0 ..... *6*

I.2.3 Key Technologies of Industry 4.0 ..... *7*

I.3 MAINTENANCE 8

I.3.1 Importance of Machine Maintenance *8*

I.3.2 Types of Maintenance *9*

I.3.3 Integration of IoT Technologies in Predictive Maintenance (PdM) *12*

I.4 TYPES OF MACHINE FAULTS 13

I.5 CONCLUSION 15

CHAPTER II: FUNDAMENTALS OF IOT AND ML FOR PREDICTIVE MAINTENANCE

II.1 INTRODUCTION 17

II.2 INTERNET OF THINGS (IOT) 17

II.2.1 Definition of IoT *17*

II.2.2 IoT Architecture *17*

II.2.3 IoT in Predictive Maintenance *18*

II.3 MACHINE LEARNING 19

II.3.1 Definition of Machine Learning *19*

II.3.2 Types of Machine Learning *19*

II.4 MACHINE LEARNING ALGORITHMS 22

II.4.1 Logistic Regression *22*

II.4.2 K-Nearest Neighbors (KNN) *23*

II.4.3 Random Forest *24*

II.5 ARTIFICIAL NEURAL NETWORKS (ANN) 24

II.5.1 Structure of Neural Networks *25*

II.5.2 Working Principle of Neural Networks *26*

II.5.3 Long Short-Term Memory (LSTM) *27*

II.6 RELATED WORK 28

II.7 CONCLUSION 30

CHAPTER III: IOT SYSTEM DESIGN AND IMPLEMENTATION

III.1 INTRODUCTION 32

III.2 DEVELOPMENT ENVIRONMENT 32

III.2.1 Google Colab *32*

III.2.2 Visual Studio Code *33*

III.2.3 Python *33*
III.2.4 Pandas *33*
III.2.5 NumPy *34*
III.2.6 Keras *34*
III.2.7 TensorFlow *34*
III.3 IoT System Design 35
III.3.1 IoT System Architecture *35*
III.3.2 System Components *36*
III.3.3 Data Flow Process *37*
III.3.4 System Workflow *38*
III.3.5 System State Diagram *39*
III.4 Simulation Model and Implementation 40
III.4.1 Model Components and Functionality *41*
III.5 Conclusion 42

CHAPTER IV: IMPLEMENTATION AND RESULTS

IV.1 Introduction 44
IV.2 Dataset Description 45
IV.2.1 Simulated BRB Degradation Dataset *45*
IV.2.2 Real Run-to-Failure Dataset: XJTU-SY *46*
IV.3 Data Preprocessing and Feature Extraction 47
IV.3.1 Signal Preparation *47*
IV.3.2 Feature Extraction *47*
IV.3.3 RUL Target Construction *49*
IV.3.4 Data Normalization and Model Input Preparation *50*
IV.3.5 Data Processing Pipeline *50*
IV.4 RUL Prediction Models 50
IV.4.1 Random Forest Regressor *51*
IV.4.2 Long Short-Term Memory Network *51*
IV.4.3 Model Selection Strategy *52*
IV.4.4 Training Configuration and Hyperparameters *52*
IV.5 Evaluation Metrics 53
IV.5.1 Mean Absolute Error (MAE) *53*
IV.5.2 Root Mean Squared Error (RMSE) *53*
IV.5.3 Coefficient of Determination (R^2) *53*
IV.5.4 Evaluation Procedure *54*
IV.6 Results and Discussion 54
IV.6.1 Results on the Simulated BRB Dataset *54*
IV.6.2 Results on the Real XJTU-SY Dataset *57*
IV.6.3 General Discussion *59*
IV.7 Conclusion 61

GENERAL CONCLUSION

REFERENCES **64**

List of Figures

Figure I.1: Evolution of Industrial Revolutions from Industry 1.0 to Industry 4.0 6
Figure II.1: Layered Architecture of the Internet of Things (IoT) [5] 18
Figure II.2: Supervised Learning Model Using Labeled Data 20
Figure II.3: Classification Technique for Fault Detection 20
Figure II.4: Regression Model for Predicting Remaining Useful Life (RUL) 21
Figure II.5: Unsupervised Learning for Pattern Discovery and Anomaly Detection 21
Figure II.6: Logistic Regression Model for Binary Classification 22
Figure II.7: K-Nearest Neighbors (KNN) Algorithm for Similarity-Based Classification 23
Figure II.8: Random Forest Algorithm Based on Multiple Decision Trees 24
Figure II.9: Basic Structure of an Artificial Neural Network 25
Figure II.10: Structure and Training Process of an Artificial Neural Network 26
Figure II.11: Architecture of a Long Short-Term Memory (LSTM) Network 28
Figure III.1: Google Colab Platform Interface 32
Figure III.2: Visual Studio Code Logo 33
Figure III.3: Python Programming Language Logo 33
Figure III.4: Pandas Library Logo 33
Figure III.5: NumPy Library Logo 34
Figure III.6: Keras Library Logo 34
Figure III.7: TensorFlow Framework Logo 34
Figure III.8: Proposed IoT-Based Architecture for Induction Motor Predictive Maintenance 35
Figure III.9: Sequence Diagram of the Proposed IoT-Based Predictive Maintenance System 37
Figure III.10: Workflow of the Proposed IoT System for Predictive Maintenance 38
Figure III.11: State Machine Diagram of the Proposed Predictive Maintenance System 39
Figure III.12: MATLAB/Simulink Model of the Induction Motor for Data Acquisition 41
Figure IV.1: Simulink model of the induction motor system used for BRB degradation simulation. 45
Figure IV.2: Architecture of the LSTM model used for sequence-based RUL prediction. 51
Figure IV.3: Actual and predicted normalized RUL for the simulated BRB dataset using Random Forest. 55
Figure IV.4: Actual and predicted normalized RUL for the simulated BRB dataset using LSTM. 55
Figure IV.5: Training and validation loss curve of the BRB LSTM model. 56
Figure IV.6: Actual and predicted RUL for the XJTU-SY Condition 1 baseline experiment. 57
Figure IV.7: Actual, raw predicted, and smoothed predicted RUL for Bearing2_2 under Condition 2. 57
Figure IV.8: Actual and predicted normalized RUL for Bearing3_1 using LSTM. 58
Figure IV.9: Actual and predicted normalized RUL for Bearing3_5 using LSTM. 58

List of Tables

Table II.1: Comparison of Related Works 29
Table III.1: Induction Motor Parameters 40
Table IV.1: Summary of the simulated BRB degradation dataset 46
Table IV.2: Selected XJTU-SY bearings used in the real-data RUL experiments 46
Table IV.3: Extracted features used for RUL prediction 47
Table IV.4: Summary of adopted prediction models 52
Table IV.5: Random Forest hyperparameters used in this study 52
Table IV.6: Training configuration and hyperparameters 53
Table IV.7: Evaluation metrics used for RUL prediction models 54
Table IV.8: Summary of BRB simulation model performance 56
Table IV.9: Summary of real-data model performance on the XJTU-SY dataset 59
Table IV.10: Overall summary of the main experimental findings 60

General Introduction

The rapid evolution of industrial technologies and the emergence of Industry 4.0 have encouraged industries to adopt smarter and more efficient solutions for managing production systems. In modern industrial environments, maintenance has become an essential factor for ensuring system reliability, reducing unexpected failures, and minimizing production downtime. Traditional maintenance strategies such as corrective and preventive maintenance are often limited when dealing with complex industrial systems and continuously changing operating conditions [1].

In this context, Predictive Maintenance (PdM) has gained significant attention as an advanced maintenance approach based on continuous monitoring and early fault prediction. This strategy combines technologies such as the Internet of Things (IoT), smart sensors, and Machine Learning (ML) algorithms to collect and analyze operational data from industrial equipment in real time [2].

One of the main challenges faced by industrial companies is the early detection of machine degradation before critical failures occur. Conventional maintenance methods based on periodic inspections or fixed maintenance schedules may lead to unnecessary maintenance operations or unexpected breakdowns. At the same time, modern industrial machines generate large amounts of operational data that can be used to improve maintenance decisions.

To address these challenges, this work proposes an IoT-based predictive maintenance system combined with Machine Learning techniques for monitoring industrial equipment and predicting potential failures. The proposed approach relies on real-time data acquisition, signal processing, and intelligent analysis to estimate the Remaining Useful Life (RUL) of industrial components and support proactive maintenance strategies.

The main objective of this research is to design and develop a predictive maintenance framework capable of improving fault detection and reducing maintenance costs. This study also investigates IoT communication technologies, data processing methods, and Machine Learning models used for predictive maintenance applications.

This thesis is organized into four main chapters:

- **Chapter I:** General Overview of IoT-Based Predictive Maintenance. This chapter introduces the concepts of Industry 4.0 and predictive maintenance, and explains the role of IoT and Machine Learning in improving industrial system reliability.
- **Chapter II:** Fundamentals of IoT and Machine Learning Technologies. This chapter presents the theoretical background of IoT communication systems, machine learning algorithms, and neural networks used in predictive maintenance applications.
- **Chapter III:** IoT System Design and Implementation. This chapter describes the proposed system architecture, the data flow process, and the MATLAB/Simulink model developed to simulate induction motor degradation and generate operational data.
- **Chapter IV:** Experimental Results and RUL Evaluation. This chapter focuses on data preprocessing, feature extraction, model training, and Remaining Useful Life (RUL) prediction. It also presents and discusses the results obtained from both simulated and real-world datasets.

This work demonstrates how the integration of IoT technologies and Machine Learning techniques can contribute to the development of intelligent predictive maintenance systems capable of improving equipment reliability and supporting industrial decision-making.

Chapter I:

General Overview of IoT-Based Predictive Maintenance and ML

I.1 Introduction

The development of modern industrial technologies has changed the way industries operate and maintain their equipment. With the emergence of Industry 4.0, many industrial sectors have started adopting technologies such as the Internet of Things (IoT), Big Data, and Artificial Intelligence in order to improve system performance, reliability, and production efficiency.

Industrial maintenance is an essential part of modern production systems because machine failures can cause production interruptions and high maintenance costs. For this reason, different maintenance strategies have been introduced over the years, including reactive, corrective, preventive, and predictive maintenance.

This chapter presents the main concepts related to Industry 4.0 and industrial maintenance. It also introduces the different maintenance strategies, the common types of machine faults, and the role of IoT technologies in predictive maintenance applications.

I.2 Industry 4.0

The term Industry 4.0 was introduced during the Hannover Fair in 2011 and marked the beginning of a new phase in industrial development [3]. Initially associated with the modernization of German industry, the concept rapidly expanded and became a global industrial trend.

Industry 4.0 represents the fourth industrial revolution, characterized by the integration of digital technologies into industrial systems. While the previous industrial revolutions focused on mechanization, electrification, and automation, this new phase is mainly based on connectivity, intelligent systems, and data exchange.

One of the main characteristics of Industry 4.0 is the integration of cyber-physical systems, the Internet of Things (IoT), and digital technologies into manufacturing processes. These technologies contribute to improving production efficiency, monitoring industrial equipment, and supporting real-time decision-making.

An important concept related to Industry 4.0 is the smart factory. Smart factories use technologies such as IoT, artificial intelligence, and automation to create connected and flexible production environments. This allows industries to improve operational efficiency and respond more effectively to changing industrial requirements [3].

The evolution of industrial revolutions from Industry 1.0 to Industry 4.0 is illustrated in **Figure I.1**.

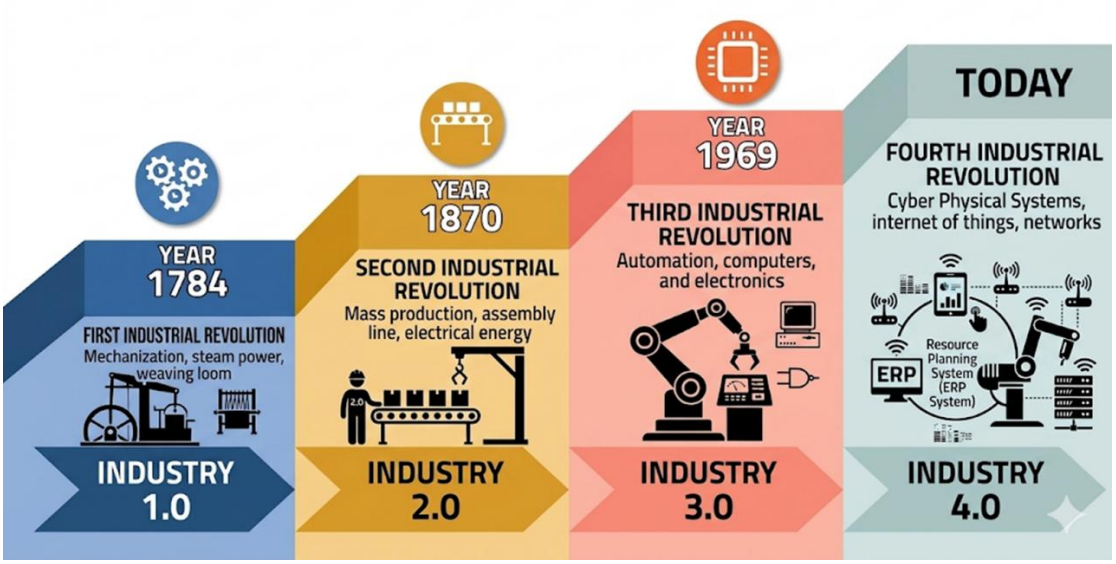


Figure I.1: Evolution of Industrial Revolutions from Industry 1.0 to Industry 4.0

I.2.1 Advantages of Industry 4.0

Industry 4.0 brings substantial benefits to modern manufacturing through the adoption of digital and cyber-physical technologies. The integration of advanced systems such as the Internet of Things (IoT), automation, and data analytics enables companies to enhance productivity and optimize operational efficiency by reducing errors and downtime, and by improving the speed of decision-making through real-time information access. Furthermore, these technologies allow for greater flexibility in production processes and customization of products to meet diverse customer needs. In addition to economic benefits, smart manufacturing improves worker safety by automating hazardous tasks and provides new opportunities for sustainable resource management, which can reduce waste and environmental impact. These advantages contribute to increased competitiveness and responsiveness within rapidly changing global markets [4].

I.2.2 Challenges and Limitations of Industry 4.0

Despite its many potential benefits, Industry 4.0 also presents significant challenges that manufacturers must address. One major barrier is the high initial cost of implementing advanced technologies, including hardware, software, and training, which can be prohibitive for smaller companies. Additionally, the transition to smart manufacturing requires a workforce with specialized skills, creating a need for extensive reskilling and education. The increased interconnectedness of systems also raises cybersecurity concerns, as digital networks can be vulnerable to attacks that could disrupt operations or compromise sensitive data. Integration complexities, such as ensuring interoperability between new technologies and legacy systems,

further complicate adoption. Therefore, while Industry 4.0 holds promise for industrial transformation, careful strategic planning and investment in human and technological resources are essential to overcome these challenges [4].

I.2.3 Key Technologies of Industry 4.0

I.2.3.1 Internet of Things (IoT)

The Internet of Things (IoT) refers to a network of connected devices and sensors capable of collecting, exchanging, and processing data through communication networks. IoT combines different information and communication technologies (ICT) to allow physical objects to interact and communicate over the Internet.

The term IoT was introduced in 1999 by Kevin Ashton [5]. At that time, the concept mainly focused on the use of RFID technology for tracking products automatically in supply chains. With the evolution of communication technologies and embedded systems, IoT has expanded into many application areas, including industrial systems, healthcare, transportation, and smart environments.

An IoT system generally consists of physical devices equipped with sensors and communication modules capable of collecting real-time data and transmitting them through the network [6]. In industrial environments, IoT technologies are widely used for equipment monitoring, process automation, and predictive maintenance applications.

The integration of IoT with Cyber-Physical Systems (CPS) enables industries to improve operational efficiency, optimize maintenance operations, and support data-driven decision-making through continuous monitoring and analysis of industrial processes [6].

I.2.3.2 Big Data

The term *Big Data* refers to extremely large datasets, both structured and unstructured, generated by organizations. Big Data analytics enables companies to extract actionable insights from these vast amounts of information, thereby optimizing decision-making processes and improving business strategies. In the context of Industry 4.0, Big Data complements IoT by allowing real-time analysis of industrial processes, enhancing productivity, quality, and operational reliability [7].

I.2.3.3 Artificial Intelligence (AI) and Machine Learning (ML)

Artificial Intelligence (AI) and Machine Learning (ML) are essential technologies in modern industrial systems, particularly within the context of Industry 4.0 and predictive maintenance. These technologies enable the analysis of large volumes of data generated from various sources such as sensors, machines, and industrial monitoring systems. By processing

this data, AI algorithms can identify patterns, optimize production processes, and predict potential equipment failures before they occur [8].

Machine Learning focuses on developing models that learn from historical and real-time data in order to make predictions or detect anomalies in equipment behavior. Through statistical analysis and data mining techniques, ML algorithms can recognize patterns that indicate early signs of malfunction, allowing maintenance teams to take preventive actions and reduce unexpected downtime [8].

I.3 Maintenance

Maintenance refers to all technical, administrative, and management activities performed during the operating life of equipment in order to maintain its normal operation or restore it after a malfunction. In industrial systems, maintenance is important for improving reliability, reducing downtime, and ensuring continuous production [9].

Several concepts are closely related to maintenance and fault analysis:

Symptom: an observable change or abnormal behavior in a measurable parameter that may indicate the presence of a problem in the system.

Fault: an abnormal condition or defect affecting a component or system, which may eventually lead to failure.

Failure: the inability of a component or system to perform its intended function.

Partial failure: a situation where the system continues to operate but with degraded performance or reduced efficiency.

Complete failure: the total loss of the system's ability to perform its required function [9].

I.3.1 Importance of Machine Maintenance

A - Prevention of Failures and Breakdowns

Regular maintenance of machines is essential to reduce the risk of failures. By ensuring that all components remain in good working condition, potential faults can be detected and addressed before they occur during production. This approach helps prevent unexpected downtime, costly component replacements, high repair expenses, and decreases in productivity.

B - Efficient Operation of Machines

Proper maintenance ensures that machines operate efficiently and reliably. Well-maintained equipment performs its functions under optimal conditions, which improves overall system performance, reduces energy consumption, and enhances production quality.

C - Extension of Equipment Lifespan

Regular maintenance helps extend the lifespan of machines and industrial equipment. By replacing worn components and ensuring that systems operate under proper conditions, organizations can significantly increase the durability of their assets and delay the need for expensive equipment replacements.

D - Improved Safety

Maintenance plays an important role in ensuring the safety of operators and industrial environments. Faulty or poorly maintained machines may lead to accidents or hazardous situations. Regular inspection and maintenance help identify potential risks and ensure that equipment operates safely.

E - Cost Reduction

Effective maintenance strategies help reduce overall operational costs. Preventive and predictive maintenance allow organizations to detect issues early and perform timely interventions, avoiding major failures that require expensive repairs or complete system replacements.

F - Improved Product Quality

Well-maintained machines operate with greater accuracy and stability, which contributes to improved product quality. When equipment functions correctly, production processes become more consistent and reliable, reducing defects and waste.

I.3.2 Types of Maintenance

In industrial practice, maintenance strategies have evolved from simple reactive approaches to more advanced predictive techniques. Each type of maintenance has distinct objectives, advantages, and limitations that affect equipment performance, reliability, and operational efficiency. The following sections summarize the main categories as recognized in engineering literature [8].

I.3.2.1 Reactive Maintenance (RM)

Reactive maintenance refers to maintenance actions carried out only after equipment failure occurs. In this strategy, no prior monitoring or maintenance planning is performed, and the system is repaired only when it stops operating. Although this approach may reduce short-term maintenance costs, it often results in unexpected downtime and increased repair expenses in the long term [10].

Advantages of Reactive Maintenance

- **Low initial cost:** Since maintenance is performed only after a breakdown, there are minimal upfront expenses [10].

- **Simplicity in execution:** No extensive planning or monitoring is required before failure occurs.

Drawbacks of Reactive Maintenance

- **High overall costs:** Emergency repairs, unplanned spare part procurement, and unscheduled downtime increase total operational costs.
- **Reduced reliability:** Equipment failures occur unpredictably, leading to disruptive production losses.
- **Safety hazards:** Operating machines until failure can present risks to workers due to unforeseen breakdown behavior.
- **Inefficient use of resources:** Technicians often respond without prior diagnosis, increasing repair time and effort [10].
- **Energy inefficiency:** Failed or poorly maintained equipment may consume more energy due to friction, overheating, or poor calibration.

I.3.2.2 Corrective Maintenance (CM)

Corrective maintenance is performed after the detection of a fault in order to restore the system to its normal operating condition. Unlike reactive maintenance, corrective actions may be taken before complete failure occurs, allowing maintenance operations to be planned more efficiently.

According to industrial standards such as ISO 14224, corrective maintenance generally includes all maintenance actions performed after fault occurrence. Therefore, the distinction between reactive and corrective maintenance may depend on the industrial context and adopted definitions.

In this work, corrective maintenance is considered as a maintenance strategy applied after fault detection to prevent complete system failure.

Advantages of Corrective Maintenance

- **Targeted problem resolution:** Issues are addressed before major disruptions occur.
- **Reduced urgency interventions:** Planned corrective actions decrease emergency repairs.
- **Improved safety:** Equipment restored to working order supports a safer operational environment.

Drawbacks of Corrective Maintenance

- **Dependency on monitoring:** Without effective condition monitoring, corrective actions may still follow failures.

- **Unplanned delays:** Limited early warning can prolong downtime if spare parts or diagnostics are unavailable [10].

I.3.2.3 Preventive Maintenance (PM)

Preventive maintenance is a maintenance strategy based on performing maintenance operations at scheduled time intervals, regardless of the actual condition of the equipment [10]. Its main objective is to reduce the probability of unexpected failures and extend equipment lifetime.

However, this approach may sometimes lead to unnecessary maintenance actions on components that are still operating correctly. In some cases, maintenance interventions themselves may introduce new faults due to installation errors or handling issues. This phenomenon is commonly known as "infant mortality," where recently replaced components fail prematurely.

Advantages of Preventive Maintenance:

- **Reduced unexpected failures:** By performing regular maintenance, equipment is less likely to fail unexpectedly [10].
- **Ease of planning:** Maintenance schedules can be planned in advance, simplifying workforce allocation and spare parts management [8].
- **Improved reliability:** Regular checks ensure equipment continues to operate at optimal performance levels [10].

Drawbacks of Preventive Maintenance:

- **Potential unnecessary maintenance:** Parts or components may be replaced or serviced even if still in good condition, leading to wasted resources [10].
- **Fixed schedule limitations:** Maintenance timing is based on a calendar rather than actual equipment condition, which may not reflect real needs.

I.3.2.4 Predictive Maintenance (PdM)

Predictive maintenance is an advanced maintenance approach based on continuous condition monitoring and data analysis to predict equipment failures before they occur. It relies on sensors, data acquisition systems, and intelligent algorithms to evaluate equipment condition and estimate the Remaining Useful Life (RUL) of components.

Unlike preventive maintenance, maintenance actions in PdM are performed only when there are indications of degradation or abnormal behavior. This reduces unnecessary interventions and helps improve equipment reliability, system availability, and maintenance efficiency while reducing operational costs [8].

Advantages of Predictive Maintenance

- **Reduced unplanned failures:** Early detection of degradation prevents unexpected breakdowns.
- **Optimized maintenance planning:** Maintenance is based on actual condition rather than arbitrary schedules.
- **Cost efficiency:** Minimizes unnecessary parts replacement and labor tasks.
- **Enhanced availability:** Equipment remains in service longer with fewer interruptions.
- **Integration with advanced technologies:** IoT, machine learning, and Big Data analytics enhance predictive capabilities.

Drawbacks of Predictive Maintenance

- **Complex data interpretation:** Misinterpretation of sensor data can trigger incorrect maintenance actions.
- **Implementation costs:** Deploying sensors, data infrastructure, and analytics tools can involve significant investment.
- **Contextual limitations:** Predictive models may not always factor in all contextual variables such as operational environment or equipment aging.
- **Risk of reduced physical inspection:** Reliance on data may unintentionally reduce hands-on inspection practices [10].
- **Schedule constraints:** In some cases, traditional preventive routines may still be triggered despite real condition data.

I.3.3 Integration of IoT Technologies in Predictive Maintenance (PdM)

The integration of Internet of Things (IoT) technologies into predictive maintenance systems has improved the way industrial equipment is monitored and maintained. IoT enables the continuous collection of real-time data from sensors installed on industrial machines. These sensors monitor important operating parameters such as temperature, vibration, pressure, and rotational speed, providing information about equipment condition and system health [11].

Using IoT technologies, maintenance teams can monitor machine behavior continuously and detect abnormal operating conditions at an early stage. This helps reduce unexpected failures and allows maintenance operations to be scheduled before serious breakdowns occur. As a result, industries can move from traditional time-based maintenance toward condition-based maintenance strategies [11].

IoT data can also be combined with analytical and machine learning techniques to improve fault diagnosis and maintenance decision-making. By analyzing real-time operating

data, it becomes possible to identify degradation patterns and estimate possible equipment failures more accurately.

Another advantage of IoT-based predictive maintenance is the possibility of remote monitoring and supervision. Maintenance teams can access equipment data remotely, reducing the need for continuous on-site inspections and helping improve maintenance efficiency and system availability [11].

I.4 Types of Machine Faults

Industrial machines may experience several types of faults that affect their performance, reliability, and operational safety. These faults can originate from mechanical degradation, electrical malfunctions, thermal stress, lubrication problems, or environmental conditions. Identifying these faults at an early stage is essential to ensure effective maintenance and to prevent unexpected system failures [12], [13]. These faults can be classified into several categories as follows:

I.4.1 Mechanical Faults

Mechanical faults occur in the physical components of machines due to wear, fatigue, or excessive mechanical loads. Common examples include wear, cracks, deformation, and component breakage, which may lead to vibration, misalignment, and reduced machine efficiency [12].

I.4.2 Electrical Faults

Electrical faults occur in electrical circuits and equipment such as motors, cables, or control systems. Typical faults include short circuits, overvoltage, undervoltage, and insulation failure, which may cause abnormal current flow and lead to equipment damage or shutdown [13].

I.4.3 Thermal Faults

Thermal faults are related to abnormal temperature conditions in machines. Overheating or insufficient cooling may damage machine components, reduce insulation life, and accelerate equipment deterioration [12].

I.4.4 Hydraulic Faults

Hydraulic systems may suffer from faults such as fluid leakage, blockage in pipelines or valves, and cavitation, which can reduce pressure levels and negatively affect system performance [13].

I.4.5 Lubrication Faults

Lubrication faults occur when the lubricant is insufficient, contaminated, or degraded, leading to increased friction, overheating, and accelerated wear of mechanical parts [12].

I.4.6 Environmental Faults

Environmental factors such as corrosion, dust accumulation, and excessive vibrations may also contribute to the degradation of machine components and reduce their operational lifespan [13].

I.4.7 Control and Communication Faults

Modern industrial machines rely on sensors and communication systems for monitoring and control. Sensor failures or communication faults may lead to incorrect data transmission and inaccurate system monitoring, affecting the reliability of maintenance systems [13].

I.5 Conclusion

This chapter introduced the main concepts related to Industry 4.0 and industrial maintenance. The basic characteristics of Industry 4.0 technologies were presented, with a focus on technologies such as IoT, Big Data, and Artificial Intelligence.

The chapter also discussed the importance of maintenance in industrial environments and presented the different maintenance strategies, including reactive, corrective, preventive, and predictive maintenance. In addition, the common categories of machine faults affecting industrial equipment were briefly explained.

Finally, the role of IoT technologies in predictive maintenance systems was presented, particularly for machine monitoring and fault detection. The concepts discussed in this chapter provide the necessary background for the following chapters of this work.

Chapter II:

Fundamentals of IoT and ML for Predictive Maintenance.

II.1 Introduction

The previous chapter introduced the general concepts of predictive maintenance and the different categories of machine faults. In this chapter, the focus is placed on the main technologies used in predictive maintenance systems, particularly the Internet of Things (IoT) and Machine Learning (ML).

IoT technologies allow industrial equipment to be monitored through sensors capable of collecting real-time operational data. These data can then be analyzed using machine learning techniques to identify abnormal machine behavior and predict possible failures before they occur.

This chapter presents the basic concepts of IoT, its architecture, and its role in predictive maintenance applications. It also introduces the fundamental principles of machine learning, its main categories, and several algorithms commonly used for fault detection and prediction.

II.2 Internet of Things (IoT)

The Internet of Things (IoT) has emerged as one of the most important technologies of modern digital systems. It enables physical devices to be interconnected and communicate with each other through the internet. By integrating sensors, communication technologies, and data processing systems, IoT allows the collection and exchange of large amounts of data in real time. In industrial environments, IoT plays a crucial role in monitoring equipment conditions and supporting intelligent maintenance strategies such as predictive maintenance [14].

II.2.1 Definition of IoT

The Internet of Things can be defined as a system in which physical objects are connected to the internet and are able to communicate with each other and with external systems through embedded sensors and communication technologies. These connected devices collect large amounts of data from their surrounding environment, which can then be processed and analyzed to support intelligent decision-making processes. IoT technologies are widely used in various fields such as smart cities, healthcare, transportation, and industrial automation [14], [5].

II.2.2 IoT Architecture

The architecture of an IoT system generally consists of several layers that work together to enable data collection, communication, processing, and analysis. The first layer is the perception layer, which includes sensors and devices responsible for collecting data from the physical environment. The second layer is the network layer, which ensures the transmission of collected data through communication technologies such as wireless networks or the internet. The third layer is the processing or middleware layer, where the collected data are stored and analyzed using various data processing techniques. Finally, the application layer provides

services to end users by transforming the processed data into useful information for monitoring, control, and decision-making [5].

The general architecture of an IoT system is illustrated in Figure II.1.

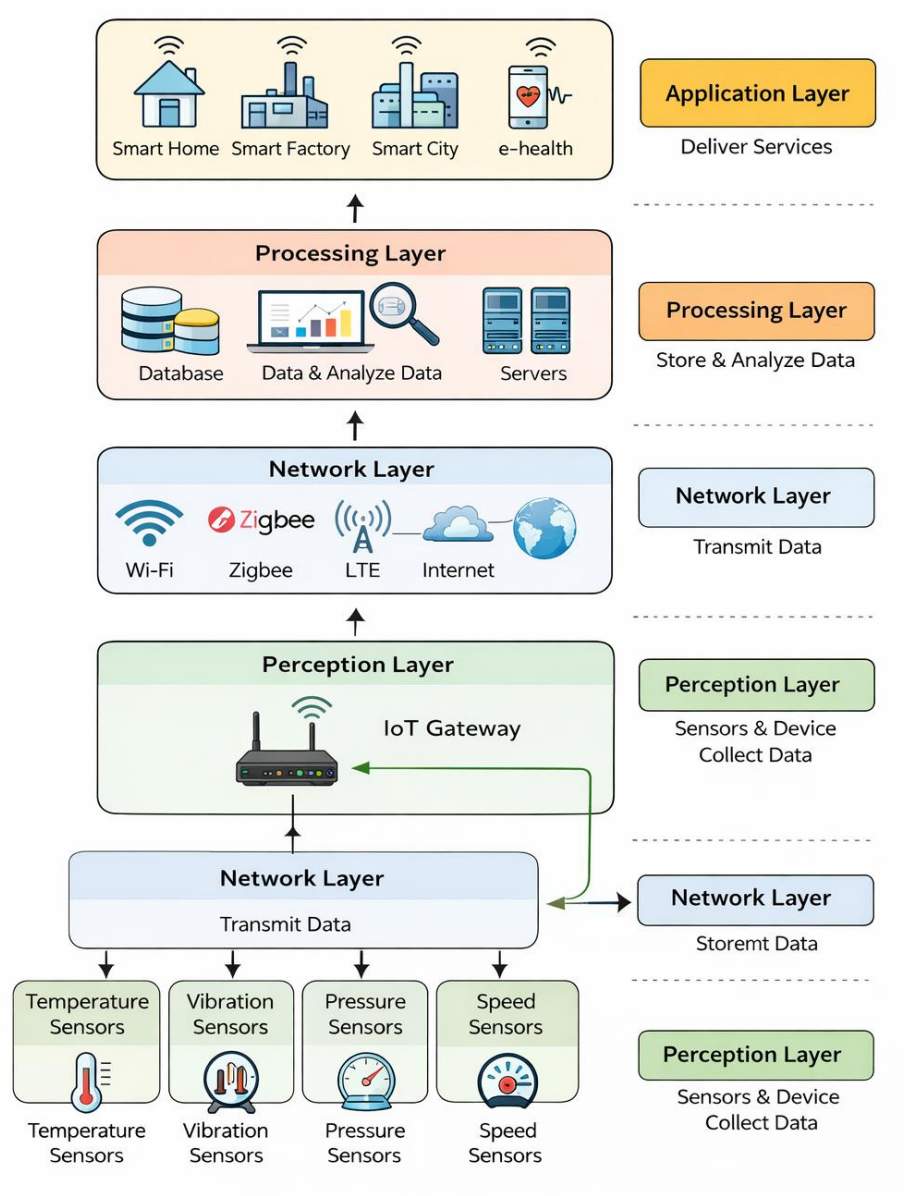


Figure II.1: Layered Architecture of the Internet of Things (IoT) [5]

II.2.3 IoT in Predictive Maintenance

In predictive maintenance systems, IoT technologies are used to continuously monitor the operational conditions of industrial equipment through various sensors such as vibration, temperature, pressure, and speed sensors. These sensors collect real-time data from machines and transmit them to data processing platforms where advanced analytical techniques can be applied. By analyzing these data, it becomes possible to detect abnormal patterns and predict potential failures before they occur. As a result, IoT-based predictive maintenance systems help

industries reduce maintenance costs, minimize unexpected downtime, and improve equipment reliability and productivity [15].

II.3 Machine Learning

Machine Learning (ML) is a key technology in modern intelligent systems that enables computers to learn from data and improve their performance without being explicitly programmed. It is widely used in various applications such as image recognition, natural language processing, and predictive maintenance. In industrial environments, machine learning techniques are used to analyze large volumes of data collected from sensors in order to detect patterns, identify anomalies, and predict potential equipment failures. This makes ML an essential component in IoT-based predictive maintenance systems [16].

II.3.1 Definition of Machine Learning

Machine Learning can be defined as a field of artificial intelligence that focuses on developing algorithms capable of learning from data and making predictions or decisions based on it. Instead of relying on predefined rules, machine learning models improve their performance automatically as they are exposed to more data. These models can identify hidden patterns and relationships within data, making them suitable for complex tasks such as fault detection and prediction in industrial systems [16]

II.3.2 Types of Machine Learning

Machine Learning techniques can be broadly classified into different categories depending on the type of data and learning process involved. The most common types are supervised learning and unsupervised learning, which are widely used in predictive maintenance applications.

II.3.2.1 Supervised Learning

Supervised learning is a type of machine learning in which the model is trained using labeled data. This means that each input data is associated with a known output, allowing the model to learn the relationship between inputs and outputs. Once trained, the model can predict the output for new unseen data. Supervised learning is commonly used in predictive maintenance for tasks such as fault classification and failure prediction [17]. this is illustrated in Figure II.2.

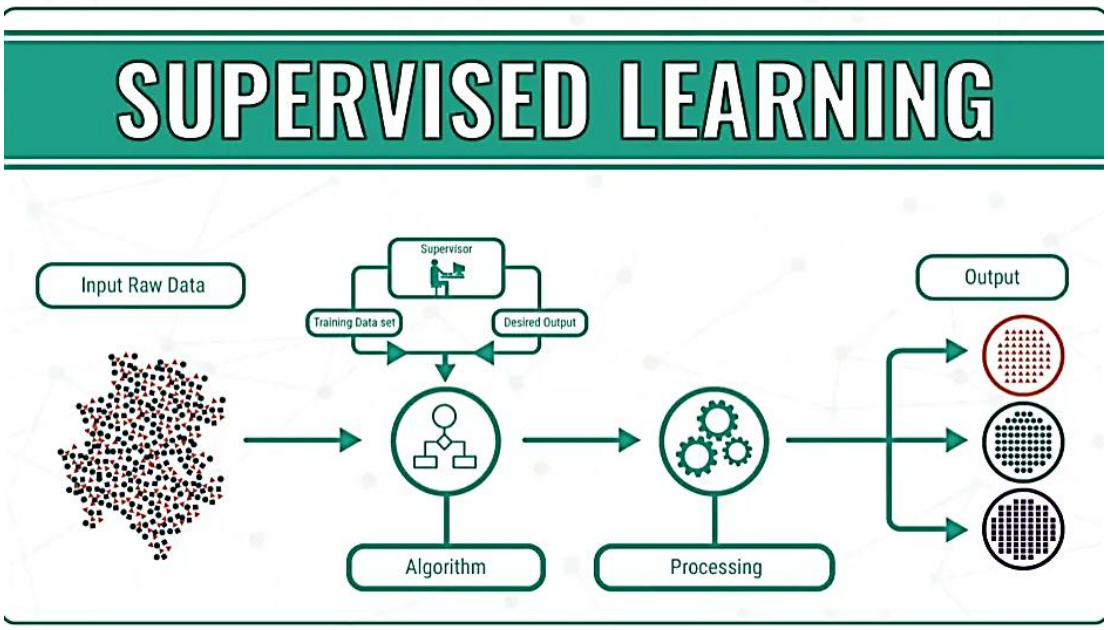


Figure II.2: Supervised Learning Model Using Labeled Data

Classification

Classification is a supervised learning technique used to assign input data to predefined categories or classes. In predictive maintenance, classification algorithms can be used to determine whether a machine is operating normally or experiencing a fault. this is illustrated in Figure II.3.

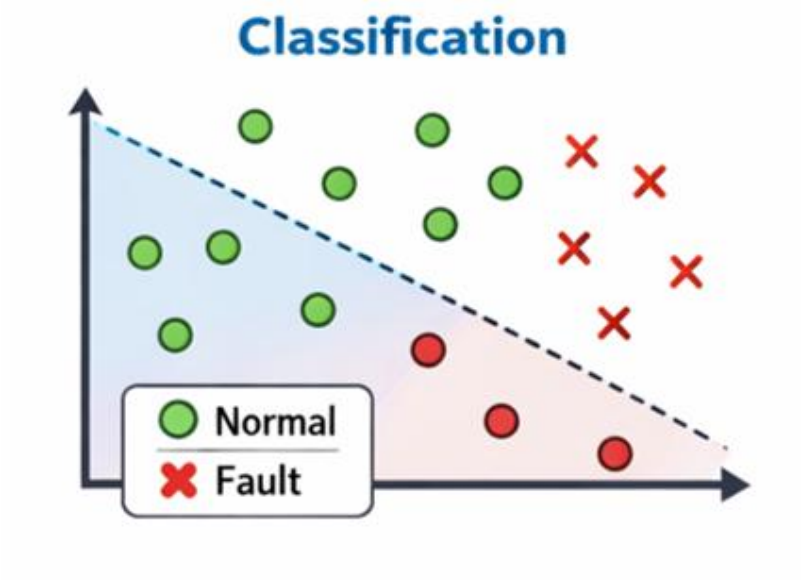


Figure II.3: Classification Technique for Fault Detection

Regression

Regression is another supervised learning technique used to predict continuous numerical values. In industrial applications, regression models can be used to estimate parameters such as remaining useful life (RUL) of equipment or future temperature values. this is illustrated in Figure II.4.

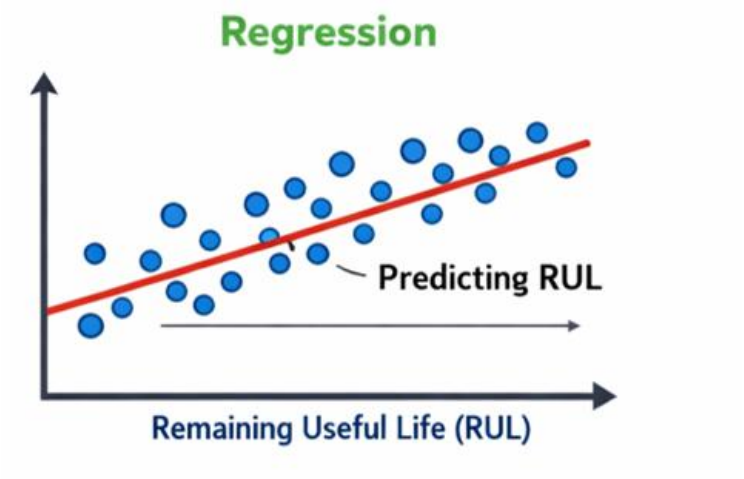


Figure II.4: Regression Model for Predicting Remaining Useful Life (RUL)

II.3.2.2 Unsupervised Learning

Unsupervised learning is a type of machine learning where the model is trained using unlabeled data. Unlike supervised learning, there are no predefined outputs, and the model tries to discover hidden patterns or structures within the data. This approach is useful for tasks such as clustering and anomaly detection, which are important in predictive maintenance for identifying unusual machine behavior [17]. this is illustrated in Figure II.5.

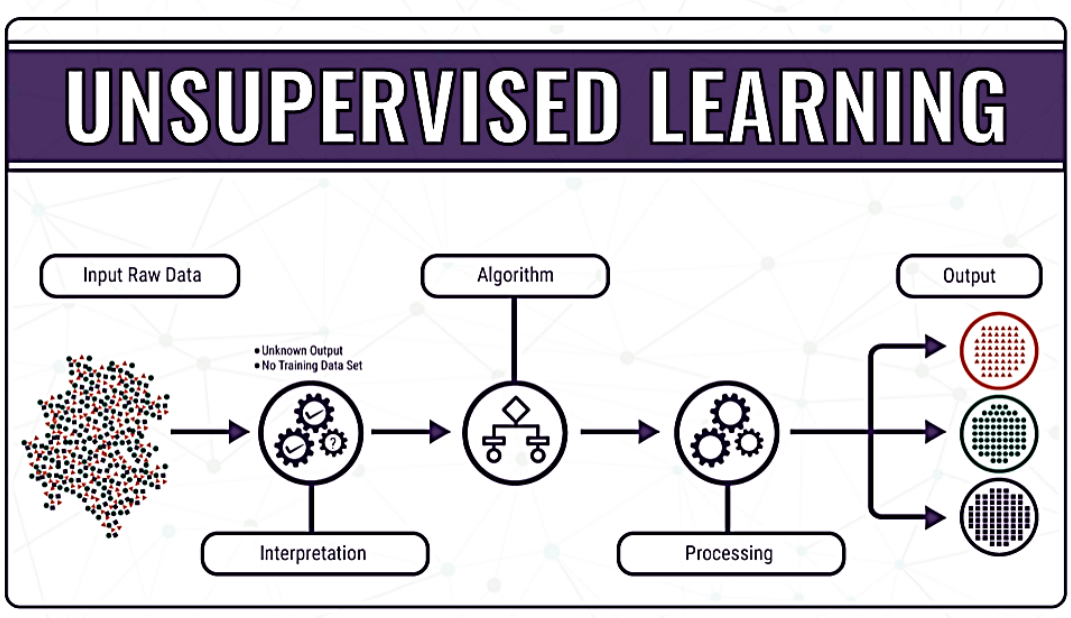


Figure II.5: Unsupervised Learning for Pattern Discovery and Anomaly Detection

II.4 Machine Learning Algorithms

Machine learning algorithms are essential tools used to analyze data, identify patterns, and make predictions. In predictive maintenance, these algorithms process data collected from IoT sensors to detect faults, classify machine conditions, and estimate future failures. Among the most widely used algorithms in this domain are Logistic Regression, K-Nearest Neighbors (KNN), and Random Forest, due to their effectiveness and reliability in handling industrial data [16], [17].

II.4.1 Logistic Regression

Logistic Regression is a supervised machine learning algorithm mainly used for classification problems. It estimates the probability that a given input belongs to a particular class by using a logistic (sigmoid) function. The output is typically binary, such as normal or faulty condition in predictive maintenance systems.

This algorithm is widely used due to its simplicity, interpretability, and efficiency when dealing with linearly separable data. In industrial applications, logistic regression can be used to predict whether a machine is likely to fail based on sensor data [16]. this is illustrated in Figure II.6.

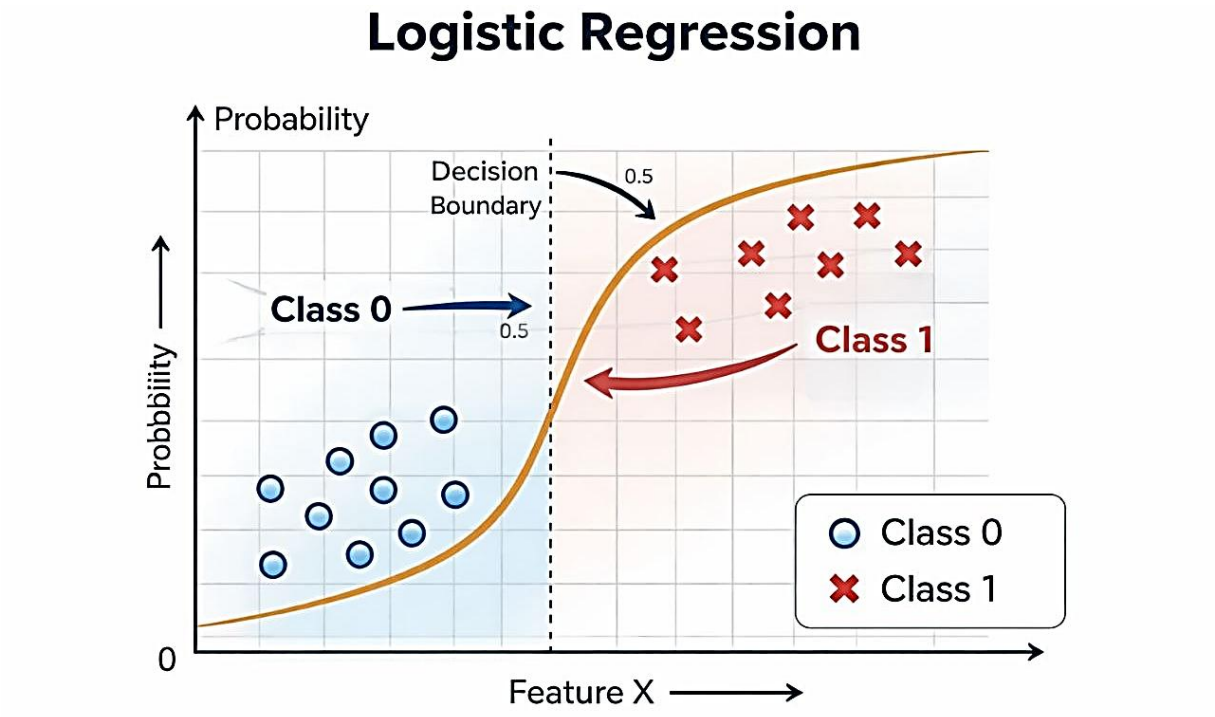


Figure II.6: Logistic Regression Model for Binary Classification

II.4.2 K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) is a supervised learning algorithm used for both classification and regression tasks. It operates by identifying the "k" closest data points to a given input based on a distance metric and assigning the most frequent class among its neighbors.

KNN is a non-parametric and simple algorithm that is particularly useful for pattern recognition and fault classification in predictive maintenance applications. However, its performance is highly dependent on the choice of the parameter *k* and the distance metric used.

One important limitation of KNN is the **curse of dimensionality**. In high-dimensional datasets, such as those generated by IoT sensors, the distance between data points becomes less meaningful, making it difficult to accurately identify nearest neighbors. As a result, the performance of KNN may significantly degrade when dealing with large numbers of features.

[17] this is illustrated in Figure II.7.

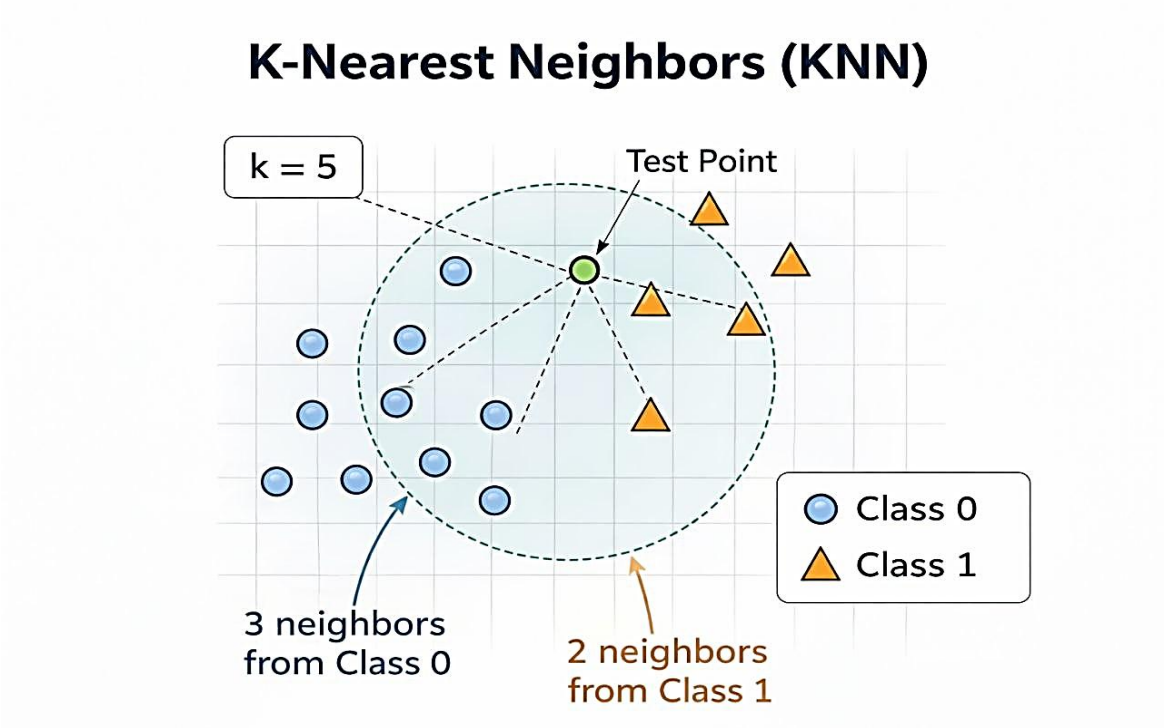


Figure II.7: K-Nearest Neighbors (KNN) Algorithm for Similarity-Based Classification

II.4.3 Random Forest

Random Forest is an ensemble learning algorithm that combines multiple decision trees to improve prediction accuracy and reduce overfitting. Each tree in the forest is trained on a random subset of the data, and the final prediction is obtained by aggregating the results of all trees.

This algorithm is highly effective in handling large datasets and complex relationships between variables. In predictive maintenance, Random Forest is commonly used for fault detection and classification due to its robustness and high accuracy [17] this is illustrated in Figure II.8.

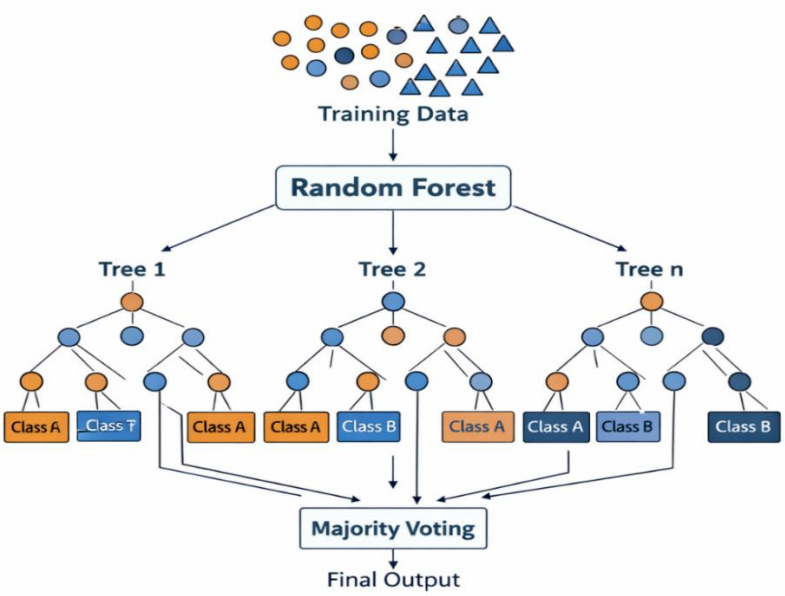


Figure II.8: Random Forest Algorithm Based on Multiple Decision Trees

II.5 Artificial Neural Networks (ANN)

Artificial Neural Networks (ANNs) are computational models inspired by the structure and functioning of the human brain. The concept of artificial neurons was first introduced by Warren McCulloch and Walter Pitts in 1943, where they proposed a mathematical model capable of representing simple logical functions. Since then, ANNs have evolved into powerful tools for pattern recognition and data analysis.

An artificial neural network consists of a set of interconnected processing units (neurons) that work together to learn complex relationships within data. These networks are designed to mimic certain characteristics of biological neural systems. They are typically:

- Massively parallel in structure
- Capable of learning from data
- Able to store information within weighted connections
- Robust in handling incomplete or noisy data

Due to these properties, ANNs are widely used in industrial applications such as fault diagnosis, predictive maintenance, and pattern recognition **[18]**.

The basic structure of an artificial neural network, including input, hidden, and output layers, is illustrated in Figure II.9.

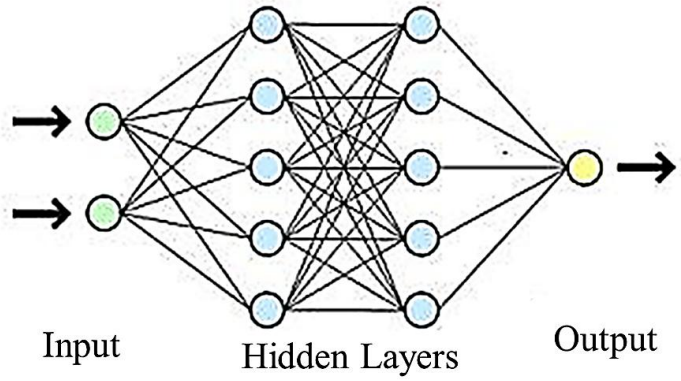


Figure II.9: Basic Structure of an Artificial Neural Network

II.5.1 Structure of Neural Networks

Artificial Neural Networks (ANNs) are computational models inspired by the structure and functioning of the human brain. They consist of interconnected layers of neurons that process input data to produce meaningful outputs.

A typical ANN structure includes an input layer, one or more hidden layers, and an output layer. The input layer receives raw data, which is then propagated through successive hidden layers. Each layer applies weighted transformations to the input data using adjustable parameters called **weights**.

The input data are first processed through multiple layers to generate a prediction output (Y). This predicted output is then compared with the true label using a **loss function**, which measures the error between the predicted and actual values.

To improve the model's performance, an optimizer is used to update the weights based on the computed error. This process is repeated iteratively during training in order to minimize the loss function and enhance the accuracy of the neural network.

This architecture not only represents the structural components of the network but also illustrates the learning mechanism through which the model adjusts its parameters to achieve better predictions**19]]**.

The general structure and training process of an artificial neural network are illustrated in Figure II.10.

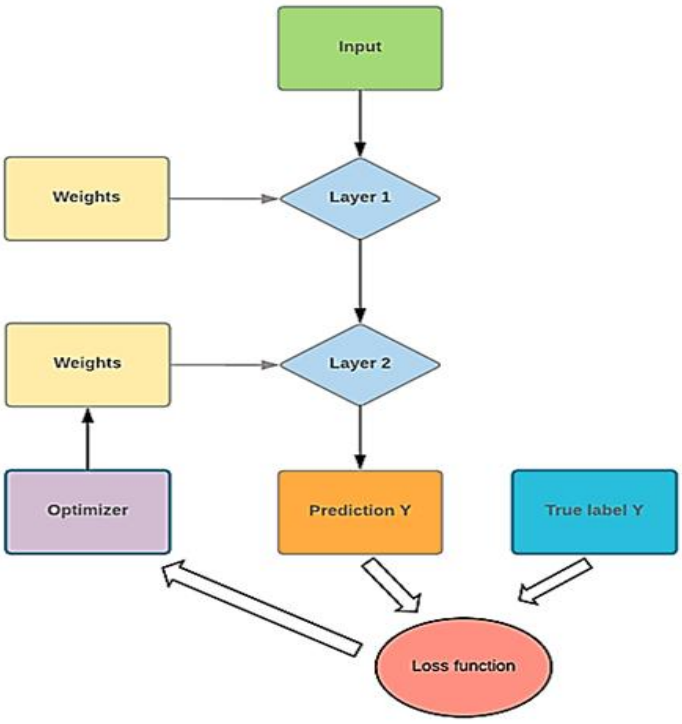


Figure II.10: Structure and Training Process of an Artificial Neural Network

II.5.2 Working Principle of Neural Networks

The operation of an Artificial Neural Network (ANN) is based on a sequence of steps that enable it to process data and learn from it:

Input Stage (Input Layer):

The input data are fed into the network through the input neurons, where each neuron represents a specific feature of the dataset.

Forward Propagation (Hidden Layers):

The input data are propagated through one or more hidden layers. Each neuron computes a weighted sum of its inputs and applies an activation function to introduce non-linearity. This process enables the network to learn complex patterns and relationships within the data.

Output Generation (Output Layer):

The processed data reach the output layer, where the final prediction or classification result is produced.

Learning Process (Training Phase):

During training, the network adjusts its weights using optimization algorithms in order to minimize the difference between predicted and actual outputs.

Backpropagation:

Backpropagation is a key learning mechanism used to update the network weights. It works by propagating the error backward from the output layer to the input layer and computing the gradient of the loss function with respect to each weight. This allows the model to iteratively adjust its parameters using optimization methods such as gradient descent.

This learning capability enables neural networks to improve their performance over time and makes them highly effective for predictive maintenance applications [20].

II.5.3 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) designed to process sequential and time-dependent data. Unlike traditional neural networks, LSTM networks can retain relevant information over long periods, making them suitable for applications involving temporal patterns and time-series analysis [21].

An LSTM network consists of memory cells and a set of gates that regulate the flow of information. These gates control which information should be stored, updated, or discarded during the learning process. This structure helps overcome the limitations of conventional RNNs, particularly the vanishing gradient problem encountered when learning long sequences [21].

Due to their ability to capture temporal dependencies, LSTM networks have been widely adopted in predictive maintenance applications. They are particularly effective for analyzing sensor measurements collected over time and estimating the Remaining Useful Life (RUL) of industrial equipment. In this work, the LSTM model is used to learn degradation patterns from sequential data and improve RUL prediction accuracy. The basic architecture of an LSTM network is illustrated in Figure II.11.

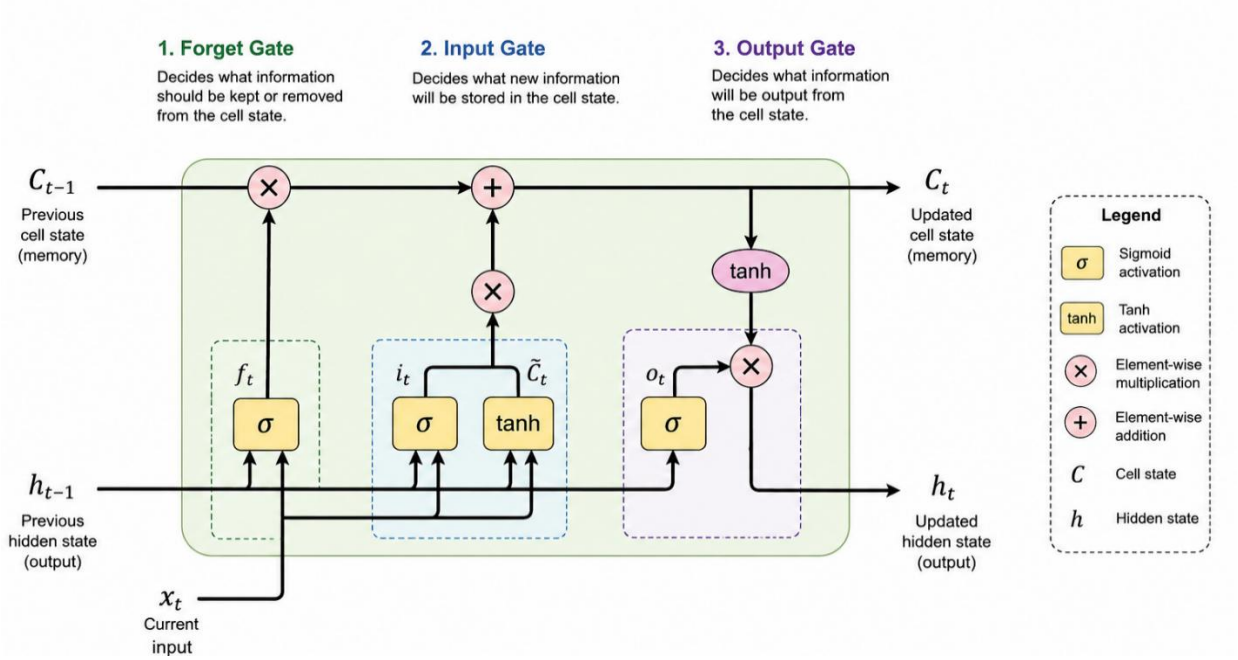


Figure II.11: Architecture of a Long Short-Term Memory (LSTM) Network

The LSTM architecture consists of memory cells and three main gates: the forget gate, the input gate, and the output gate. These components allow the network to retain relevant information over long sequences and model temporal dependencies effectively.

II.6 Related Work

Study 1

Abd El-Aziz et al. proposed a predictive maintenance system based on the Internet of Things (IoT) and machine learning techniques for industrial applications. In their approach, sensors installed on industrial machines collect operational data such as temperature, vibration, and speed. These data are transmitted through IoT networks and analyzed using machine learning algorithms to detect abnormal behavior and predict potential failures. The results of the study showed that integrating IoT technologies with machine learning models can significantly improve fault detection accuracy and reduce unexpected machine downtime in industrial environments [22].

Study 2

Aboshosha et al. presented an IoT-based predictive maintenance framework that combines artificial intelligence techniques and data-driven analysis to monitor industrial equipment. The proposed system uses IoT sensors to continuously collect machine data, which are then analyzed using artificial neural networks and fuzzy systems to detect anomalies and predict failures. The study demonstrated that intelligent predictive maintenance systems can

improve equipment reliability and help industries reduce maintenance costs and increase operational efficiency [23].

Study 3

Wang et al. proposed an IoT-enabled Prognostics and Health Management (PHM) system for manufacturing environments. The framework combines IoT technologies with deep learning techniques to monitor equipment health and estimate degradation trends. The authors integrated sensor data acquisition, cloud-based processing, and deep learning models to improve fault diagnosis and Remaining Useful Life (RUL) prediction. The results demonstrated that the proposed approach can effectively support predictive maintenance and reduce unplanned downtime in industrial systems [24].

Study 4

Yu et al. presented an edge computing-assisted IoT framework for predictive maintenance in smart manufacturing. The proposed architecture combines Industrial IoT devices with an autoencoder-based machine learning model for fault detection. Data collected from industrial sensors are processed near the edge to reduce latency and communication overhead. Experimental results showed that the framework improves fault detection performance while supporting real-time maintenance decision-making [25].

Table II.1: Comparison of Related Works

Study	IoT	ML/DL	Fault Detection	RUL Prediction
Abd El-Aziz et al. [22]	✓	✓	✓	✗
Aboshosha et al. [23]	✓	✓	✓	✗
Wang et al. [24]	✓	✓ (Deep Learning)	✓	✓
Yu et al. [25]	✓	✓ (Autoencoder)	✓	✗
Proposed Work	✓	✓ (RF + LSTM)	✓	✓

The comparison shows that most previous studies focused on fault detection and equipment monitoring using IoT and machine learning techniques. However, only a limited number of works addressed Remaining Useful Life (RUL) prediction. The proposed work combines IoT-based monitoring with both machine learning (Random Forest) and deep learning (LSTM) models to estimate RUL and support predictive maintenance decisions.

II.7 Conclusion

This chapter presented the main technologies involved in predictive maintenance systems, with a focus on the Internet of Things (IoT) and Machine Learning (ML). IoT technologies enable the collection of real-time data from industrial equipment, while machine learning methods help analyze these data to detect abnormal conditions and predict machine faults.

Several machine learning algorithms were introduced, including Logistic Regression, K-Nearest Neighbors (KNN), and Random Forest, in addition to Artificial Neural Networks (ANN). These techniques play an important role in improving maintenance strategies and supporting intelligent monitoring systems.

The next chapter will present the design and implementation of the proposed IoT-based predictive maintenance system.

Chapter III:

IoT System Design and Implementation.

III.1 Introduction

The previous chapters presented the theoretical background related to predictive maintenance, Industry 4.0, IoT, and machine learning. These concepts are important for understanding the development of intelligent maintenance systems for industrial applications.

This chapter focuses on the practical part of the proposed work. It presents the development environment and describes the architecture of the proposed IoT-based system, including its main components such as sensors, communication modules, and data processing units.

The chapter also explains the data flow process, starting from data acquisition and transmission to data processing and storage. In addition, a MATLAB/Simulink model is introduced to simulate induction motor behavior and generate operational data for predictive maintenance applications.

III.2 Development Environment

This section presents the development environment and tools used to design and implement the proposed IoT system. Python was adopted due to its simplicity and extensive libraries for data processing. Google Colab was used for coding and testing in a cloud environment, while Visual Studio Code was used for code development and organization. These tools provide an efficient framework for system implementation and data preparation.

III.2.1 Google Colab

Google Colab is a cloud-based platform that enables users to write and execute Python code through a web browser, providing access to computational resources such as CPUs and GPUs [26].

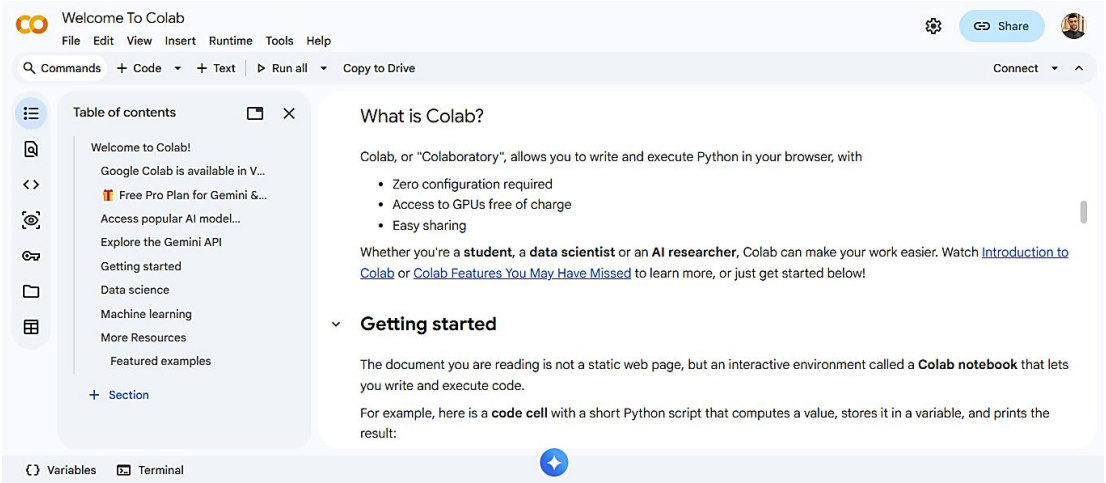


Figure III.1: Google Colab Platform Interface

III.2.2 Visual Studio Code

Visual Studio Code is a lightweight source-code editor developed by Microsoft, offering features such as debugging, extensions, and support for multiple programming languages [27].



Figure III.2: Visual Studio Code Logo

III.2.3 Python

Python is a high-level programming language widely used in data science and system development due to its readability and extensive ecosystem of libraries [28].



Figure III.3: Python Programming Language Logo

III.2.4 Pandas

Pandas is a Python library used for data manipulation and analysis, providing flexible data structures such as DataFrames [29].



Figure III.4: Pandas Library Logo

III.2.5 NumPy

NumPy is a fundamental library for numerical computing in Python, supporting large arrays and mathematical operations [30].



Figure III.5: NumPy Library Logo

III.2.6 Keras

Keras is a high-level deep learning API used for building and training neural networks efficiently [31].



Figure III.6: Keras Library Logo

III.2.7 TensorFlow

TensorFlow is an open-source platform developed by Google for machine learning and deep learning applications [32].

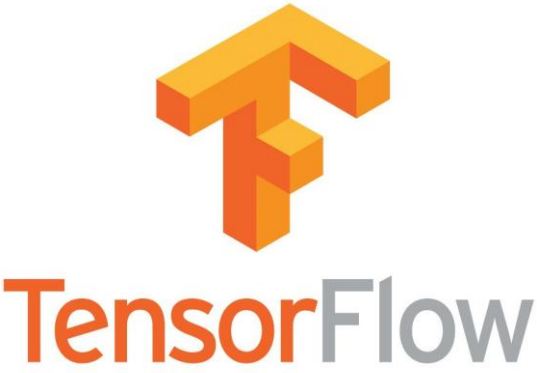


Figure III.7: TensorFlow Framework Logo

III.3 IoT System Design

This section presents the design of the proposed IoT-based system for predictive maintenance. The primary objective is to develop a framework capable of collecting, transmitting, and processing data from industrial sensors in order to monitor machine health and prepare datasets for intelligent analysis [33]. The system is designed to be platform-independent, allowing implementation through either simulation environments or real hardware systems depending on application requirements.

The proposed architecture follows a multi-layer approach, ensuring modularity, scalability, and efficient data management, which are essential characteristics of modern Industrial Internet of Things (IoT) systems [34].

This structure enables a seamless flow of data from the physical perception layer to the final application stage.

III.3.1 IoT System Architecture

The proposed IoT-based system for predictive maintenance is designed following a layered architecture, which ensures modularity, scalability, and flexibility. This design approach is widely adopted in industrial IoT (IoT) systems, as it enables the separation of functionalities and facilitates system integration and future expansion.

As illustrated in Figure III.8, the architecture is organized into four main layers:

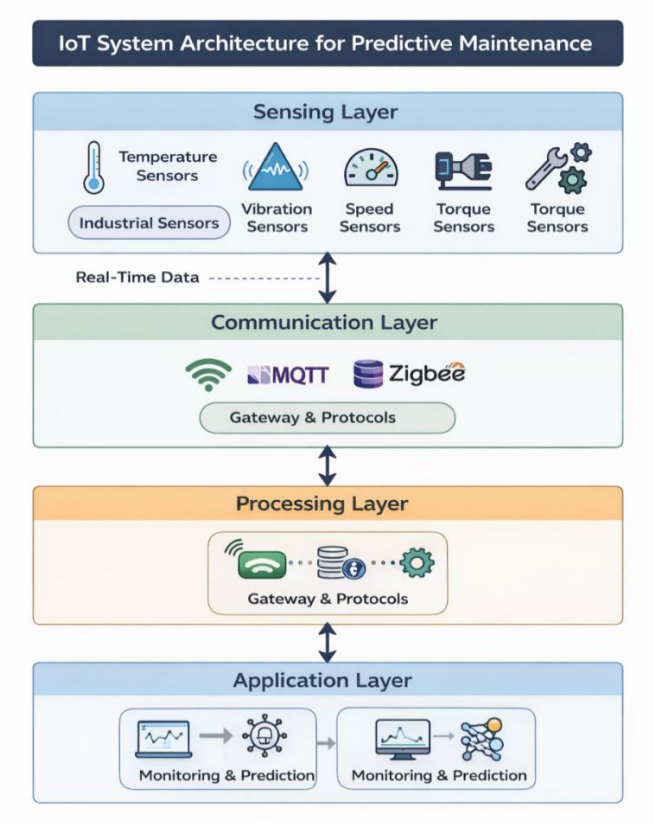


Figure III.8: Proposed IoT-Based Architecture for Induction Motor Predictive Maintenance

Sensing Layer:

This layer is responsible for the acquisition of physical data from the monitored system. It captures key operational parameters such as temperature, vibration, rotational speed, and torque, which reflect the real-time condition of the machine [35].

Communication Layer:

The communication layer ensures the transfer of data from the sensing components to the processing unit. It relies on standard communication technologies such as Wi-Fi, MQTT, or Zigbee to guarantee reliable and efficient data transmission [36]. In this work, this layer is conceptually represented, while the data exchange is simulated within the MATLAB environment.

Processing Layer:

This layer is responsible for data processing, filtering, normalization, and storage. It transforms raw sensor data into a structured format suitable for analysis and modeling [37]. In this study, it is implemented using MATLAB/Simulink.

Application Layer:

The application layer represents the final stage of the system, where processed data are used for monitoring and decision-making. It integrates machine learning techniques for fault detection and predictive maintenance, including Remaining Useful Life (RUL) estimation [33].

III.3.2 System Components

The proposed IoT-based system is composed of several essential components, each performing a specific function to ensure reliable data acquisition and processing:

Industrial Sensors:

High-precision sensors such as temperature, vibration, and speed sensors are used to monitor the operating conditions of the induction motor. These sensors continuously capture real-time physical parameters that reflect the health state of the machine.

Data Acquisition (DAQ) Unit:

The DAQ unit is responsible for collecting signals from the sensors and converting analog measurements into digital data. This conversion ensures that the captured signals can be processed and analyzed within the computational environment.

Gateway/Communication Module:

This module ensures connectivity between the different layers of the system. It manages the transmission of data from the sensing layer to the processing unit using standard communication mechanisms [36]. In this work, the communication process is simulated through data transfer within the MATLAB environment.

Processing Unit:

The processing unit represents the core computational environment where data are filtered, normalized, and stored. It prepares the data in a structured format suitable for machine learning models used in predictive maintenance.

Each component plays a crucial role in ensuring the accuracy, reliability, and efficiency of the overall system, enabling seamless data flow from acquisition to analysis.

III.3.3 Data Flow Process

The data flow within the proposed system follows a structured sequence that ensures efficient handling and transformation of raw sensor data into actionable information:

1) Data Acquisition:

Real-time data are collected from industrial sensors, capturing key physical parameters such as vibration, temperature, and rotational speed.

2) Data Transmission:

The collected data are transferred through the communication layer to the processing unit. In this work, this step is simulated via data exchange within the MATLAB environment.

3) Data Processing and Storage:

The received data are filtered, normalized, and stored. This stage ensures data quality by removing noise and preparing the dataset for further analysis.

4) Data Preparation for Machine Learning:

The processed data are structured into a suitable format, enabling their use as input for predictive models in the next stage of the system.

This workflow helps organize the collected data and prepares them for machine learning analysis.

To further illustrate the interaction between the different components of the proposed system, a sequence diagram is presented in Figure III.9.

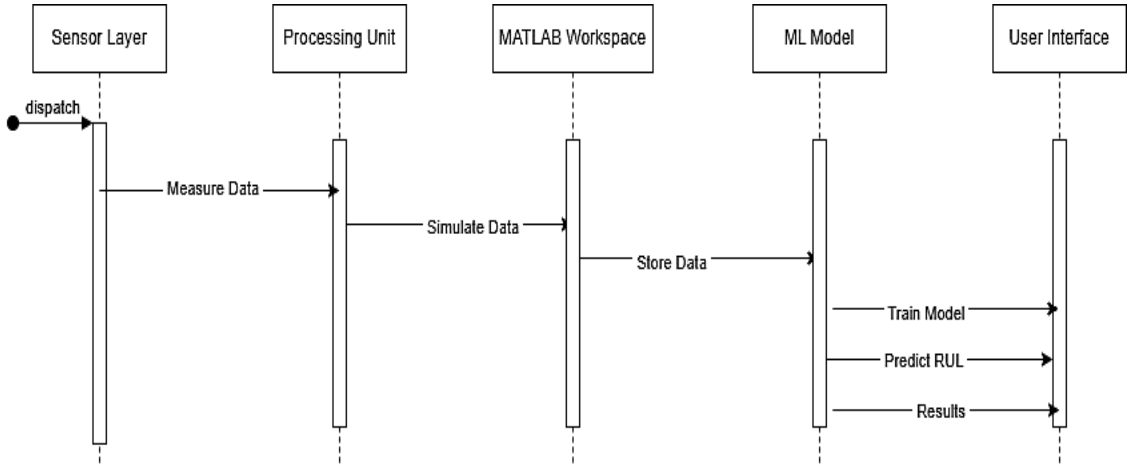


Figure III.9: Sequence Diagram of the Proposed IoT-Based Predictive Maintenance System

The sequence diagram shows how data move through the proposed system, starting from the sensing layer and ending at the machine learning stage. The generated data are stored in the MATLAB Workspace and later used for RUL prediction and system monitoring.

III.3.4 System Workflow

The operational workflow of the proposed system describes the transition from physical measurements to actionable insights, as illustrated in Figure III.10. This process can be divided into four main stages:

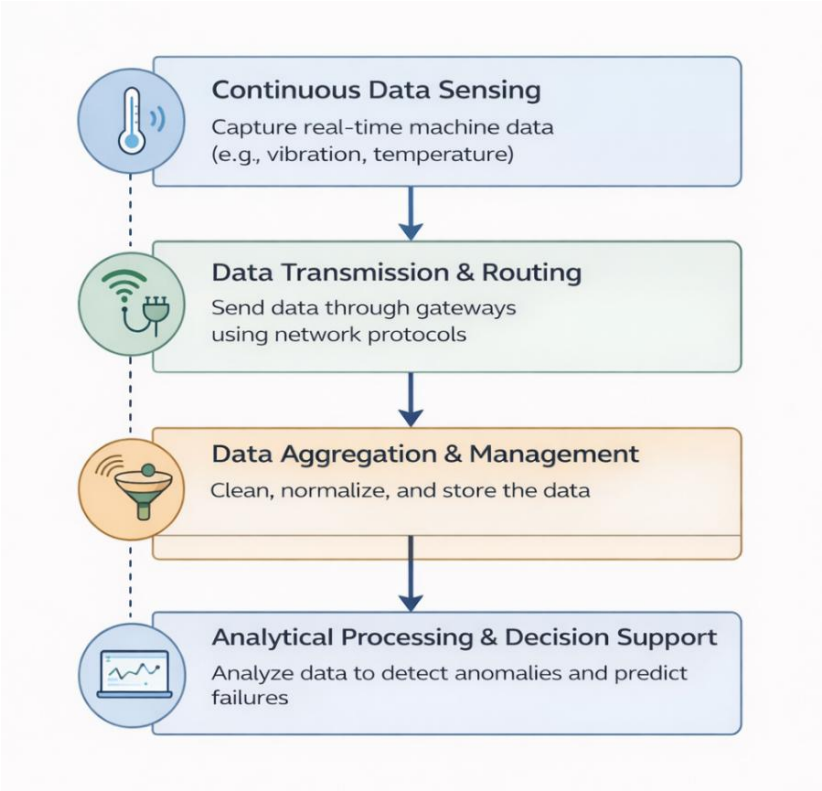


Figure III.10: Workflow of the Proposed IoT System for Predictive Maintenance

Continuous Data Sensing:

The system continuously monitors machine conditions using sensors that capture physical parameters such as vibration and temperature. These signals represent the primary input of the system [35].

Data Transmission and Routing:

The acquired data are transmitted through the communication layer using appropriate protocols and routing mechanisms, ensuring reliable and low-latency data transfer [36].

Data Aggregation and Management:

In the processing layer, the incoming data are aggregated, cleaned, and stored. This stage includes noise reduction and handling missing values to ensure data consistency and quality [37].

Analytical Processing and Decision Support:

Finally, the processed data are used for predictive maintenance applications. Machine learning models analyze the data to detect anomalies, predict failures, and support maintenance decision-making [33], [38].

III.3.5 System State Diagram

To better describe the operational behavior of the proposed predictive maintenance system, a state machine diagram is presented in Figure III.11.

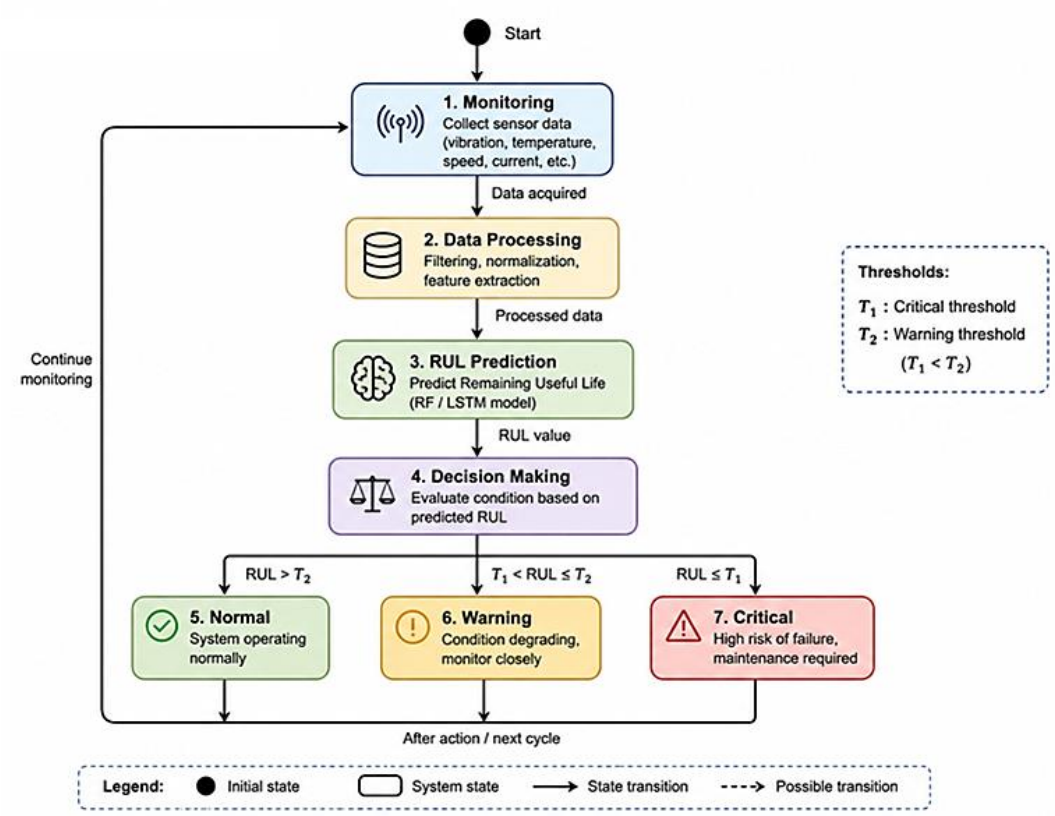


Figure III.11: State Machine Diagram of the Proposed Predictive Maintenance System

The state machine diagram shows how the proposed system operates during the monitoring process. The system begins by collecting sensor data, then processes the acquired information and predicts the Remaining Useful Life (RUL) of the machine. According to the predicted condition, the system may remain in a normal operating state, enter a warning state, or indicate a critical condition that requires maintenance. The monitoring cycle is then repeated continuously.

III.4 Simulation Model and Implementation

To validate the proposed IoT architecture, a simulation model was developed using MATLAB/Simulink. This stage represents the practical implementation of the sensing and processing layers discussed previously. The model focuses on an induction motor, which is widely used in industrial systems and commonly studied in predictive maintenance applications.

In this work, data transfer is simulated using the MATLAB Workspace. In a real industrial implementation, communication protocols such as MQTT or Wi-Fi could be used instead.

The simulation was configured using a continuous solver with a simulation time of 10 seconds, ensuring stable and accurate system behavior. This setup enables the generation of dynamic data that simulate real industrial operating conditions.

The induction motor used in this model represents a typical industrial machine widely employed in manufacturing systems. Its electrical and mechanical characteristics play a crucial role in generating realistic operational data for predictive maintenance analysis [39].

The main parameters of the induction motor used in the simulation are summarized in Table III.1.

Table III.1: Induction Motor Parameters

Parameter	Value
Rated Power	1.5 kW
Rated Voltage	400 V
Frequency	50 Hz
Rated Speed	1500 rpm
Number of Poles	4
Stator Resistance	1.2 Ω
Rotor Resistance	1.0 Ω

The developed model is illustrated in Figure III.12.

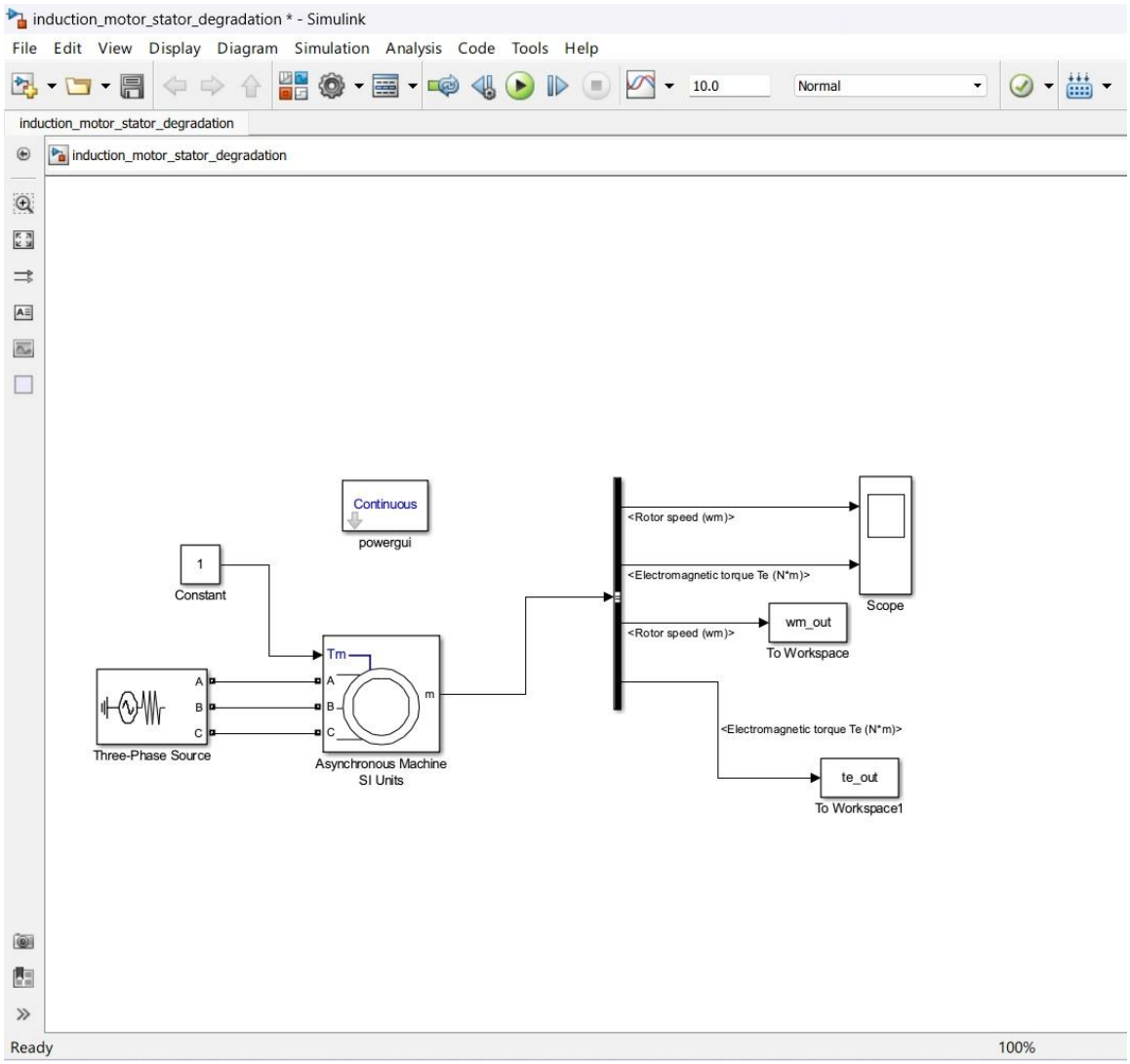


Figure III.12: MATLAB/Simulink Model of the Induction Motor for Data Acquisition

III.4.1 Model Components and Functionality

The developed Simulink model consists of the following main components:

Three-Phase Source and Powergui:

These blocks provide the electrical supply and define the simulation parameters required for accurate power system modeling.

Asynchronous Machine (Induction Motor):

This is the core component of the model, simulating the electrical and mechanical behavior of the motor and generating key operational data.

Mechanical Load (Constant Block):

A constant torque is applied to represent steady-state operating conditions of the machine.

Data Extraction and Sensing Interface:

Measurement blocks are used to capture key parameters such as rotor speed (wm) and electromagnetic torque (Te), simulating sensor outputs.

Output and Data Storage:

The generated signals are visualized using Scope blocks and exported to the MATLAB Workspace, where they are stored and prepared for further analysis.

The generated signals are treated as IoT sensor data and will be used in the next chapter for model training and evaluation.

III.5 Conclusion

This chapter presented the design and implementation of the proposed IoT-based predictive maintenance system. The main system architecture, components, data flow process, and operational workflow were described in detail.

In addition, a MATLAB/Simulink model was developed to simulate induction motor operation and generate data related to machine behavior and degradation conditions. This implementation provides the necessary data for the next stage of the work, where machine learning models will be trained and evaluated for fault prediction and Remaining Useful Life (RUL) estimation.

Chapter IV:

Implementation and Results.

IV.1 Introduction

This chapter presents the implementation and evaluation of the proposed predictive maintenance approach for Remaining Useful Life (RUL) prediction. The study is based on two datasets: a simulated dataset generated using a MATLAB/Simulink induction motor model under Broken Rotor Bar (BRB) conditions, and the real XJTU-SY bearing dataset.

This chapter explains how the collected signals were prepared before model training, including preprocessing, feature extraction, normalization, and RUL generation. It also presents the machine learning and deep learning models used for prediction and discusses their evaluation using regression metrics such as MAE, RMSE, and R^2.

The chapter is organized as follows: Section IV.2 presents the datasets, Section IV.3 describes data preparation, Section IV.4 introduces the prediction models, Section IV.5 explains the evaluation metrics, Section IV.6 discusses the obtained results, and Section IV.7 concludes the chapter.

IV.2 Dataset Description

This section briefly presents the two datasets used in this study. The detailed preprocessing steps and feature definitions are provided later in Section IV.3. The purpose of this section is to clarify the origin, structure, and role of each dataset before model development.

The first dataset was generated in Simulink under a simulated Broken Rotor Bar (BRB) degradation scenario. It represents the controlled experimental part of the work. The second dataset was the XJTU-SY bearing dataset, which was used as a real run-to-failure case study. The main properties of both datasets are summarized in Tables IV.1 and IV.2.

IV.2.1 Simulated BRB Degradation Dataset

The simulated dataset was generated using an induction motor model developed in Simulink. The selected simulated fault was Broken Rotor Bar (BRB) degradation, represented as a progressive degradation process rather than a sudden failure. The recorded signals included rotor speed, electromagnetic torque, and the three-phase stator currents ia, ib, and ic.

This is illustrated in Figure IV.1.

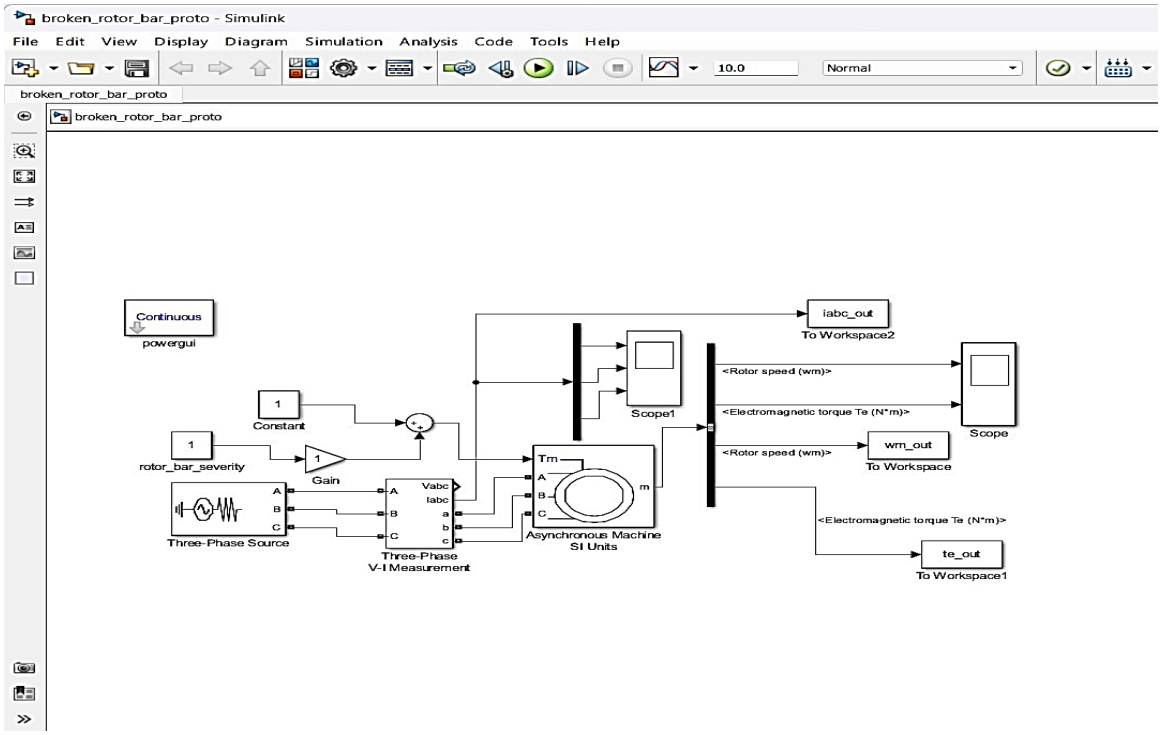


Figure IV.1: Simulink model of the induction motor system used for BRB degradation simulation.

Table IV.1: Summary of the simulated BRB degradation dataset

Item	Description
Simulated system	Simulink induction motor model
Fault type	Broken Rotor Bar degradation
Degradation strategy	Progressive severity levels
Dataset size	60 simulation runs
Operating conditions	A, B, C
Recorded signals	Speed, torque, ia, ib, ic
Target	Normalized RUL
Train/Test split	Train: A+B, Test: C
Models	Random Forest, LSTM

As shown in Table IV.1, Conditions A and B were used for training, while Condition C was reserved for testing. This split was adopted to evaluate the model under an unseen operating condition.

IV.2.2 Real Run-to-Failure Dataset: XJTU-SY

To complement the simulation-based study, the XJTU-SY dataset was used as a real run-to-failure case study. The selected experiments focused on outer race degradation because this fault type is available across multiple operating conditions. The selected bearings are summarized in Table IV.2.

Table IV.2: Selected XJTU-SY bearings used in the real-data RUL experiments

Operating condition	Working condition	Bearings used	Fault type	Role in the study
Condition 1	35 Hz / 12 kN	Bearing1_1, Bearing1_2, Bearing1_3	Outer race	Baseline development and testing
Condition 2	37.5 Hz / 11 kN	Bearing2_2, Bearing2_4, Bearing2_5	Outer race	Cross-condition generalization testing
Condition 3	40 Hz / 10 kN	Bearing3_1, Bearing3_5	Outer race	Difficult generalization analysis

Compared with the controlled BRB simulation, the XJTU-SY data provide a more realistic validation case. The main implications of this difference are discussed in Section IV.6.3 rather than repeated throughout the chapter.

IV.3 Data Preprocessing and Feature Extraction

After describing the adopted datasets in Section IV.2, this section focuses on how the raw signals were transformed into model-ready inputs. The preprocessing stage consisted of organizing the signals in temporal order, extracting representative features, constructing the RUL target, and preparing the data for Random Forest and LSTM models.

The same general processing logic was applied to both datasets. However, the simulated BRB dataset used rotor speed, electromagnetic torque, and three-phase currents, while the XJTU-SY dataset used horizontal and vertical vibration signals.

IV.3.1 Signal Preparation

For the simulated BRB dataset, the simulation runs were organized according to operating condition, repetition number, and degradation severity. For the XJTU-SY dataset, the CSV files of each bearing were sorted chronologically to preserve the run-to-failure order. This temporal organization was necessary before feature extraction and RUL target construction.

The training and testing split was performed at the level of complete operating conditions or complete bearing sequences rather than by randomly splitting individual samples. This split helps prevent data leakage and allows the models to be tested under unseen operating conditions.

IV.3.2 Feature Extraction

Feature extraction was applied to convert each raw signal into a compact set of numerical indicators. See Table IV.3 for the complete list of extracted feature groups used in this study. For the simulated BRB case, current imbalance features were added because rotor-related faults can introduce asymmetry between the phase currents.

Table IV.3: Extracted features used for RUL prediction

Feature group	Features	Applied signals
Basic time-domain features	Mean, standard deviation, RMS, maximum, minimum, peak-to-peak	Vibration, speed, torque, current
Statistical shape features	Skewness, kurtosis	Vibration, speed, torque, current
Vibration indicators	Crest factor, shape factor, impulse factor, margin factor	XJTU-SY vibration signals
Current imbalance features	RMS spread, RMS standard deviation, imbalance ratio, phase-to-phase RMS differences	Simulated BRB currents

IV.3.2.1 Mathematical Definition of Selected Features

To make the feature extraction process more explicit, this subsection presents the mathematical definition of the main features used in this study. For a signal window $x = [x1, x2, \ldots, xN]$, where N is the number of samples in the window, the following time-domain features were computed.

$$Mean = \left(\frac{1}{N}\right) \times \Sigma\, xi$$

$$Standard\ Deviation = sqrt\left(\left(\frac{1}{N}\right) \times \Sigma\, (xi - \mu)^2\right)$$

$$RMS = sqrt\left(\left(\frac{1}{N}\right) \times \Sigma\, xi^2\right)$$

$$Peak - to - Peak = \max(x) - \min(x)$$

$$Skewness = \left(\frac{1}{N}\right) \times \Sigma \left(\frac{(xi - \mu)}{\sigma}\right)^3$$

$$Kurtosis = \left(\frac{1}{N}\right) \times \Sigma \left(\frac{(xi - \mu)}{\sigma}\right)^4$$

where μ represents the mean value of the signal window and σ represents its standard deviation. These features describe amplitude, energy, dispersion, asymmetry, and impulsiveness.

For the vibration signals in the XJTU-SY dataset, additional indicators were used to describe impulsive degradation behavior:

$$Crest\ Factor = max(|x|) \,/\, RMS$$

$$Shape\ Factor = RMS \,/\, mean(|x|)$$

$$Impulse\ Factor = max(|x|) \,/\, mean(|x|)$$

$$Margin\ Factor = max(|x|) \,/\, (mean(sqrt(|x|))^2$$

These vibration indicators are useful because bearing degradation often produces impulsive components and changes in signal amplitude distribution.

IV.3.2.2 Current Imbalance Features for BRB

For the simulated BRB dataset, additional current imbalance features were computed from the three-phase currents. Let $RMSa, RMSb$, and $RMSc$ represent the RMS values of ia, ib, and ic, respectively.

$$RMS\ Spread\ =\ max(RMSa, RMSb, RMSc)\ -\ min(RMSa, RMSb, RMSc)\ RMS\ Standard$$

$$Deviation\ =\ std(RMSa, RMSb, RMSc)$$

$$Mean\ RMS\ =\ (RMSa\ +\ RMSb\ +\ RMSc)\ /\ 3$$

$$Imbalance\ Ratio\ =\ RMS\ Spread\ /\ Mean\ RMS$$

$$Phase-to-Phase\ RMS\ Differences\ =\ |RMSa\ -\ RMSb|, |RMSa\ -\ RMSc|, |RMSb\ -\ RMSc|$$

The RMS spread measures the amplitude difference between the strongest and weakest phase currents. The RMS standard deviation measures the dispersion of the three phase RMS values. The imbalance ratio normalizes the RMS spread by the mean RMS value, and the phase-to-phase RMS differences quantify pairwise current imbalance.

IV.3.2.3 Feature Selection Justification

The selected features were mainly based on time-domain statistical indicators because they are simple, interpretable, and suitable for both simulated motor signals and real vibration signals. RMS reflects signal energy, standard deviation and peak-to-peak describe variability, and skewness and kurtosis help detect changes in distribution shape and impulsiveness.

For the XJTU-SY dataset, vibration-specific indicators were used to represent bearing degradation behavior. For the simulated BRB dataset, current imbalance features were introduced because a broken rotor bar can affect the symmetry of the three-phase currents. Frequency-domain features were not adopted as the main feature set in this chapter in order to keep the pipeline simple and comparable; they remain a possible future improvement.

IV.3.3 RUL Target Construction

Since this study focuses on Remaining Useful Life prediction, the output variable is continuous rather than categorical. Therefore, the target construction step was essential for transforming the degradation process into a supervised regression problem.

For the simulated BRB dataset, the RUL target was built directly from the predefined severity levels. The healthy state was assigned a normalized RUL close to 1.0, while the near-failure state was assigned a normalized RUL close to 0.0.

For the XJTU-SY dataset, the RUL target was constructed from the run-to-failure sequence of each bearing. The earliest file corresponds to the highest remaining life, while the

final file corresponds to failure and was assigned an RUL value of zero. Because the selected bearings have different run-to-failure lengths, normalized RUL was later used to scale each sequence between 1.0 and 0.0. This made degradation trajectories more comparable across bearings and operating conditions.

IV.3.4 Data Normalization and Model Input Preparation

For Random Forest, the extracted features were used directly as tabular inputs. For LSTM, the data were arranged as feature sequences. In the simulated BRB case, each sequence represented the evolution of features across degradation severity levels under a given operating condition and repetition.

As explained in Section IV.3.3, normalized RUL was especially important for the real-data experiments because bearing lifetimes were not identical. The effect of this choice is reported in Section IV.6.2.

IV.3.5 Data Processing Pipeline

Building upon the previous preprocessing steps, the overall workflow used in this study can be summarized as follows:

1. Raw signals
2. Signal ordering and simulation-run organization
3. Feature extraction
4. RUL target construction
5. Normalization
6. Train/Test split
7. Model training
8. Prediction and evaluation

This pipeline ensured that both datasets followed the same supervised learning structure: input features extracted from degradation signals and output labels represented by RUL values.

IV.4 RUL Prediction Models

Building upon the data preprocessing steps presented earlier, two model families were developed for RUL estimation: Random Forest Regressor and Long Short-Term Memory (LSTM). The first model provides a strong feature-based baseline, while the second investigates whether temporal modeling can improve prediction when sequences are available.

IV.4.1 Random Forest Regressor

Random Forest Regressor was used as the main baseline model. It is an ensemble learning method based on multiple decision trees, where the final prediction is obtained by averaging the predictions of all trees. In this work, Random Forest was applied to the extracted feature tables.

For the simulated BRB dataset, Random Forest was first trained using speed and torque features only, then with current features, and finally with current imbalance indicators. This progressive setup helped evaluate how each feature group contributed to prediction performance.

IV.4.2 Long Short-Term Memory Network

Long Short-Term Memory (LSTM) is a recurrent neural network architecture designed to model sequential data. Unlike feature-based models that treat samples independently, LSTM can learn dependencies between consecutive time steps. This makes it suitable for degradation modeling when the temporal evolution of features is informative.

In the simulated BRB experiment, the LSTM model was trained using sequences constructed from the degradation progression. In the real XJTU-SY experiments, LSTM was mainly used for the more difficult generalization cases in Condition 3.

This is illustrated in Figure IV.2.

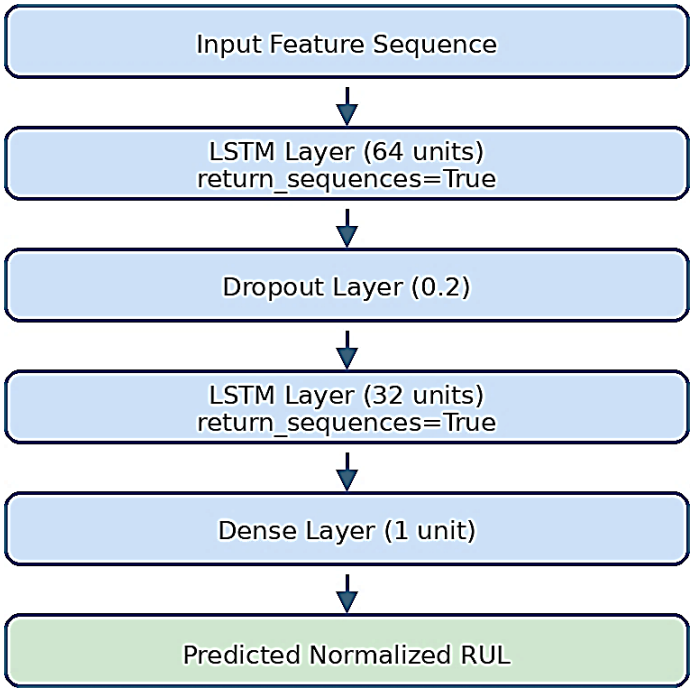


Figure IV.2: Architecture of the LSTM model used for sequence-based RUL prediction.

The LSTM architecture receives a sequence of extracted feature vectors as input. The first LSTM layer learns high-level temporal degradation patterns, the dropout layer reduces overfitting, the second LSTM layer refines the sequential representation, and the dense layer outputs the predicted normalized RUL value.

IV.4.3 Model Selection Strategy

Having established a strong baseline with Random Forest, the study further explored LSTM when the degradation process was expected to benefit from sequence-based modeling. This gradual strategy allowed the comparison between a simpler feature-based model and a more complex temporal model.

Table IV.4: Summary of adopted prediction models

Model	Input representation	Main purpose
Random Forest Regressor	Extracted feature table	Baseline RUL regression and feature-based prediction
LSTM	Feature sequences	Temporal degradation modeling and sequence-based RUL prediction

IV.4.4 Training Configuration and Hyperparameters

To improve reproducibility, the main training configurations and model hyperparameters used in this study are summarized in this section. Since the objective was RUL prediction, all models were trained as regression models.

Table IV.5: Random Forest hyperparameters used in this study

Hyperparameter	Value used	Description
N estimators	100	Number of decision trees in the forest
Max depth	None	Trees were expanded until the default stopping criteria were reached
random state	42	Fixed seed used to make the results reproducible
criterion	Squared error	Splitting criterion used for regression
Min samples split	2	Minimum number of samples required to split an internal node
Min samples leaf	1	Minimum number of samples required at a leaf node

The Random Forest setup was intentionally kept simple to maintain it as a clear baseline. A fixed random_state was used for reproducibility, and the remaining values were kept close to common default settings.

Table IV.6: Training configuration and hyperparameters

Model	Input format	Main hyperparameters	Training settings
Random Forest Regressor	Feature table	100 trees,max_depth = Noe random state=42	Trained on selected training bearings/conditions and tested on unseen bearing/condition
BRB LSTM	Feature sequences	LSTM (64) -> Dropout (0.2) -> LSTM (32) -> Dense (1)	Adam optimizer, MSE loss, 60 epochs maximum, batch size = 2, early stopping patience = 6
XJTU-SY LSTM	Feature sequences	Stacked LSTM-based sequence model	Trained on feature sequences from selected bearings and evaluated on unseen test bearings

IV.5 Evaluation Metrics

Since the objective is RUL prediction, the problem is treated as a regression task. Therefore, classification metrics such as accuracy, confusion matrix, precision, recall, and F1-score were not used. The developed models were evaluated using MAE, RMSE, and R^2.

IV.5.1 Mean Absolute Error (MAE)

MAE measures the average absolute difference between the true RUL values and predicted RUL values. A lower MAE indicates more accurate predictions on average.

$$MAE = (1/n) \times \Sigma |yi - \hat{y}i|$$

IV.5.2 Root Mean Squared Error (RMSE)

RMSE measures the square root of the average squared prediction error. It penalizes large errors more strongly than MAE.

$$RMSE = sqrt((1/n) \times \Sigma (yi - \hat{y}i)^2)$$

IV.5.3 Coefficient of Determination (R^2)

R^2 measures how well the model explains the variation in the target RUL values. A value close to 1 indicates good fit, while a negative value indicates poor generalization.

$$R^2 = 1 - [\Sigma(yi - \hat{y}i)^2 / \Sigma(yi - \bar{y})^2]$$

Table IV.7: Evaluation metrics used for RUL prediction models

Metric	Full name	Description	Interpretation
MAE	Mean Absolute Error	Average absolute prediction error	Lower is better
RMSE	Root Mean Squared Error	Square root of average squared error	Lower is better; penalizes large errors
R^2	Coefficient of Determination	Explains variation in RUL target	Higher is better; negative means poor generalization

IV.5.4 Evaluation Procedure

The evaluation procedure was designed to assess prediction accuracy and generalization. The models were not evaluated using random sample splitting. Instead, the test data were complete operating conditions or complete bearing run-to-failure sequences that were not used during training.

For the simulated BRB dataset, the models were trained on Conditions A and B and tested on Condition C. For XJTU-SY, evaluation was performed at the bearing level. Smoothing was applied only after prediction in selected real-data experiments and was not used during training.

IV.6 Results and Discussion

This section presents the results obtained from the simulated BRB dataset and the real XJTU-SY dataset. The discussion follows a result-cause-meaning structure to reduce repetition and focus on the main findings.

IV.6.1 Results on the Simulated BRB Dataset

The simulated BRB dataset was first evaluated using Random Forest. The baseline using speed and torque features achieved $R^2 = 0.4649$. Adding current features improved R^2 to 0.5208, and adding imbalance features improved it further to 0.5381. This progression confirms that current-related features are relevant for rotor-related degradation.

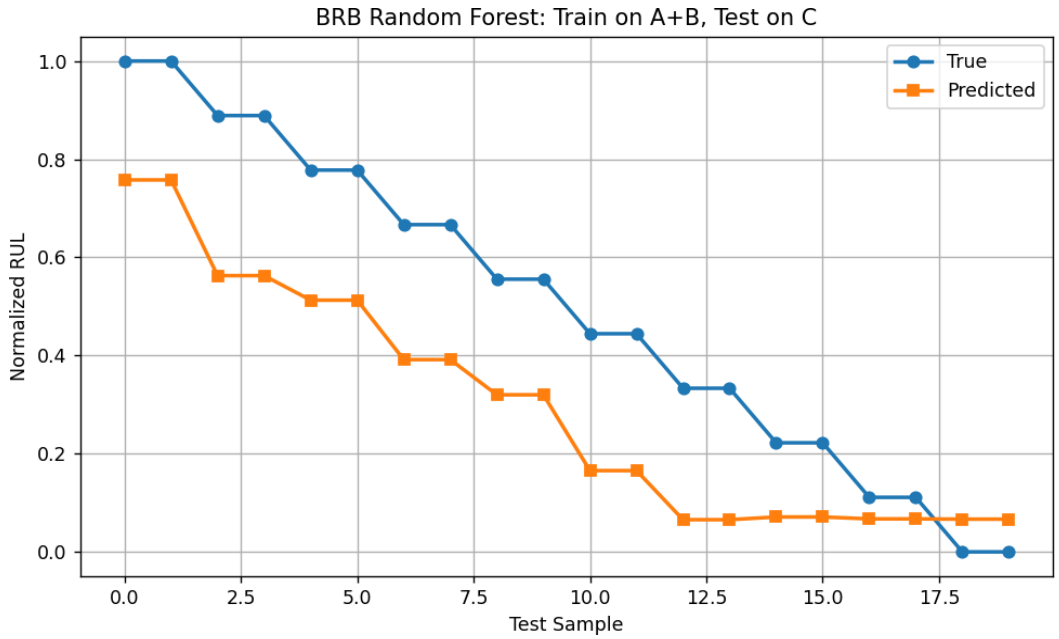


Figure IV.3: Actual and predicted normalized RUL for the simulated BRB dataset using Random Forest.

The best BRB result was obtained using LSTM sequences, with MAE = 0.1269, RMSE = 0.1509, and R^2 = 0.7764. This means that the temporal evolution of the degradation process was better captured when the data were modeled as sequences rather than independent samples.

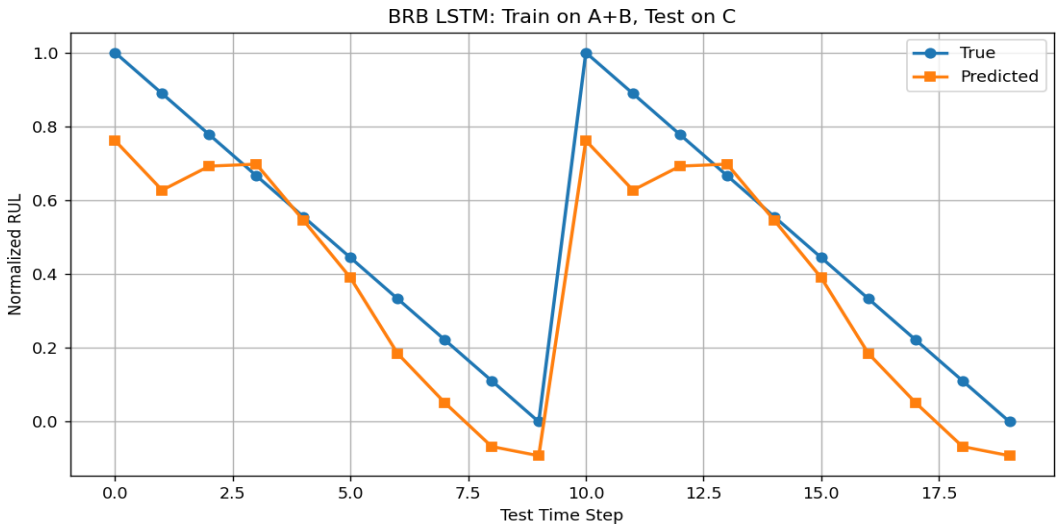


Figure IV.4: Actual and predicted normalized RUL for the simulated BRB dataset using LSTM.



Figure IV.5: Training and validation loss curve of the BRB LSTM model.

Table IV.8: Summary of BRB simulation model performance

Experiment	**Model**	**Features/Input**	**MAE**	**RMSE**	**R^2**
Baseline	Random Forest	Speed + torque	0.2152	0.2335	0.4649
Current features	Random Forest	Speed + torque + currents	0.2019	0.2209	0.5208
Imbalance features	Random Forest	Current + imbalance features	0.1973	0.2169	0.5381
Sequence model	LSTM	Sequential features	0.1269	0.1509	0.7764

IV.6.2 Results on the Real XJTU-SY Dataset

The first real-data experiment was conducted within Condition 1 using Bearing1_1 and Bearing1_2 for training and Bearing1_3 for testing. The Random Forest baseline achieved R^2 = 0.570, and smoothing slightly improved it to 0.586. This result indicates that the model captured the general degradation trend within the same operating condition.

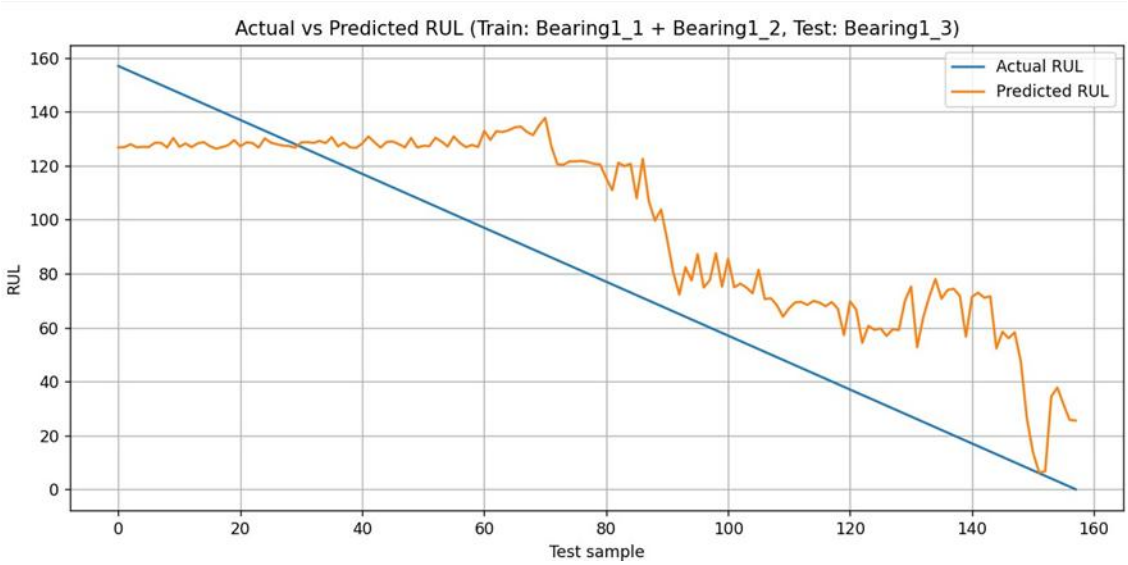


Figure IV.6: Actual and predicted RUL for the XJTU-SY Condition 1 baseline experiment.

For Condition 2, normalized RUL was adopted as explained in Section IV.3.3. The model achieved R^2 values of 0.7238 on Bearing2_2, 0.5736 on Bearing2_4, and 0.5122 on Bearing2_5. This shows that normalized RUL improved comparison across bearings with different lifetimes.

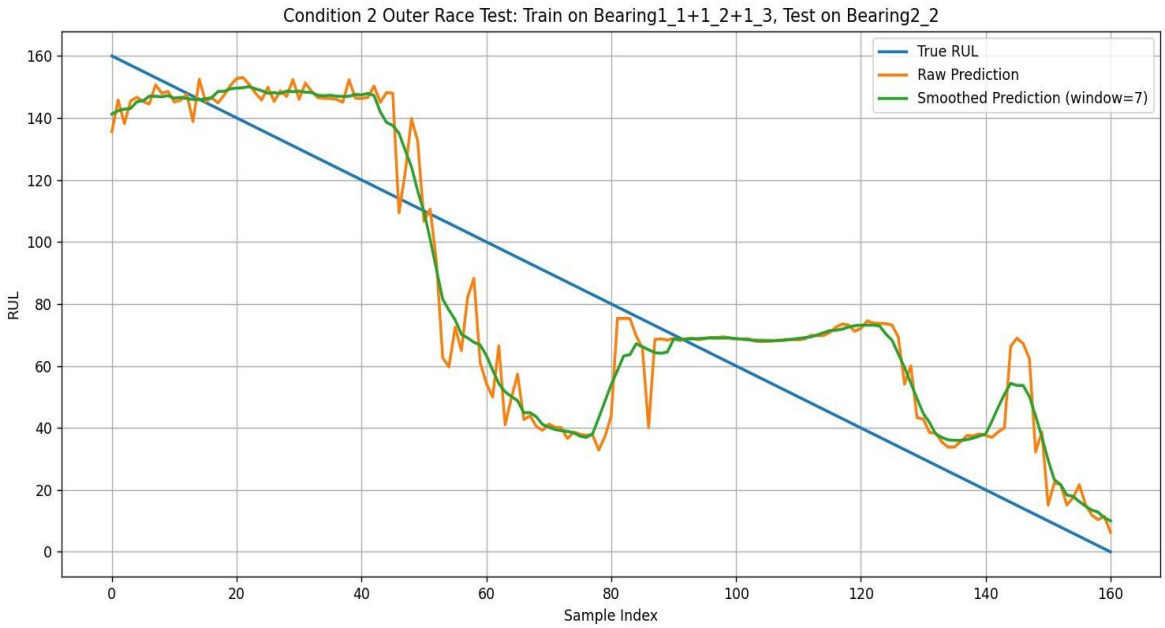


Figure IV.7: Actual, raw predicted, and smoothed predicted RUL for Bearing2_2 under Condition 2.

Condition 3 was more difficult. Random Forest failed to generalize properly on Bearing3_1, while LSTM improved the result to $R^2 \approx 0.347$. However, the same LSTM approach did not generalize well to Bearing3_5, where R^2 remained negative.

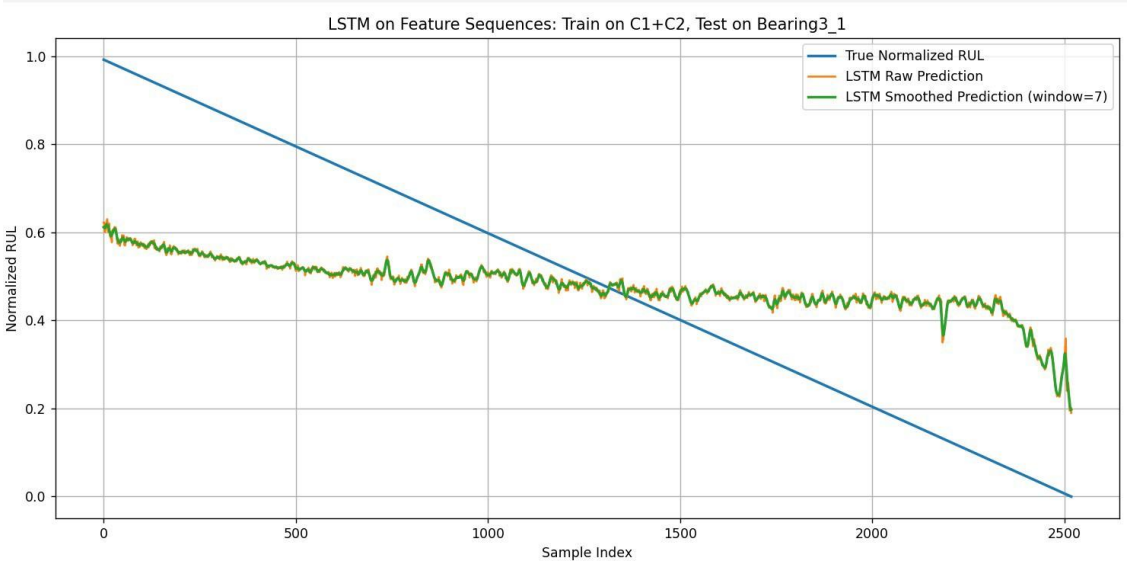


Figure IV.8: Actual and predicted normalized RUL for Bearing3_1 using LSTM.

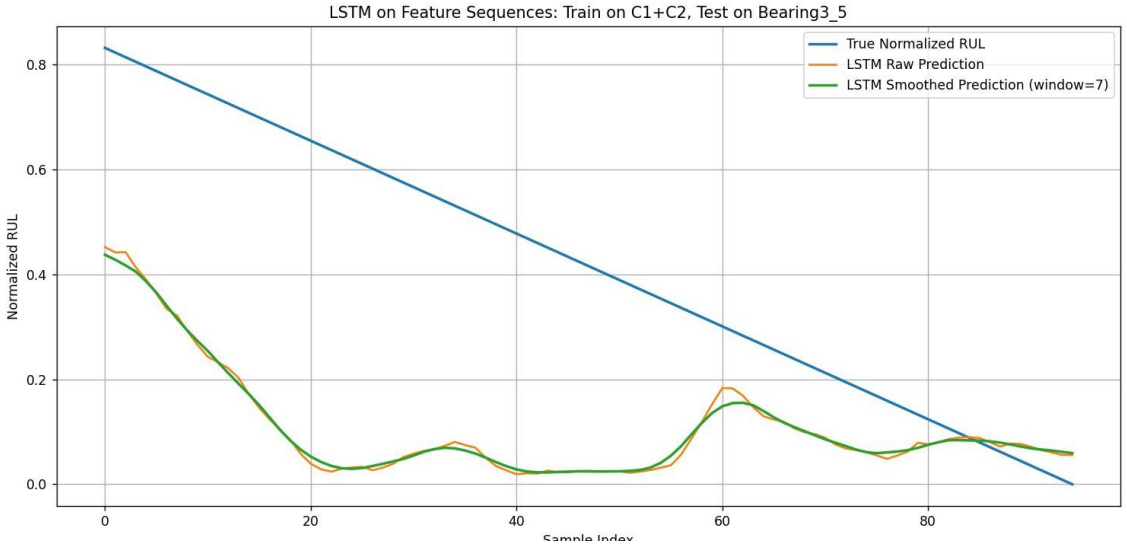


Figure IV.9: Actual and predicted normalized RUL for Bearing3_5 using LSTM.

Table IV.9: Summary of real-data model performance on the XJTU-SY dataset

Dataset part	Train/Test setup	Model	Target	Best R^2	Interpretation
Condition 1 baseline	B1_1 + B1_2 -> B1_3	Random Forest	Raw RUL	0.586	Acceptable baseline
Condition 2	C1 bearings -> B2_2, B2_4, B2_5	Random Forest	Normalized RUL	0.724	Best real-data generalization
Condition 3 / Bearing3_1	C1 + C2 -> B3_1	LSTM	Normalized RUL	0.347	Partial improvement
Condition 3 / Bearing3_5	C1 + C2 -> B3_5	LSTM	Normalized RUL	-0.335	Limited generalization

IV.6.3 General Discussion

The simulated BRB dataset produced clearer results because the degradation severity was controlled. As a result, the models could learn a more systematic degradation pattern. The best simulated result was achieved by LSTM, indicating that BRB degradation benefited from sequence-based modeling.

In contrast to the promising results on the simulated dataset, the real-data experiments revealed stronger generalization challenges. The XJTU-SY results improved when normalized RUL was used, but Condition 3 remained difficult, especially for Bearing3_5. Further discussion on these generalization limits is presented in Section IV.6.3.1.

IV.6.3.1 Failure Analysis and Generalization Limits

The weak performance on Bearing3_5 can be explained by the variability of real bearing degradation. Even when the final fault type is the same, the degradation trajectory may differ due to noise, nonlinear degradation rates, and operating-condition effects. Therefore, the issue is not only model selection, but also the variability of real run-to-failure behavior.

This result confirms that RUL models should be evaluated on complete unseen trajectories rather than randomly selected samples. A good result on one bearing does not necessarily guarantee robust generalization to all bearings.

IV.6.3.2 Accuracy-Complexity Trade-off

Random Forest is simpler and faster to train because it uses extracted feature vectors directly. LSTM is more complex because it requires sequence construction, feature scaling, validation monitoring, and iterative training. In the simulated BRB case, this additional

complexity was justified because LSTM improved R^2 from 0.5381 to 0.7764. However, in the real XJTU-SY experiments, LSTM improved Bearing3_1 but failed on Bearing3_5.

Therefore, model selection should consider not only accuracy, but also complexity, interpretability, data size, and generalization ability.

Table IV.10: Overall summary of the main experimental findings

Part	Best model / setting	Best result	Main conclusion
Simulated BRB	LSTM sequence model	$R^2 = 0.7764$	Temporal modeling captured BRB degradation better than feature-based RF
XJTU-SY Condition 1	Random Forest + smoothing	$R^2 = 0.586$	Acceptable baseline within the same operating condition
XJTU-SY Condition 2	Random Forest + normalized RUL	$R^2 = 0.724$	Normalized RUL improved cross-condition comparison
XJTU-SY Condition 3	LSTM on Bearing3_1	$R^2 = 0.347$	Partial improvement, but generalization remained limited

IV.7 Conclusion

This chapter presented the implementation and evaluation of the proposed RUL prediction approach using both simulated and real industrial datasets. The experimental results showed that simulation data are useful for developing and testing predictive models, while real datasets highlight the practical challenges of machine degradation and model generalization.

The obtained results also showed that current and vibration features play an important role in fault prediction. Random Forest achieved stable prediction performance using statistical features, whereas the LSTM model provided better results for sequential degradation patterns.

Overall, the combination of simulated and real datasets provided a better understanding of the strengths and limitations of the proposed prediction models for predictive maintenance applications.

General conclusion

This work focused on the application of IoT and Artificial Intelligence techniques for predictive maintenance of induction motors. The main objective was to move from traditional maintenance strategies toward a more proactive approach by estimating the Remaining Useful Life (RUL) using a data-driven methodology.

During this study, an IoT-based architecture was proposed and implemented using MATLAB/Simulink simulations together with the real XJTU-SY bearing dataset. The simulation environment provided a controlled framework for studying Broken Rotor Bar (BRB) degradation scenarios, while the real dataset reflected the complexity and variability of real industrial conditions.

The comparative analysis between Random Forest and LSTM models showed that each approach has its advantages. Random Forest achieved good performance using statistical features, whereas the LSTM model was more effective in learning temporal degradation patterns from sequential vibration signals. In addition, the use of Normalized RUL improved the consistency of the prediction process under different operating conditions.

The obtained results demonstrate that combining simulation tools with machine learning and deep learning techniques can provide an effective solution for predictive maintenance applications, even when large industrial datasets are not available. Although the implementation was carried out in a simulated environment, the proposed system can be extended toward real industrial deployment through the integration of physical sensors and IoT communication technologies such as MQTT.

As future work, the system can be improved by implementing real-time communication, testing additional deep learning models, and deploying the prediction models on embedded hardware for edge-based industrial monitoring applications.

References

[1] R. Gulati, *Maintenance and Reliability Best Practices*. New York, NY, USA: Industrial Press, 2013.

[2] T. Zhu, Y. Ran, X. Zhou, and Y. Wen, "A Survey of Predictive Maintenance: Systems, Purposes and Approaches," *IEEE Access*, 2019.

[3] D. Kohler and J.-D. Weisz, "Industry 4.0, an industrial and societal revolution," *Futuribles*, vol. 424, no. 3, pp. 47–68, 2018. Available: https://shs.cairn.info/revue-futuribles-2018-3-page-47?lang=fr

[Accessed: 11-May-2026].

[4] IEEE, "Preparing the Packaging Industry for Industry 4.0," *IEEE Future Directions*, Jan. 2023. Available: https://cmte.ieee.org/futuredirections/tech-policy-ethics/january-2023/preparing-the-packaging-industry-for-industry-4-0

[Accessed: 11-May-2026].

[5] K. Ashton, "That 'Internet of Things' Thing," *RFID Journal*, 2009. Available: https://www.rfidjournal.com/articles/view?4986

[Accessed: 11-May-2026].

[6] L. D. Xu, W. He, and S. Li, "Internet of Things in Industries: A Survey*," IEEE Trans. Ind. Informat*., vol. 10, no. 4, pp. 2233–2243, 2014.

[7] M. Chen, S. Mao, and Y. Liu, "Big Data: A Survey," *Mobile Netw. Appl*., vol. 19, no. 2, pp. 171–209, 2014.

[8] A. K. S. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mech. Syst. Signal Process*., vol. 20, no. 7, pp. 1483–1510, 2006.

[9] ISO, *Petroleum, petrochemical and natural gas industries — Collection and exchange of reliability and maintenance data for equipment*, ISO 14224:2016, 2016. Available: https://www.iso.org/standard/64076.html

[Accessed: 11-May-2026].

[10] R. K. Mobley, *An Introduction to Predictive Maintenance*. Elsevier, 2002.

[11] "Application of IoT-Based Predictive Maintenance in Industrial Machinery: Enhancing Operational Efficiency," *Int. J. Artif. Intell*., vol. 5, no. 6, pp. 2286–2288, 2025.

[12] R. K. Mobley, *Maintenance Engineering Handbook*, 8th ed. New York, NY, USA: McGraw-Hill, 2008.

[13] A. K. S. Jardine and A. H. C. Tsang, *Maintenance, Replacement, and Reliability: Theory and Applications*. Boca Raton, FL, USA: CRC Press, 2013.

[14] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[15] J. Lee, H. A. Kao, and S. Yang, "Service innovation and smart analytics for Industry 4.0 and big data environment," *Procedia CIRP*, vol. 16, pp. 3–8, 2014.

[16] T. M. Mitchell, *Machine Learning*. New York, NY, USA: McGraw-Hill, 1997.

[17] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[18] W. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity*," Bull. Math. Biophys*., 1943.

[19] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[20] S. Haykin, *Neural Networks and Learning Machines*, 3rd ed. Pearson, 2009.

[21] S.Hochreiter and J.Schmidhuber, "Long Short-Term Memory,"*Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[22] A. E. Abd El-Aziz et al., "Machine learning and IoT-based predictive maintenance approach for industrial applications," *Alexandria Eng. J.*, 2023.

[23] A. Aboshosha et al., "IoT-based data-driven predictive maintenance relying on fuzzy system and artificial neural networks," *Sci. Rep.*, vol. 13, 2023.

[24] L. Wang et al., "An augmented reality-assisted prognostics and health management system based on deep learning for IoT-enabled manufacturing," Sensors, vol. 22, no. 17, 2022.

[25] W. Yu et al., "Edge computing-assisted IoT framework with an autoencoder for fault detection in manufacturing predictive maintenance," IEEE Trans. Ind. Informat., vol. 19, no. 2, 2023.

[26] Google, "*Colaboratory,*" 2023. Available: https://colab.research.google.com/ [Accessed: 11-May-2026].

[27] Microsoft, "*Visual Studio Code*," 2023. Available: https://code.visualstudio.com/ [Accessed: 11-May-2026].

[28] Python Software Foundation, "*Python Language Reference*," 2023. Available: https://www.python.org/ [Accessed: 11-May-2026].

[29] W. McKinney, "Data Structures for Statistical Computing in Python," *in Proc. 9th Python Sci. Conf.*, 2010, pp. 51–56.

[30] S. van der Walt, S. C. Colbert, and G. Varoquaux, "The NumPy Array: A Structure for Efficient Numerical Computation," *Comput. Sci. Eng.*, vol. 13, no. 2, pp. 22–30, 2011.

[31] F. Chollet, "*Keras: Deep Learning for Humans*," 2015. Available: https://keras.io/

[Accessed: 11-May-2026].

[32] M. Abadi et al., "TensorFlow: A System for Large-Scale Machine Learning," *in Proc. OSDI*, 2016, pp. 265–283.

[33] R. K. Singh and R. Sharma, "Internet of Things (IoT) in Predictive Maintenance: A Review," *IEEE Internet Things J.*, vol. 8, no. 3, pp. 1245–1260, 2021.

[34] ISO/IEC, *Information technology — Internet of Things (IoT) — Reference Architecture*, ISO/IEC 30141:2018, 2018.

[35] M. Ghofrani and M. A. Hejazi, "The Evolution of IoT-based Sensing Technologies in Industry 4.0," *IEEE Sensors Lett.*, vol. 4, no. 10, pp. 1–4, 2020.

[36] J. Lin et al., "A Survey on Communication Protocols for IoT-Based Smart Environments*," IEEE Access*, vol. 5, pp. 11667–11681, 2017.

[37] P. Sethi and S. R. Sarangi, "Internet of Things: Architectures, Protocols, and Applications," *J. Electr. Comput. Eng.*, 2017.

[38] S. Schmidt and B. Wang, "Cloud-Based Predictive Maintenance Frameworks for Industrial Equipment," *IEEE Trans. Ind. Informat.*, vol. 14, no. 3, pp. 890–901, 2018.

[39] A. E. Fitzgerald, C. Kingsley, and S. D. Umans, *Electric Machinery*, 6th ed. New *York, NY, USA: McGraw-Hill, 2003.*