UNIVERSITY OF MOHAMED BOUDIAF – MSILA

FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
DEPARTMENT OF COMPUTER SCIENCE

In partial fulfillment of the requirements for the degree of

Master in **Computer science**

By

Amira ARIOUA-AI

Imane CHALABI- SIGL

Title of the thesis

FactoryNova: An intelligent Platform for Predictive Maintenance and Energy consumption Optimization in Smart Grid Environment

---

Under the supervision of

Said GADRI

Composition of the jury

Mahmoud Brahimi	University of Msila	President
Said GADRI	University of Msila	Reporter
Salah Guessmia	University of Msila	Examiner

2025-2026

الإهداء

إلى من انحنيا لكي أستقيم... إلى أبي وأمي

إلى أبي... سندي الأول، وظلي الذي احتميت به في كل مراحل حياتي، يا من تشققت يداه لتبقى يدي ممسكة بالقلم، ويا من بذل من عمره وجهده ليصنع لي مستقبلاً أفضل. كنتَ دائماً مصدر القوة والعزيمة، وكنتَ ترى في نجاحي حلماً يستحق كل التضحية. أهديك هذا الإنجاز المتواضع، فهو ثمرة من ثمار تعبك وصبرك وعطائك الذي لا يُقدَّر بثمن.

وإلى أمي... نبض قلبي وسر دعائي، يا من سهرتِ الليالي وشاركتِني لحظات التعب والقلق والأمل، ويا من كانت دعواتها ترافقني في كل خطوة. أمي الحبيبة، إن كل نجاح أصل إليه يحمل بصمتك، وكل إنجاز أحققه هو امتداد لتضحياتك ومحبتك اللامحدودة. أهديك هذا العمل عربون امتنان ومحبة، فلولاكِ لما وصلت إلى ما أنا عليه اليوم.

أبي....أمي....

مهما كتبت من كلمات فلن أوفيكما حقكما، فأنتم أصل الحلم وسبب الوصول، وهذه الشهادة هي قبل كل شيء ثمرة جهودكما وتضحياتكما، أهديها إليكما بكل حب وفخر وامتنان.

الى صديق أبي سالمي هشام:

الذي كان دائمًا سندًا حقيقيًا له ولنا، ووقف معنا في كل الظروف، أُهدي هذا العمل تقديرًا وامتنانًا لمواقفه النبيلة ودعمه الصادق.

إلى إخوتي وأخواتي:

***ياسمين، محمد، توفيق، هارون،** مريم*

إلى من شاركوني تفاصيل الحياة، وساندوني بالكلمة الطيبة والدعاء والتشجيع، وكانوا دائماً مصدر قوة وسعادة في مسيرتي.

أهديكم هذا النجاح وأتمنى أن يكون مصدر فخر لنا جميعاً.

إلى أبناء أختي أمير ولؤي وأختي الصغرى فاطمة الزهراء:

إلى ضحكاتكم البريئة التي كانت تزرع الفرح في قلبي، وإلى محبتكم الصادقة التي كانت تمنحني طاقة للاستمرار، أسأل الله أن يبارك أعماركم، وأن يجعلكم من الناجحين والمتفوقين، وأن أراكم يوماً تحققون أحلامكم وتفخر بكم عائلتكم كما أفخر بكم اليوم.

إلى جميع صديقاتي:

شكراً لكل لحظة جميلة، ولكل كلمة دعم وتحفيز، ولكل ذكرى رافقت سنوات الدراسة وجعلتها أكثر سهولة وجمالاً.

إلى كل أستاذ علمني حرفاً، وإلى كل من مدّ لي يد العون والنصح والتوجيه، وإلى كل من شجعني وآمن بقدراتي من قريب أو من بعيد،

أهديكم هذا العمل المتواضع، راجيةً أن يكون ثمرةً نافعةً لما تعلمته منكم.

وإلى نفسي

التي صبرت واجتهدت وتجاوزت الصعوبات، ولم تستسلم رغم التحديات، أهدي هذا الإنجاز تقديراً لكل الجهد المبذول في سبيل الوصول إلى هذا اليوم.

أميرة عريوة

﴿رَبِّ أَوْزِعْنِي أَنْ أَشْكُرَ نِعْمَتَكَ الَّتِي أَنْعَمْتَ عَلَيَّ وَعَلَى وَالِدَيَّ﴾ [النمل: 19]

إلى والديَّ العزيزين

إلى من كان حبهما ملاذاً، ودعاؤهما سندًا، وعطاؤهما نهرًا لا ينضب؛ إلى من بدأت بهما أجمل فصول حياتي، وتعلّمت على أيديهما

أن للعلم قيمة، وللصبر ثمرة، وللنجاح طريقًا يُعبَّد بالإرادة.

إلى أخويَّ العزيزين

إلى من تقاسمت معهما تفاصيل الأيام، فكانا في لحظات التعب قوةً، وفي لحظات الفرح شركاءَ سعادةٍ لا تُنسى.

إلى عائلتي الكريمة

التي كانت لي وطناً من الطمأنينة، ومصدرًا للمحبة والدعم والثقة.

أهدي هذا العمل المتواضع، لا وفاءً لحقهم فحقهم أعظم من أن توفيه الكلمات، وإنما اعترافاً بجميل أثرهم الذي يرافقني في كل

خطوة، وامتناناً لقلوبٍ منحتني الكثير دون أن تنتظر شيئاً في المقابل.

إليكم جميعًا، يبقى هذا العمل شاهدًا صغيرًا على امتنانٍ كبير لا تكفيه الكلمات، وفضلٍ لا يزول أثره.

شلابي ايمان

Acknowledgments

We would like to express our sincere gratitude to our supervisor, **Mr. Said Gadri**, for his valuable guidance, constructive feedback, and continuous support throughout this work. His expertise and encouragement were essential to the successful completion of this thesis.

We also extend our sincere thanks to **the members of the examination committee** for their time, effort, and valuable evaluation of this research.

Our appreciation goes to the Faculty of Mathematics and Computer Science, especially its **Dean,** for providing a supportive academic environment and the necessary resources to carry out this work.

We would also like to thank **the Business Incubator of the University of M'sila** and all its staff members for their guidance, support, and encouragement throughout the development of this project.

Special thanks are extended to the **Sonelgaz Production Unit of M'sila** for welcoming us during our internship and providing valuable industrial experience that enriched this research.

In particular, we would like to express our sincere appreciation to **Engineer Islam Eddine Arioua** for his supervision, technical guidance, continuous support, and assistance throughout the internship and the development of this project.

Finally, we would like to thank everyone who contributed, directly or indirectly, to the completion of this thesis through academic, technical, or moral support. Their contributions are deeply appreciated.

Table of content:

List of figures

List of tables

General Introduction

The increasing complexity of modern industrial infrastructures and the continuous growth in electricity demand have intensified the need for more reliable, efficient, and intelligent energy production systems. In recent years, Smart Grid technologies have emerged as a promising approach to enhancing the performance of conventional electrical networks through the integration of advanced sensing, communication, and data analytics capabilities. By leveraging the large volumes of operational data generated by equipment, sensors, Supervisory Control and Data Acquisition (SCADA) systems, and Internet of Things (IoT) devices, intelligent solutions can support real-time monitoring, predictive maintenance, fault diagnosis, and energy optimization, thereby improve operational efficiency and reduce maintenance costs.

Despite significant technological advances, many power generation facilities still rely primarily on reactive or preventive maintenance strategies, where interventions are performed only after failures occur or according to fixed schedules. Such approaches often result in unplanned downtime, increased operational costs, reduced equipment availability, and inefficient resource utilization. Moreover, the vast amounts of operational data continuously produced by these facilities remain largely underexploited, limiting the ability of organizations to extract actionable insights and support informed decision-making. These challenges are particularly critical in power generation environments, where equipment reliability and energy efficiency directly affect both operational performance and the stability of the electrical grid.

This work was conducted in collaboration with the M'sila Production Unit of SONALGAZ, one of Algeria's major electricity generation facilities, with an installed capacity of approximately 824 MW. A field investigation carried out during the internship revealed several operational limitations, including the absence of predictive maintenance mechanisms, the limited use of sensor-generated data for decision support, and the lack of intelligent energy management tools. These observations highlighted the need for an integrated artificial intelligence solution capable of transforming raw operational data into valuable knowledge for maintenance and energy management.

Although numerous studies have addressed predictive maintenance, fault diagnosis, and energy forecasting independently, few have integrated these functionalities into a unified framework tailored to real-world power generation environments. Furthermore, limited research has focused on Algerian industrial infrastructures using operational data collected directly from production facilities. This gap between existing research and industrial practice motivated the development of the proposed **FactoryNova** Platform, which combines multiple artificial intelligence techniques within a single decision-support platform designed for industrial applications.

The primary objective of this work is to design and implement an intelligent Platform for predictive maintenance and energy optimization in a Smart Grid-oriented environment. To achieve this objective, three complementary artificial intelligence modules were developed. The first focuses on anomaly detection in gas turbine operational data using a deep Autoencoder architecture trained on real measurements collected with a Testo 350 industrial gas analyzer. The second addresses fault detection and severity classification in rotating machinery through a dual CatBoost-based classification approach applied to vibration signal data. The third is dedicated to electricity demand forecasting and energy optimization using an ensemble learning strategy that combines XGBoost, LightGBM, and CatBoost models. These modules were subsequently integrated into a unified web-based platform, named FactoryNova, providing monitoring, predictive analytics, and decision-support functionalities through a role-based architecture.

The main contribution of this work lies in the development of an AI-driven Platform that combines anomaly detection, fault diagnosis, electricity demand forecasting, and energy management within a single platform. In addition, the study proposes a physics-informed synthetic data generation strategy to address the scarcity of labeled fault data in industrial environments and demonstrates the practical applicability of advanced machine learning and deep learning techniques in a real-world power generation context. By bridging the gap between academic research and industrial requirements, the proposed Platform contributes to enhancing operational reliability, supporting predictive maintenance strategies, and promoting more efficient energy management practices.

The remainder of this thesis is organized as follows:

- **Chapter 1: Industrial Context and Problem Definition:** Presents the host organization, analyzes the existing system, identifies its limitations, and introduces the proposed solution.
- **Chapter 2: General Concepts of Smart Grids, Predictive Maintenance, and Energy Consumption Optimization**: Reviews the theoretical concepts related to Smart Grid architectures, maintenance strategies, energy management, and the role of artificial intelligence in industrial environments.
- **Chapter 3: AI Techniques Applied to Predictive Maintenance and Energy Consumption Optimization**: Discusses the machine learning and deep learning techniques employed in this study, presents the datasets, feature engineering approaches, evaluation metrics, and reviews relevant related work.
- **Chapter 4: Analyse, Design and implementation of the proposed solution:** Describes the adopted methodology, data preparation procedures, model architectures, training processes, experimental setup, and performance evaluation of the developed AI modules.
- **Chapter 5: FactoryNova Platform Implementation and Deployment**: Presents the architecture, functionalities, user interfaces, and integration of the developed modules within the FactoryNova intelligent web platform.
- **Chapter 6: Business model and detailed financial study**: presents the proposed business model and provides a detailed financial analysis, including cost estimation, revenue projections, profitability assessment, and the overall economic feasibility of the project.

- **General Conclusion and Future Work**: Summarizes the main achievements of the study, highlights its scientific and practical contributions, discusses limitations, and proposes directions for future research and development.

Chapter1: Industrial Context and Problem Definition

1. Introduction

This chapter presents the industrial context of our work through the M'sila production unit at SONALGAZ Campany, where the internship was conducted. It aims to provide an overview of the company's structure, activities, and technical environment.

Analyzing this system allows us to identify key challenges, particularly in maintenance and data utilization, which justify the need for intelligent solutions. These observations form the basis of our project, focused on designing a Smart Grid for predictive maintenance and optimized energy consumption.

2. Industrial Context: M'sila Production Unit at SONALGAZ Campany

The M'sila production unit attached to SONALGAZ campany is a major electrical power generation facility that plays an important role in supplying energy to the national grid. It is composed of two main production sites: the M'sila site and the Bordj Bou Arreridj (BBA) site.

The M'sila site, created in 1981, extends over an area of approximately 63 hectares. It includes several gas turbine (TG) units with different capacities:

- 2 units of 23 MW
- 3 units of 100 MW
- 2 units of 215 MW

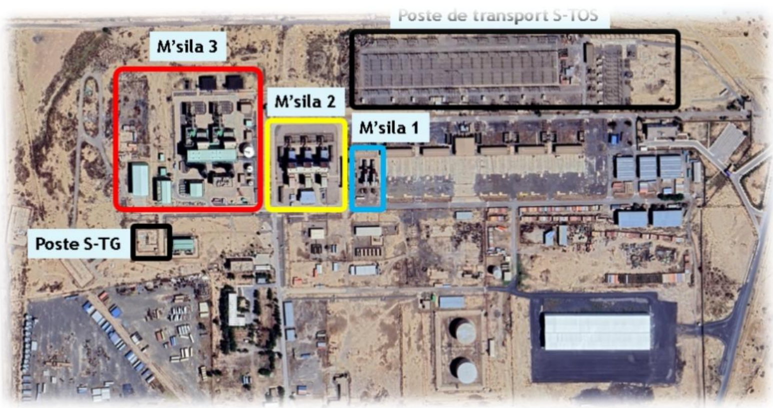


Figure 1.1. Main components of M'sila production Unit

These units ensure a significant portion of the total production capacity of the plant.

The Bordj Bou Arreridj site, commissioned in 2018, includes mobile gas turbines (TGM) with a capacity of 24 MW per unit. This site complements the production capabilities of the main plant and contributes to operational flexibility.

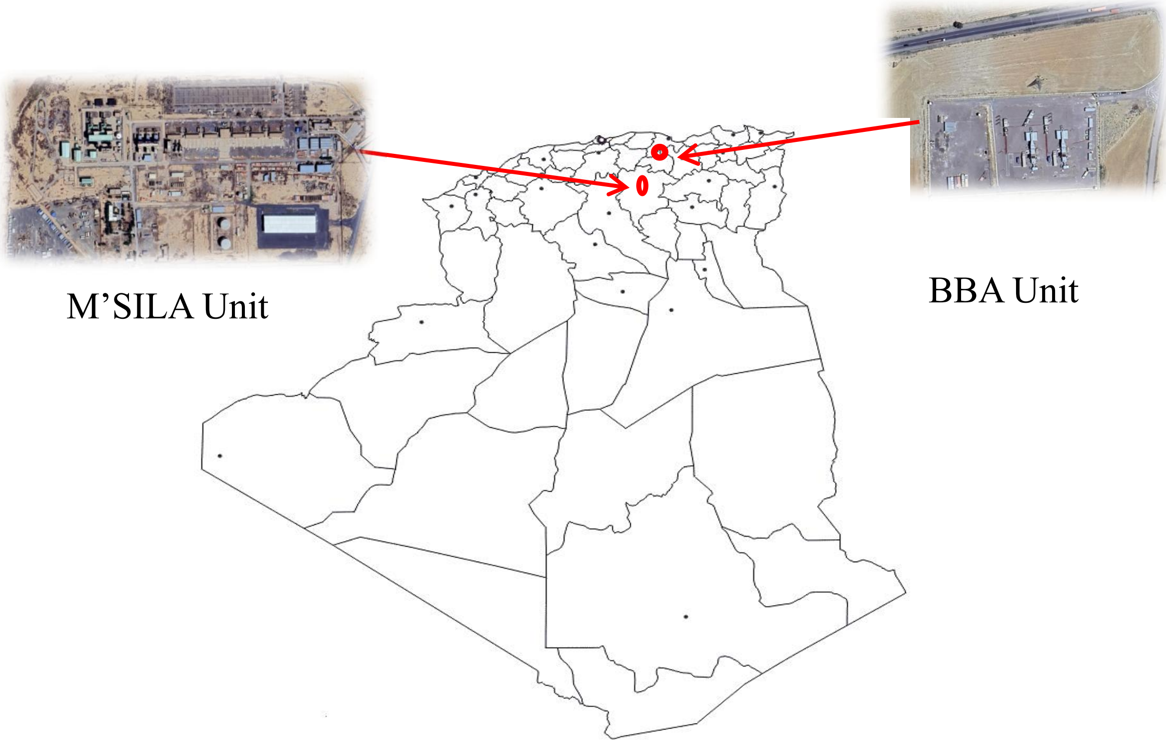


Figure 1.2. Geographical location of M'sila and BBA production Units

The total installed capacity of the unit is approximately 824 MW, making it a key contributor to the national electrical network.

The main objectives of the unit are:

- Meeting the increasing demand for electrical energy
- Ensuring the stability and security of the national grid
- Guaranteeing service continuity and reliability. [1]

Figure 1.3. Close-up Views of Main Components in M'sila Production Unit

2. Organizational Details:

The organizational chart represents the structure of the Single-Site Gas Turbine Production Unit, which is managed by a central administration supervising several divisions and departments.

The Direction of the Single-Site GT Unit is responsible for the overall management of the unit and the coordination of different divisions and departments. It includes 77 positions and ensures the smooth operation of all activities.

The Secretariat plays an administrative role by organizing correspondence, meetings, and document management. This is a unique position within the structure.

The Support and Monitoring Department assists the administration in various areas, including management, industrial safety, and information systems. It consists of 4 positions.

The Human Resources Department is responsible for personnel management, recruitment, and employee training. It includes 3 positions.

The Finance and Accounting Department manages the financial and accounting aspects of the unit. It is structured into several subdivisions and totals 5 positions.

The General Services Department is responsible for general services, logistical support, and infrastructure. It includes 5 positions. [2]

2.1. Roles of the Requested Divisions

- **Operations Division:** This division is responsible for the operation of gas turbines. Its main role is to ensure production continuity by monitoring and optimizing operations. It also ensures the implementation of safety and performance standards. This division includes 31 positions.
- **Maintenance Division:** This division is responsible for equipment maintenance and repair to ensure proper functioning. Its objective is to minimize breakdowns and ensure the reliability of installations. This division includes 27 positions.
- **General Services Department:** This department provides the general services necessary for the proper functioning of the unit. It includes tasks related to safety, cleanliness, and logistical support, ensuring an optimal working environment for all employees. It includes 5 positions. [2]

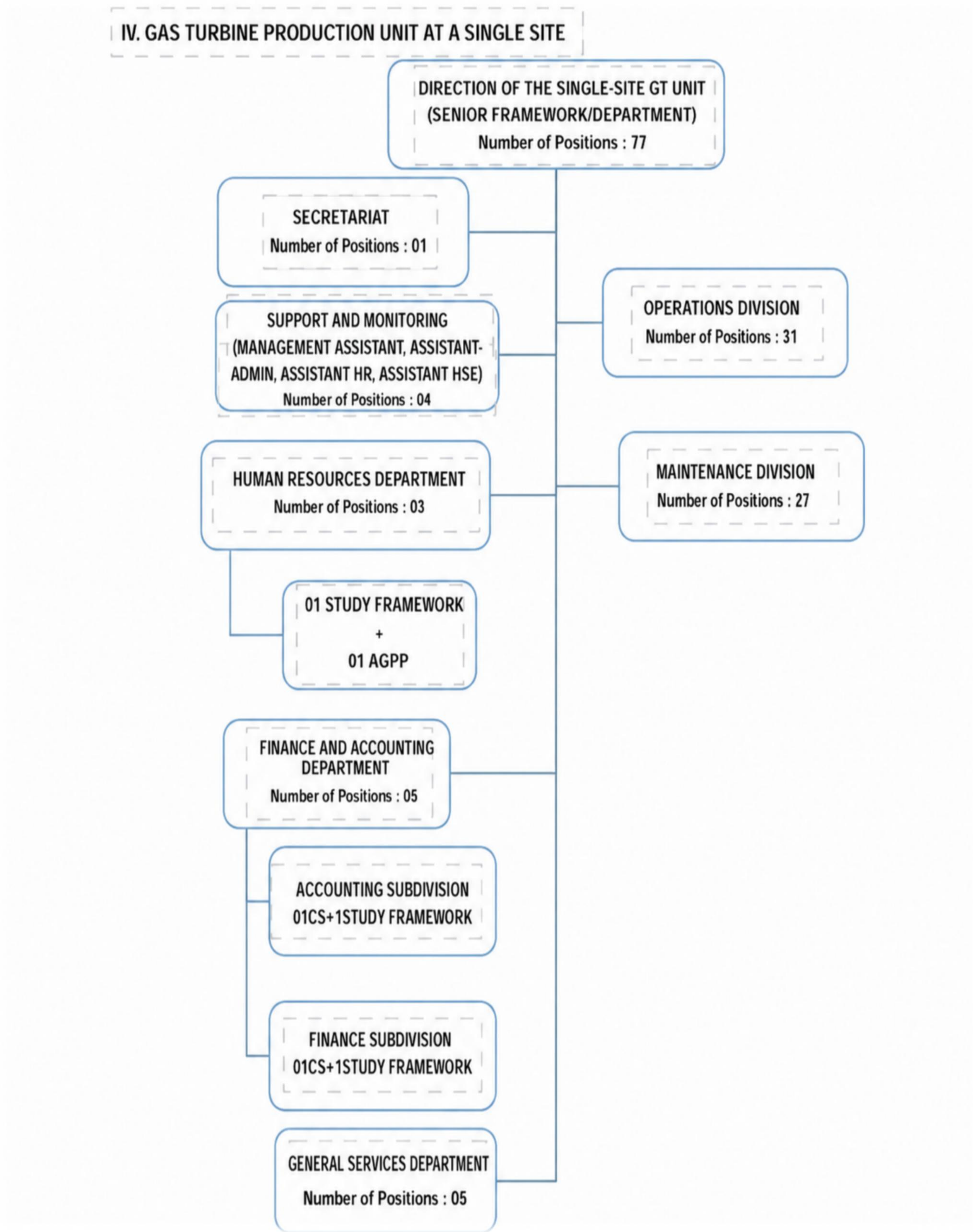


Figure 1.4. Administrative and Operational Structure of the M'sila Production Unit.

3. Existing System Analysis

The existing system is based on a conventional industrial architecture commonly used in power generation plants.

Electricity production relies primarily on gas turbines (TG and TGM), which convert fuel energy into electrical energy.

The production process can be summarized as follows:

- Fuel combustion generates mechanical energy
- Turbines convert mechanical energy into electrical energy
- Generated electricity is transmitted to the grid

The system is monitored using supervision tools such as SCADA (Supervisory Control and Data Acquisition). These systems provide real-time data on various operational parameters, including:

- Temperature
- Pressure
- Vibration
- Load conditions

This data is essential for monitoring the health of the equipment and ensuring safe operation.

Energy is transmitted to the national grid through a high-voltage transmission system (220 kV), allowing distribution to multiple regions. [2]

3.1. SCADA architecture / data flow:

SCADA (Supervisory Control and Data Acquisition) is a centralized industrial control system used to monitor, collect, and control operational data in real time. In traditional power networks, SCADA ensures system reliability, operational continuity, and remote supervision of critical electrical equipment.

3.1.1. Main Components of SCADA Architecture

- **Field Devices:** Include sensors, meters, and actuators installed on physical equipment such as turbines, transformers, and circuit breakers. Their role is to collect operational data such as temperature, voltage, current, and pressure.
- **PLC / RTU Layer**
  - PLC (Programmable Logic Controller): performs local automation and fast control operations.
  - RTU (Remote Terminal Unit): collects data from remote locations and transmits it to the control center.
- **Communication Network:** Ensures data transmission between field devices and the central system using industrial protocols such as Modbus, DNP3, and OPC.
- **SCADA Server (Master Station):** Central unit responsible for:
  - data collection and storage
  - alarm management
  - event logging
  - system monitoring
- **HMI (Human-Machine Interface):** Provides operators with real-time visualization of the system and allows remote control actions. [2]

3.1.2. Data Flow Process

- **Data Acquisition:** Sensors continuously collect data from field equipment.
- **Local Processing:** PLCs and RTUs process and validate collected signals.
- **Data Transmission:** Information is transmitted to the SCADA server through the communication network.
- **Monitoring and Analysis:** The SCADA server stores, analyzes, and visualizes operational data.
- **Control Execution:** Operators send commands through the HMI, which are transmitted back to field devices for execution.

Despite the presence of monitoring systems, the operational logic remains largely reactive, especially in terms of maintenance management. The system focuses on detecting issues rather than anticipating them. [2]

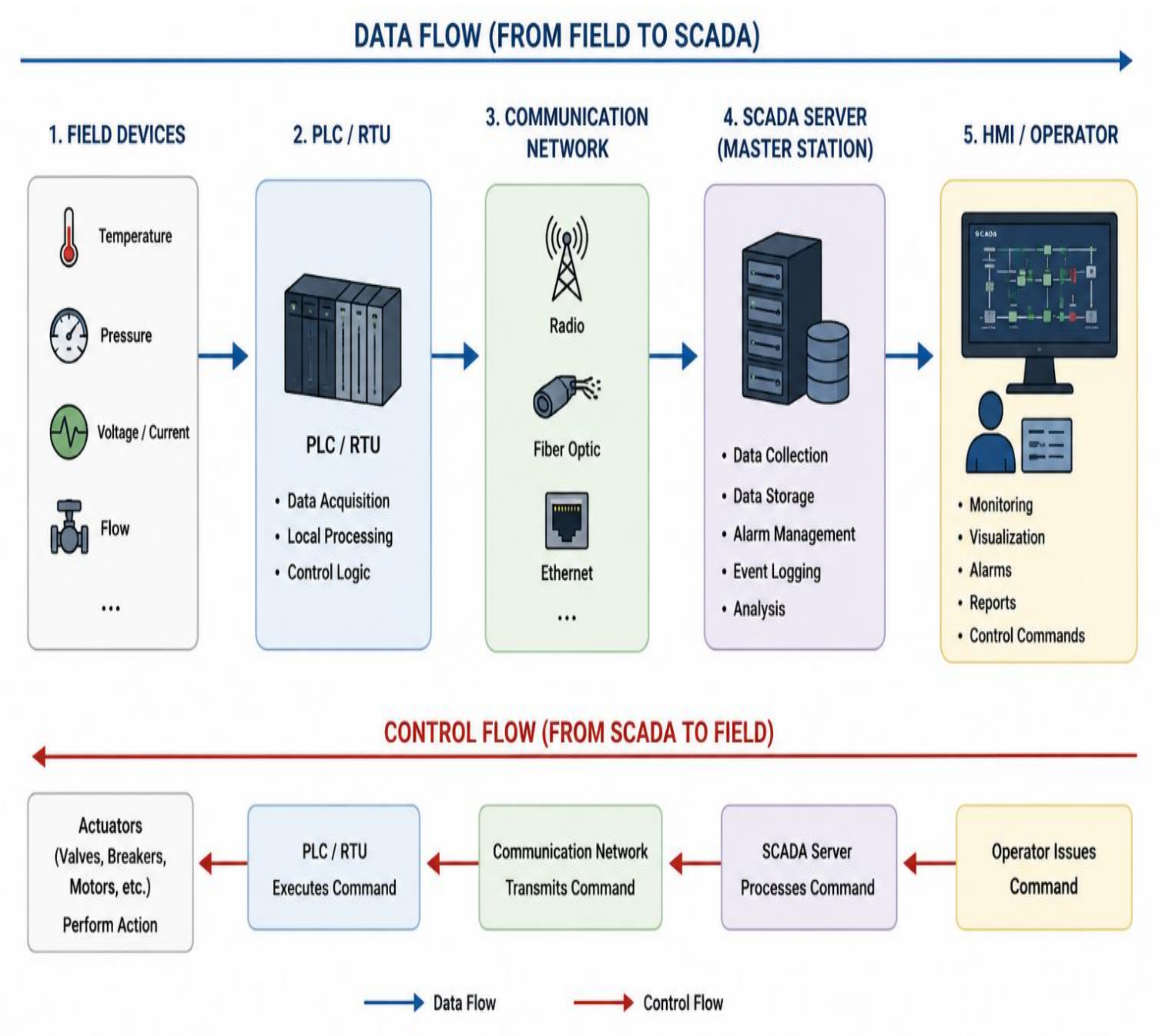


Figure 1.5. SCADA Data Flow

4. Problem Definition

4.1. Core Technical Problem

The main limitation of the current system lies in the lack of integration of intelligent, data-driven mechanisms for maintenance and energy management.

More specifically, the system relies on a corrective maintenance strategy, where interventions are performed only after the occurrence of failures. This approach does not allow anticipating faults or optimizing system performance.

This limitation represents a major gap compared to modern Smart Grid systems, which are designed to integrate:

- Real-time data processing
- Predictive analytics
- Intelligent decision-making

4.2. Identified Problems

Based on the analysis of the existing system, several key issues can be identified:

- **Reactive Maintenance Strategy:**

The current maintenance strategy is mainly reactive, meaning that maintenance operations are initiated only after faults or failures are detected. This approach increases system downtime, reduces operational reliability, and often causes delays in technical intervention, which may negatively impact production continuity.

- **Lack of Artificial Intelligence Integration:**

Although the system continuously generates large volumes of industrial and operational data, these data are not integrated with artificial intelligence or machine learning techniques. As a result, the available information is not exploited for predictive analysis, fault anticipation, or process optimization.

- **Limited Data Utilization:**

The collected data, including sensor measurements, system logs, and operational history, remain insufficiently utilized. This limits the ability to identify hidden patterns, analyze trends, and forecast future system behavior, leading to missed opportunities for performance improvement.

- **Absence of Predictive Decision Support:**

Decision-making within the existing system is primarily manual and largely dependent on operator expertise and human judgment. This absence of predictive decision support reduces the system's capacity to efficiently manage complex operational situations and respond proactively to emerging issues.

- **Suboptimal Energy Management:**

The current system lacks intelligent mechanisms for energy optimization and dynamic production adjustment. Consequently, energy consumption is not efficiently managed, resulting in suboptimal resource utilization and reduced overall system performance.

5. Limitations and Weaknesses of the Current System

The previously identified problems result in several important limitations:

- Strong dependence on reactive maintenance, increasing operational risks
- High maintenance costs due to emergency interventions
- Unplanned downtime affecting productivity and system reliability
- Lack of intelligent automation and advanced decision-support tools
- Poor exploitation of both historical and real-time data

These weaknesses highlight a clear gap between the current system and the expected functionalities of a Smart Grid system, which requires intelligence, adaptability, and data-driven operation.

6. The Proposed Solution

To address these limitations, our project entitled:

"FactoryNova: An intelligent Platform for Predictive Maintenance and Energy Optimization in Smart Grid Environment"

aims to transform the existing system into a smart, data-driven framework.

The objective is to shift from a reactive system to a predictive and optimized system, in line with Smart Grid concepts.

6.1. Our Approach

The proposed solution is based on integrating Artificial Intelligence techniques into the existing industrial system in order to improve maintenance efficiency, operational reliability, and energy management.

- **Predictive Maintenance:**

The proposed system integrates machine learning techniques to predict potential equipment failures and identify various fault categories.

By enabling proactive maintenance strategies, the system reduces operational downtime and enhances the overall reliability and stability of the infrastructure.

- **Anomaly Detection:**

Advanced anomaly detection methods are employed to continuously analyze sensor data and detect abnormal operational behaviors.

These techniques facilitate the early identification of system degradation, enabling timely corrective actions before minor anomalies escalate into critical failures.

- **Energy Consumption Optimization:**

The system analyzes energy consumption patterns to improve resource allocation and optimize operational performance.

This approach contributes to reducing energy inefficiencies, minimizing unnecessary power consumption, and supporting sustainable energy management.

- **Intelligent Decision Support:**

The integration of Artificial Intelligence provides intelligent decision-support capabilities through automated data analysis, predictive insights, and operational recommendations.

This enhances situational awareness, improves system performance, and enables faster and more accurate decision-making for system operators.

7. Conclusion

The analysis of the existing system highlights a critical need for transitioning toward a Smart Grid architecture. The current limitations, particularly the reliance on reactive maintenance and the lack of intelligent data exploitation, justify the integration of artificial intelligence techniques.

The proposed project provides a comprehensive solution by combining predictive maintenance, anomaly detection, and energy optimization, thus contributing to the development of a more efficient, reliable, and intelligent energy system.

Chapter 2: General Concepts of Smart Grids, Predictive Maintenance, and Energy Consumption Optimization

1. Introduction:

The increasing global demand for energy and the limitations of traditional electrical grids have led to the development of Smart Grids as an advanced and intelligent solution. Unlike conventional systems, Smart Grids integrate information and communication technologies to enable real-time monitoring, bidirectional communication, and more efficient energy management.

The integration of IoT devices, smart meters, and SCADA systems generates large volumes of data that can be effectively analyzed using Artificial Intelligence techniques. These technologies enable important functionalities such as energy consumption optimization, improved grid reliability, and predictive maintenance for early fault detection.

This chapter introduces the fundamental concepts related to Smart Grids, energy consumption optimization, and predictive maintenance, which form the theoretical basis of this study.

2. Smart Grid

2.1. Traditional Grids Vs Smart Grids

Traditional electrical grids rely on a one-way power flow and have limited monitoring and control capabilities, whereas Smart Grids integrate advanced communication, automation, and intelligent technologies to improve efficiency and reliability.The corresponding information is summarized in Table 2.1. [3]

Aspect	Traditional Grid	Smart Grid
Communication	Limited or no communication between utilities and consumers	Two-way communication between all grid components
Power Flow	Unidirectional flow from producers to consumers	Bidirectional flow enabling distributed energy exchange
Monitoring	Manual and periodic monitoring	Real-time monitoring using sensors and smart devices
Fault Detection	Slow fault identification and recovery	Automatic fault detection and self-healing capabilities
Energy Management	Limited optimization and control	Intelligent energy management and demand response
Renewable Energy Integration	Difficult integration of renewable sources	Efficient integration of renewable and distributed energy resources
Reliability	Higher risk of outages and failures	Improved reliability and operational stability
Data Usage	Limited data collection and analysis	Extensive use of data analytics and AI techniques
Operational Efficiency	Lower efficiency with higher losses	Higher efficiency and optimized energy consumption

Table 2. 1. presents a comparison between Traditional Grids and Smart Grids across several key aspects.

2.2. Smart Grid definition

The Smart Grid (SG) is an advanced evolution of the traditional electrical grid that integrates information and communication technologies (ICT) to improve efficiency, reliability, and sustainability. It enables bidirectional communication between utilities and consumers, supporting real-time monitoring, control, and optimization of energy flows.

This infrastructure addresses the limitations of conventional grids by reducing energy losses and enhancing overall system performance and stability. . [4]

2.3. Smart Grid Components

The Smart Grid relies on several interconnected components that enable data acquisition, communication, and intelligent decision-making :

- **Smart Meters and Sensors:** These devices collect real-time data on key electrical parameters such as voltage, current, temperature, and energy consumption, enabling system monitoring and analysis.
- **Internet of Things (IoT):** IoT provides connectivity between distributed devices in the grid, enabling data exchange and supporting automation and real-time communication.
- **SCADA Systems:** SCADA systems monitor and control grid operations using real-time data to ensure stable and efficient performance.
- **Advanced Metering Infrastructure (AMI):** AMI integrates smart meters and communication systems to enable two-way communication between utilities and consumers.
- **Energy Management Systems (EMS) and Distribution Management Systems (DMS):** EMS manages generation and transmission, while DMS focuses on distribution, fault management, and grid optimization.
- **Cybersecurity Infrastructure:** Cybersecurity mechanisms protect the Smart Grid from cyber threats and ensure data integrity and system reliability. [5]

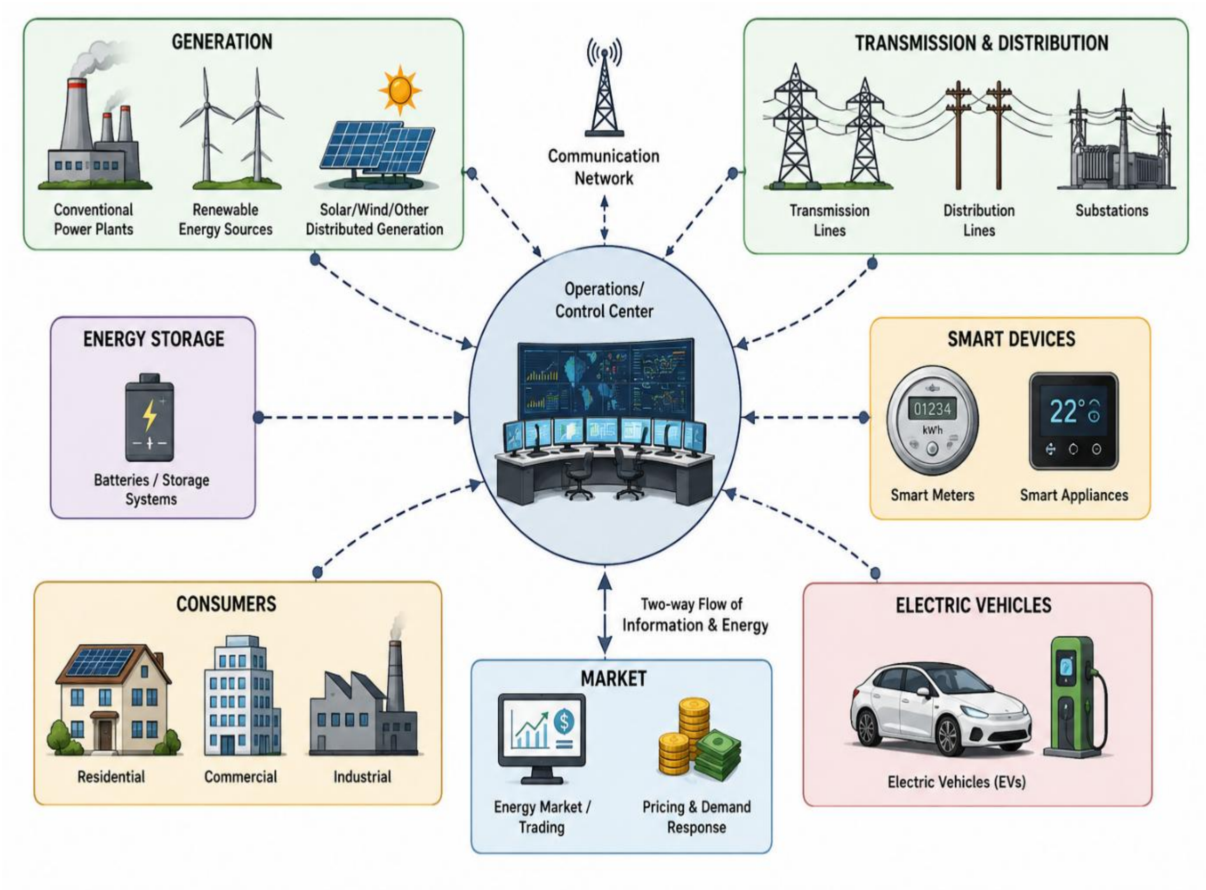


Figure 2.1: Smart Grid Components

2.4. Smart Grid Architecture: Functional Layers

The Smart Grid is organized into five functional layers, each responsible for a specific role in data flow and system operation.

- **Layer 1: Power Layer (Physical Layer):** Represents the physical electricity infrastructure including generation, transmission, and distribution systems.
- **Layer 2: Sensing and Measurement Layer:** Collects real-time operational data using smart meters, sensors, and measurement devices.
- **Layer 3: Communication Layer:** Enables data transmission between grid components using wired and wireless communication technologies.
- **Layer 4: Data Management and Processing Layer:** Handles storage, processing, and analysis of data using SCADA, EMS, DMS, and computing platforms.

- **Layer 5: Application and Intelligence Layer:** Implements advanced applications such as fault detection, forecasting, optimization, and predictive maintenance using AI techniques. [3]

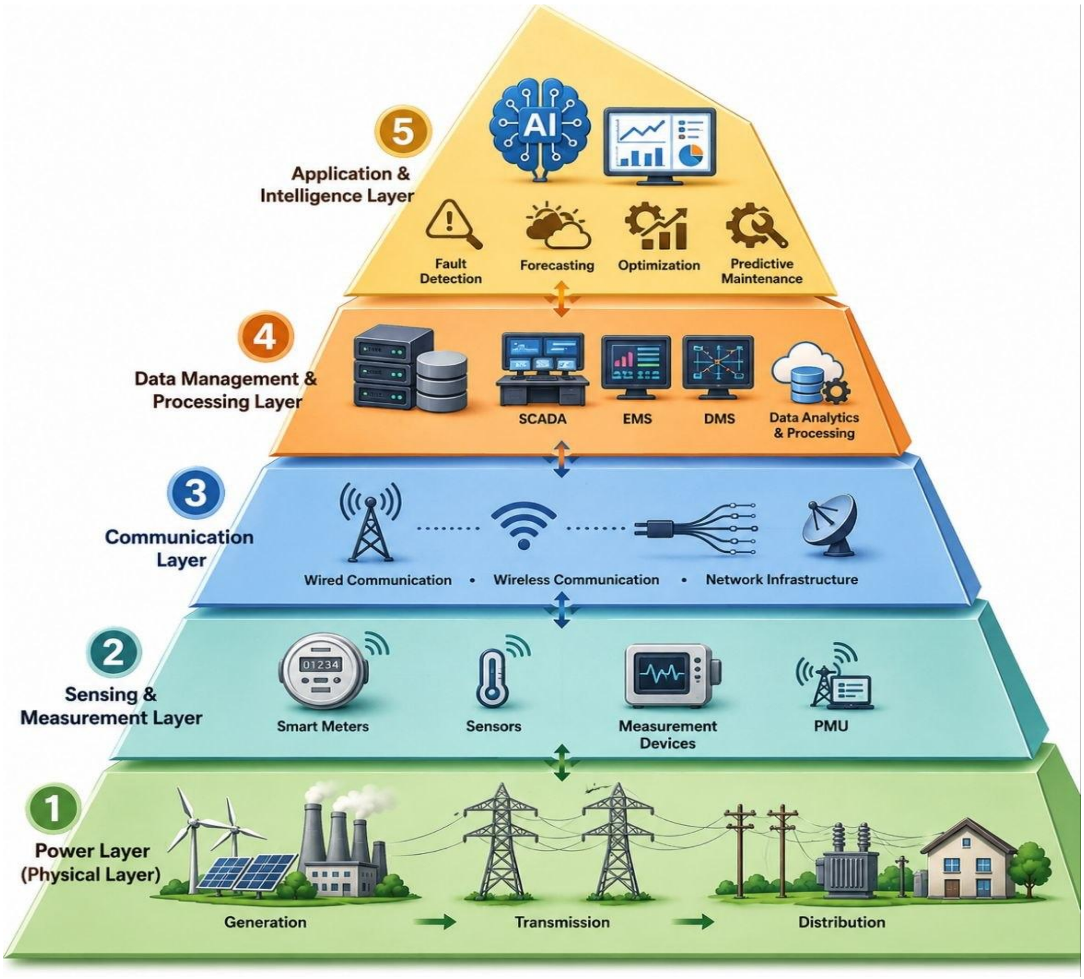


Figure 2.2: Hierarchical Structure of Smart Grid Functional Layers

2.5. Smart Grid Characteristics

The Smart Grid is characterized by several advanced features that distinguish it from traditional grids:

- Bidirectional communication between consumers and utilities
- Real-time monitoring and control

- Automation of grid operations
- Integration of distributed energy resources
- Enhanced reliability and fault tolerance

These features enable the Smart Grid to operate as an intelligent and adaptive system capable of responding dynamically to changes in energy demand and system conditions. In addition, the availability of real-time data and advanced monitoring capabilities provides a strong foundation for implementing intelligent applications such as predictive maintenance.

By leveraging data collected from sensors, smart meters, and SCADA systems, the Smart Grid can support early detection of anomalies and prediction of potential equipment failures. This capability allows utilities to perform maintenance activities proactively, reducing downtime, minimizing operational costs, and improving overall system reliability. [6]

2.6. Artificial Intelligence in Smart Grids

The increasing complexity of modern power systems and the large-scale data generated by smart infrastructure have made Artificial Intelligence an essential enabler for efficient Smart Grid operation. AI enhances system intelligence by supporting data-driven analysis, automation, and adaptive decision-making.

- **Fault Detection and Classification:** AI techniques enable rapid identification and categorization of faults using operational data from grid components. These methods outperform traditional approaches by offering higher accuracy and improved robustness under varying and noisy conditions.
- **Anomaly Detection:** Machine learning models are used to learn normal grid behavior and identify deviations that may indicate faults or abnormal events. This allows early detection of issues such as equipment degradation or cyber-related disturbances.
- **Demand Forecasting:** Intelligent predictive models estimate future electricity consumption by analyzing historical data combined with external factors such as weather and time patterns. This improves planning accuracy and supports efficient grid operation.
- **Predictive Maintenance:** AI-based systems analyze sensor data to predict potential equipment failures before they occur. This condition-based approach reduces downtime, extends asset lifespan, and optimizes maintenance costs.

- **Energy Management and Optimization:** AI techniques optimize energy distribution, storage, and consumption in real time to ensure efficient grid performance. They are particularly effective in managing the variability introduced by renewable energy sources.
- **Renewable Energy Integration:** AI improves the prediction of renewable energy generation by analyzing weather and historical production data. This enhances grid stability and supports better integration of solar and wind energy.
- **Cybersecurity and Intrusion Detection:** Intelligent models monitor network behavior to detect malicious activities and abnormal traffic patterns. This strengthens the resilience of Smart Grids against cyberattacks.
- **Grid Topology and State Estimation:** AI methods estimate the real-time state of the power system even under incomplete or noisy measurements. This improves monitoring accuracy and supports reliable operational decisions. [7]

2.7. Smart Grid Challenges

Despite its advantages, the implementation of Smart Grids faces several challenges:

- Security vulnerabilities due to integration with communication networks
- Privacy concerns related to large-scale data collection
- High implementation and infrastructure costs
- Complexity of managing large and heterogeneous datasets
- Real-time processing requirements for efficient operation. [8]

3. Predictive Maintenance:

3.1. Concept of Maintenance:

Maintenance refers to the set of activities performed to ensure the proper functioning and reliability of equipment and industrial systems. It aims to preserve system performance, prevent unexpected failures, and extend the operational lifetime of machines.

In industrial and energy systems, maintenance is essential due to the increasing complexity of equipment and the need for continuous operation without interruptions. Proper maintenance strategies help reduce downtime, improve safety, and optimize operational costs, especially in critical infrastructures such as power systems and Smart Grids. [9]

3.2. Types of Maintenance:

- **Reactive Maintenance (RM):** Reactive Maintenance is a run-to-failure strategy where maintenance actions are performed only after equipment failure occurs. Although this approach requires minimal planning, it often leads to unexpected downtime, higher repair costs, and reduced system efficiency.

- **Preventive Maintenance (PM):** Preventive Maintenance, also known as scheduled maintenance, involves performing maintenance activities at regular time intervals regardless of the actual condition of the equipment. The objective is to reduce the probability of unexpected failures and ensure continuous operation. However, this approach may result in unnecessary maintenance operations when equipment is still functioning properly.

- **Predictive Maintenance (PdM):** Unlike traditional strategies, Predictive Maintenance relies on real-time data analysis and condition monitoring to anticipate failures before they occur. By using data collected from sensors, smart meters, and monitoring systems, predictive models can detect early warning signs of faults and estimate their future progression. [10]

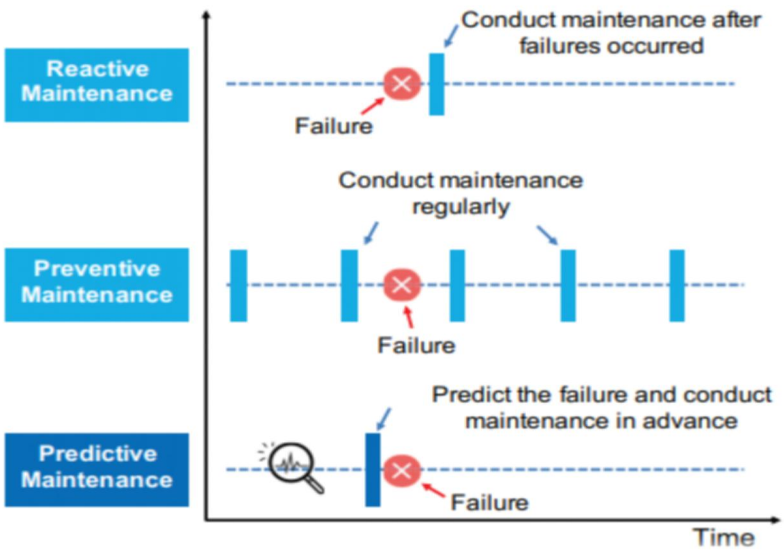


Figure 2.3 : Maintenance Types: Reactive, Preventive, and Predictive Approaches

3.3. Predictive Maintenance Techniques:

Predictive Maintenance relies on data-driven techniques to monitor system conditions and anticipate potential failures before they occur. It is based on continuous analysis of sensor data and system behavior to ensure early fault detection and improved decision-making.

The main techniques used in Predictive Maintenance include:

Detection: identifying abnormal patterns or deviations in system behavior using sensor data.

Isolation: determining the exact component or subsystem responsible for the detected anomaly.

Prediction: estimating the future evolution of faults and predicting possible failures.

Condition Monitoring: continuous observation of equipment parameters such as temperature, vibration, and electrical signals.

Artificial Intelligence and Machine Learning: advanced methods used to analyze large datasets and improve prediction accuracy.

These techniques allow the transition from reactive maintenance to intelligent and proactive maintenance strategies. [11]

3.4. Role of AI in Maintenance:

The integration of AI in maintenance provides several benefits, including:

- Artificial Intelligence (AI) plays a key role in modern maintenance systems by enabling automated analysis of large-scale industrial data.
- Machine Learning techniques are used to detect patterns, identify anomalies, and classify equipment faults.
- In Smart Grid environments, AI processes data collected from sensors, smart meters, and SCADA systems to detect early signs of failure.
- Anomaly detection models help identify abnormal system behavior that may indicate equipment degradation or operational issues.
- AI-based fault classification techniques distinguish between different failure types, supporting appropriate maintenance decisions.

- The integration of AI with monitoring systems improves maintenance accuracy, enhances reliability, reduces downtime, and lowers operational costs. [12]

3.5. The importance of predictive maintenance in smart grids:

Predictive maintenance changes from reactive to proactive, data-driven strategies. This makes energy systems more reliable, efficient, and easier to manage. Several advantages can be cited when using predictive maintenance, including:

- **Proactive Approach:** Uses machine learning and data analytics to find problems before they happen, which lets you fix them quickly and avoid expensive downtime.
- **Minimizing Operational Disruptions**: IoT data finds faults, which lets maintenance happen during times of low demand to keep the supply going. Asset Performance Optimization finds problems, makes things work better, makes equipment last longer, and lowers costs.
- **Resource Allocation Efficiency:** Only targets the assets that are needed, making the best use of labor and resources while cutting down on unnecessary inspections.
- **Improved Grid Resilience:** Predicts weaknesses and makes the grid stronger so it can handle extreme conditions.
- **Customer Satisfaction and Trust:** Builds trust in the service, makes sure it works, and keeps interruptions to a minimum. [13]

4. Energy Consumption Optimization:

4.1. Electrical Energy

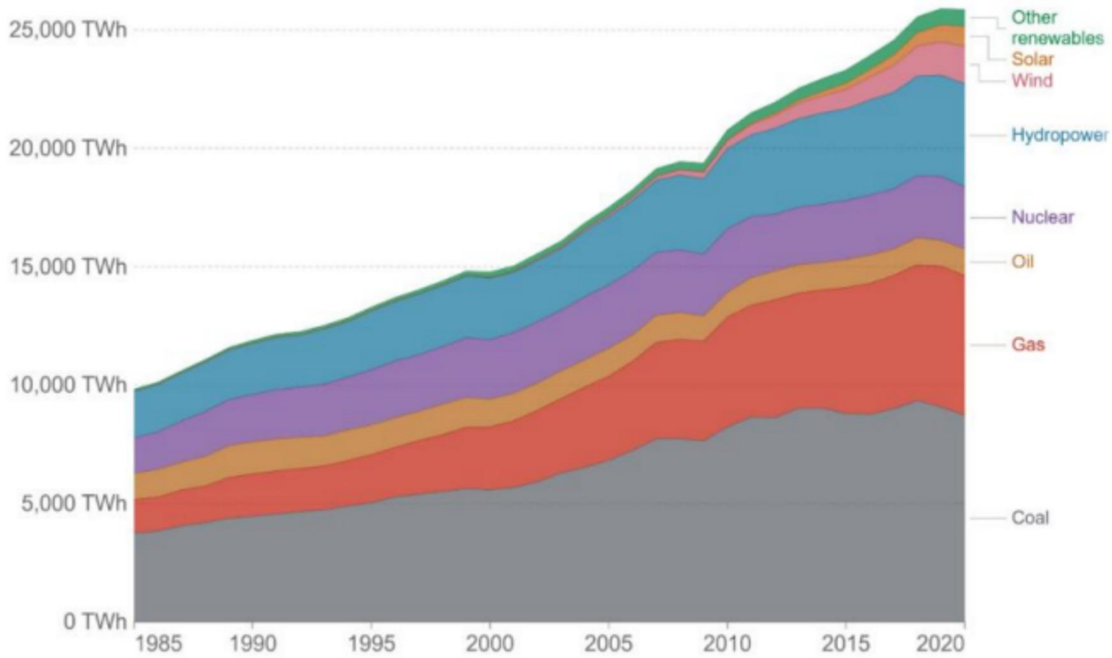
Electrical energy is a form of energy generated from renewable and non-renewable sources. It results from charged particles such as electrons and ions and can produce heat and light. [14]

4.1.1. Methods of Electrical Energy Production:

Electricity can be generated from renewable resources such as wind, solar energy, and tides, as well as from non-renewable resources including fossil fuels and nuclear energy.

- **Hydroelectric Power:** Hydroelectric plants generate electricity by using the energy of flowing or falling water to rotate turbines connected to generators.

- **Steam Power:** Steam power plants produce electricity using high-pressure steam generated in boilers. The steam drives turbines connected to electric generators.
- **Internal Combustion Engines:** Internal combustion plants use liquid fuels burned inside combustion chambers to generate mechanical rotation and produce electricity.
  - **Diesel Power:** Diesel generators are characterized by fast startup and shutdown. They are commonly used during emergencies and peak demand periods, although they consume large amounts of fuel and have limited capacity.
  - **Gas Turbine:** Gas turbines are available in various capacities and are mainly used during peak loads. They operate quickly, require less cooling water, and can use different fuel types, but they have relatively low efficiency and high fuel consumption.
- **Tidal Power:** Tidal power stations convert the energy of ocean tides into electricity using barrages and turbines.
- **Nuclear Power:** Nuclear plants generate electricity through uranium fission, which produces heat to convert water into steam that drives turbines connected to generators.
- **Wind Power:** Wind turbines convert wind energy into electrical energy through the aerodynamic movement of rotor blades connected to generators.
- **Solar Power:** Solar power plants convert sunlight into electricity using photovoltaic panels or concentrated solar radiation systems. The produced energy can also be stored in batteries or thermal storage systems.



The following figure shows electricity production in the world by energy source:

Figure 2.4: Electricity production in the world by energy sources [15]

Electrical energy production relies on multiple sources, and continuous research is being conducted to develop advanced generation methods capable of meeting the increasing energy demands of different sectors. [15]

4.2. Energy Consumption Concept:

Energy consumption refers to the amount of electrical energy used by different systems, sectors, or devices over a specific period. It represents a key indicator for evaluating energy demand and system performance.

Electricity is widely used across multiple sectors, including residential, industrial, agricultural, and public services. The increasing dependence on electrical energy is mainly driven by population growth, technological development, and economic expansion.

4.2.1. Energy Consumption Issues:

Many issues face Energy consumption, including:

- Modern power systems face increasing electricity demand due to population growth, industrial activities, and technological advancement.

- High energy demand can create peak load conditions, leading to system stress and potential instability.
- Transmission and distribution losses reduce energy efficiency and increase operational costs.
- Imbalances between energy supply and demand may result in inefficient utilization and network overload.
- Traditional power grids have limited real-time monitoring and control capabilities, reducing their ability to manage dynamic energy demands effectively. [16]

4.3. Energy Optimization:

Energy optimization refers to the process of improving the efficiency of energy production, distribution, and consumption in order to reduce costs, minimize energy losses, and enhance system performance. It plays a crucial role in modern Smart Grid systems, where energy demand is dynamic and requires intelligent management.

The primary objective of energy optimization is to ensure that energy resources are used efficiently while maintaining system stability and reliability. This includes balancing supply and demand, reducing peak loads, and integrating renewable energy sources effectively.

In Smart Grids, energy optimization is achieved through advanced monitoring systems, real-time data analysis, and automated control mechanisms. These systems enable better decision-making, improve energy distribution, and reduce environmental impact by minimizing reliance on non-renewable energy sources. [17]

4.3.1. Energy Optimization Techniques:

Various techniques are applied to optimize energy consumption and improve the efficiency and reliability of power systems, including:

- Demand Response (DR) adjusts consumer energy usage according to supply conditions to reduce peak demand and improve grid stability.
- Load balancing distributes electrical loads efficiently across the network, preventing overload and enhancing system performance.
- Smart scheduling shifts energy consumption to off-peak periods, reducing operational costs and improving energy utilization.

- Real-time monitoring systems such as SCADA and smart sensors continuously collect grid data to support rapid and accurate decision-making.
- Forecasting techniques predict future energy demand using historical and real-time data, enabling better planning and resource management.
- Demand Side Management (DSM) strategies optimize consumer energy consumption through pricing policies and usage control mechanisms.
- The integration of renewable energy sources such as solar and wind power improves sustainability and increases overall energy efficiency. [17]

4.3.2. Role of Artificial Intelligence in Energy Optimization:

Artificial Intelligence enhances energy optimization in Smart Grids by enabling data-driven analysis and smart decision-making, as outlined below:

- Artificial Intelligence (AI) enhances energy optimization in Smart Grids by analyzing large volumes of operational data from sensors, smart meters, and SCADA systems.
- AI techniques support accurate and efficient decision-making for energy management and grid operation.
- Machine Learning models are widely applied for energy demand forecasting to predict future consumption patterns and optimize energy distribution.
- AI-based optimization approaches dynamically adjust system parameters to improve efficiency and reduce energy waste.
- The integration of AI with energy management systems increases automation, adaptability, and sustainability in Smart Grid environments. [18]

5. Conclusion:

This chapter presented the fundamental concepts underlying Smart Grids, energy consumption optimization, and predictive maintenance. It outlined the transition from traditional power systems to intelligent and interconnected Smart Grid infrastructures capable of supporting modern energy requirements.

It also emphasized the importance of optimizing energy consumption to improve system efficiency and reduce operational losses, as well as the role of predictive maintenance in enhancing system reliability through early fault detection and prevention of unexpected failures.

Overall, this chapter highlights the strong interdependence between Smart Grid technologies, energy optimization strategies, and predictive maintenance, which together form the basis of efficient, reliable, and intelligent energy systems.

Chapter 3: AI Techniques Applied to Predictive Maintenance and Energy Consumption Optimization

1. Introduction

The growing complexity of industrial systems and Smart Grid infrastructures has increased the adoption of Artificial Intelligence (AI) for monitoring, prediction, optimization, and decision-making. By analyzing data generated from sources such as SCADA systems, IoT devices, and smart meters, AI techniques can improve operational efficiency, reliability, predictive maintenance, and energy management.

This chapter provides an overview of the main AI methods used in these domains, along with the commonly used datasets and evaluation metrics.

2. Data in Predictive Maintenance and Energy Consumption Optimization

2.1. Benchmark and Open Datasets

Several public datasets are widely used in Predictive Maintenance and energy optimization research for training and evaluating AI models. These datasets generally include vibration, temperature, current, and operational data collected from industrial equipment under normal and faulty conditions. Table 3.1 summarizes the most widely used datasets in predictive maintenance and energy consumption optimization research, adapted from the literature presented in: [19]. [20] , [21]

Dataset	Domain	Signals	Classes / Labels	Key Use Cases
CWRU Bearing Dataset (Case Western Reserve University)	Rotating machinery	Vibration (12 kHz / 48 kHz), 1–3 axes	Normal, Ball, Inner Race, Outer Race faults × 3 severity levels	Fault detection, classification, RUL
PRONOSTIA / FEMTO-ST	Bearing accelerated degradation	Vibration, temperature	Run-to-failure trajectories	RUL estimation, degradation modelling
NASA C-MAPSS	Aircraft turbofan engines	21 sensor channels + 3 operating conditions	RUL in cycles to failure	RUL estimation
IMS (University of Cincinnati)	Bearing	Vibration, 4 bearings simultaneously	Normal → degraded → failure	Degradation monitoring
UCI HVAC Energy Dataset	Building energy	14 variables (temperature, humidity, CO_2 …)	Continuous energy consumption (kWh)	Energy forecasting, optimisation
Pecan Street Dataport	Smart grid / residential	Electricity sub-metering per appliance	Continuous power (W) per circuit	Load disaggregation, DR optimisation
ENTSO-E Transparency Platform	Transmission grid	Generation, load, price, cross-border flows	Continuous at 15-min / hourly resolution	Load forecasting, renewable integration
GEFCom 2014	Energy forecasting competition	Temperature, solar irradiance, load	Hourly probabilistic load & price forecasts	Probabilistic energy forecasting

Table 3. 1. presents some of the most commonly used benchmark datasets in the literature.

2.2. Feature Engineering and Signal Processing

Raw sensor recordings are not directly used by models. Instead, signal preprocessing and feature extraction convert high-dimensional time-series data into compact and informative representations suitable for machine learning.

2.2.1. Time-Domain Features

Time-domain features are computed over fixed-length signal windows and include statistical measures such as:

a) RMS (Root Mean Square):

RMS measures the effective energy or power of the vibration signal. It is one of the most important indicators for detecting machine degradation because vibration energy generally increases when faults develop.

$$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2} \qquad (3.1)$$

A high RMS value often indicates abnormal vibration levels or mechanical wear.

b) Kurtosis:

Kurtosis measures the impulsiveness or peakedness of a signal distribution. It is highly sensitive to sudden shocks and impulsive faults such as bearing defects or cracks.

$$Kurtosis = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{x_i-\mu}{\sigma}\right)^4 \qquad (3.2)$$

Large kurtosis values generally indicate the presence of transient fault impulses.

c) **Skewness:**

Skewness measures the asymmetry of the signal distribution around its mean value.

$$Skewness = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{x_i-\mu}{\sigma}\right)^3 \qquad (3.3)$$

Positive or negative skewness may indicate irregular machine behavior or imbalance conditions. [22]

d) Crest Factor:

The crest factor represents the ratio between the signal peak amplitude and its RMS value.

$$Crest\ Factor = \frac{Peak\ Value}{RMS} \qquad (3.4)$$

It is commonly used as an early fault indicator because localized defects often produce sharp peaks before the RMS increases significantly.

e) Peak-to-Peak Amplitude:

Peak-to-peak amplitude measures the difference between the maximum and minimum signal values.

$$Peak\text{-}to\text{-}Peak = x_{max} - x_{min} \qquad (3.5)$$

This feature reflects the total vibration range and helps identify severe fluctuations or abnormal impacts.

Additional Time-Domain Indicators:

Other commonly used indicators include:

- **Shape Factor**: evaluates waveform shape characteristics.
- **Impulse Factor**: measures impulsive behavior relative to average amplitude.
- **Clearance Factor**: sensitive to bearing and gear faults.

Typically, 10-20 statistical features are extracted from each signal window, creating a compact representation suitable for machine learning models while reducing data dimensionality. [23]

2.2.2. Frequency-Domain Features

Frequency-domain features are extracted using the Fast Fourier Transform (FFT), which converts signals into the frequency spectrum. Faults in rotating machines appear at characteristic frequencies such as:

- **BPFO (Ball Pass Frequency Outer race):** associated with outer race bearing faults.
- **BPFI (Ball Pass Frequency Inner race):** associated with inner race defects.
- **BSF (Ball Spin Frequency):** related to rolling element faults.
- **FTF (Fundamental Train Frequency):** linked to cage defects. [24]

Other Common features include:

a) Spectral Energy

Measures the total energy contained within the frequency spectrum.

$$Spectral\ Energy = \sum |X(f)|^2 \quad (3.6)$$

It indicates the intensity of vibration components at different frequencies.

b) Peak Frequency

Represents the dominant frequency component with the highest amplitude in the spectrum. This feature helps identify rotating speed-related faults and resonance frequencies.

c) Spectral Entropy

Measures the randomness or disorder of the frequency distribution. High entropy values often indicate complex or irregular machine behavior.

d) Harmonic Ratios

Compare amplitudes between harmonic frequency components. These ratios are useful for detecting imbalance, misalignment, and gear-related defects. [25]

Short-Time Fourier Transform (STFT):

For non-stationary signals whose frequency content changes over time, the Short-Time Fourier Transform (STFT) is used to generate time-frequency representations called spectrograms. STFT allows monitoring of how frequency components evolve during machine operation. [26]

2.2.3. Time-Frequency and Wavelet Features

Time-frequency methods, such as Discrete Wavelet Transform (DWT) and Wavelet Packet Decomposition (WPD), analyze signals at multiple resolutions to extract informative features for fault detection. While DWT decomposes signals into approximation and detail components, WPD provides a more detailed spectral representation, improving the detection of complex faults. [27]

2.2.4. Windowing and Overlap

Continuous sensor signals are divided into overlapping windows to create training samples. Appropriate window sizes and overlap ratios improve data representation, while stratified splitting ensures a balanced distribution of fault classes across the datasets. [19]

3. AI Algorithms for Predictive Maintenance and Energy Optimization

Artificial intelligence enables predictive maintenance and energy optimization through fault detection, anomaly identification, and energy forecasting. The following sections present the main AI techniques used in these applications:

3.1. Machine Learning Methods

Many machine learning algorithms have been used, namely:

3.1.1. Support Vector Machines (SVM)

Support Vector Machines (SVMs) are supervised learning models that classify data by finding an optimal decision boundary between classes. They are effective in predictive maintenance when fault data is limited but require careful parameter tuning and can be computationally demanding for large datasets. [28]

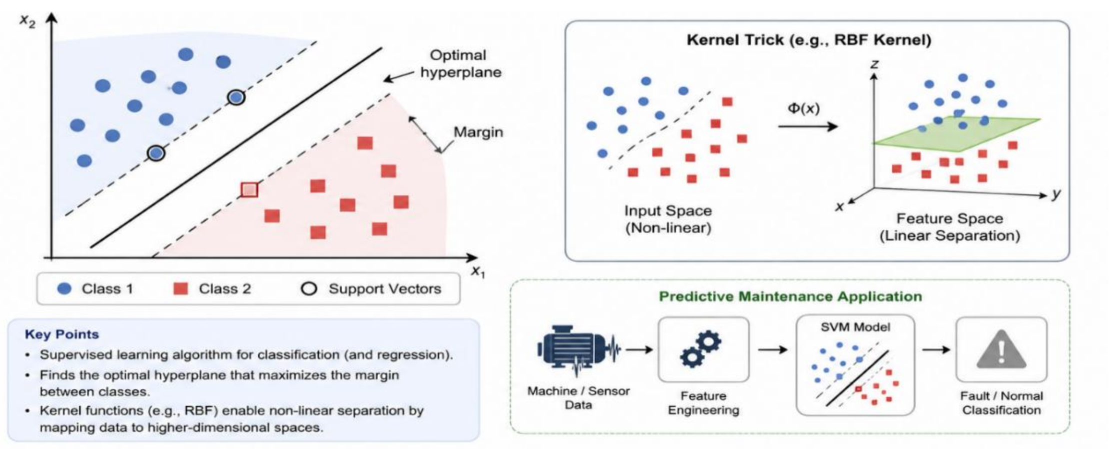
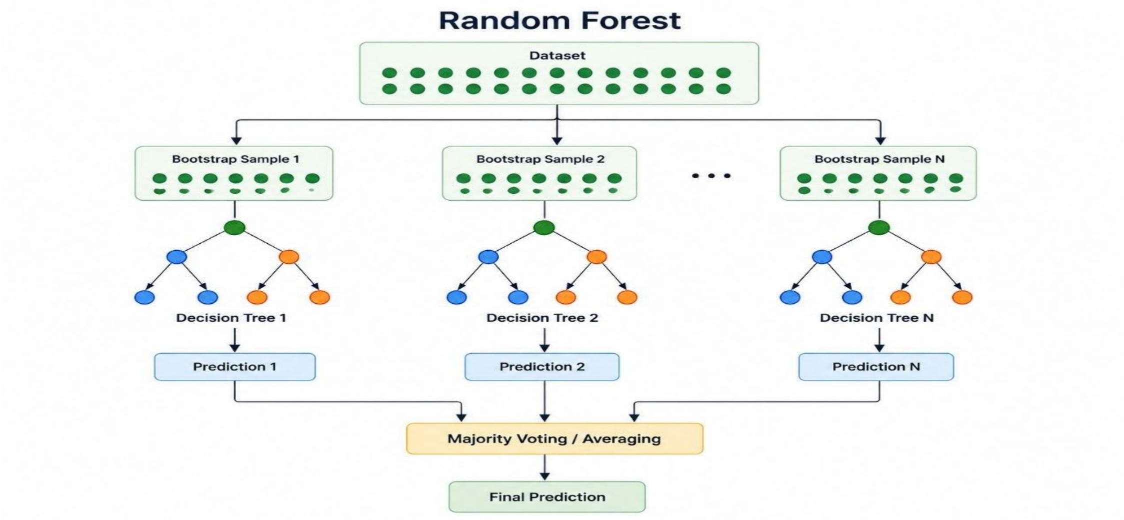


Figure 3.1: Support Vector Machines (SVM)

3.1.2. Random Forests (RF)

Random Forests (RF) are ensemble learning models that combine multiple decision trees to improve prediction accuracy and reduce overfitting. In predictive maintenance, they are widely used for fault detection and feature importance analysis, but their ability to capture temporal dependencies is limited and their performance may be affected by highly

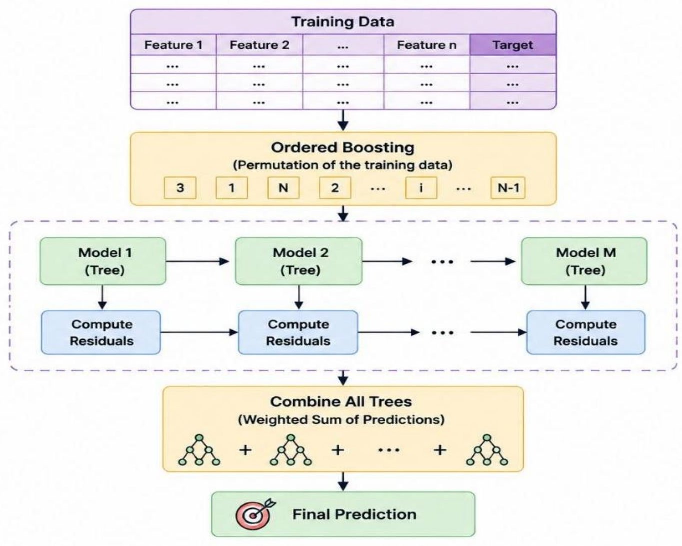


imbalanced data. [29]

Figure 3.2: Random Forests (RF)

3.1.3. Gradient Boosting Machines: XGBoost and CatBoost

Gradient Boosting models, such as XGBoost and CatBoost, build predictive models sequentially to minimize errors and improve accuracy. Due to their robustness and efficiency, they are widely used in predictive maintenance for fault detection and in smart grid applications for energy forecasting, demand prediction, and energy management, often



outperforming traditional machine learning methods on structured data. [30]

Figure 3.3: Gradient Boosting Machines: XGBoost and CatBoost

3.1.4. K-Nearest Neighbours (k-NN)

k-Nearest Neighbours (k-NN) is a simple non-parametric algorithm that classifies samples based on the nearest neighboring instances. In predictive maintenance, it is often used for baseline classification and anomaly detection, although its performance may be limited when handling large datasets due to higher computational and memory requirements. [31]

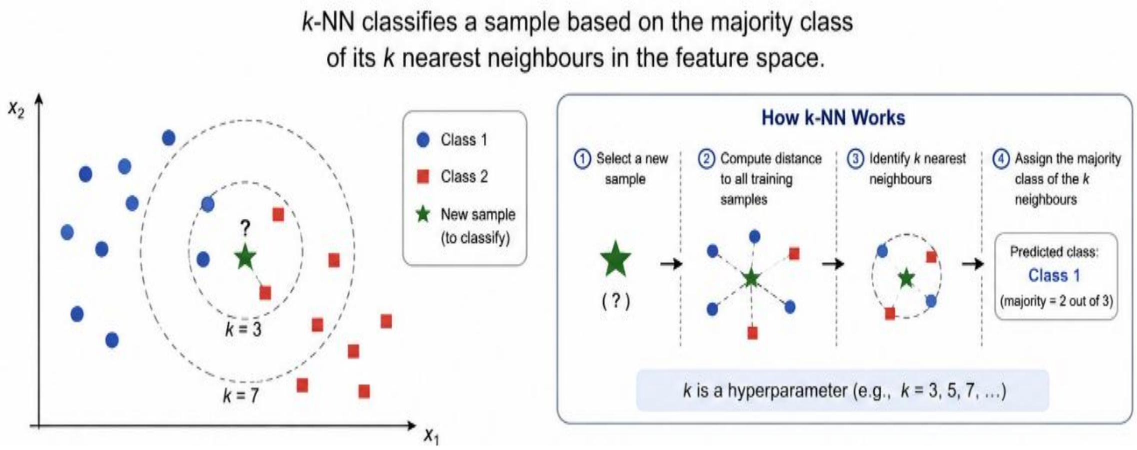


Figure 3.4: K-Nearest Neighbours (k-NN)

3.2. Deep Learning Methods

Deep learning methods have been widely adopted in this field, including:

3.2.1. Convolutional Neural Networks (CNN)

CNNs automatically extract local features from signals and images using convolution filters. In predictive maintenance, they are mainly used for vibration analysis and spectrogram-based fault detection. Their main advantage is end-to-end learning without manual feature engineering, while architectures like ResNet improve training efficiency and performance in fault diagnosis. [32]

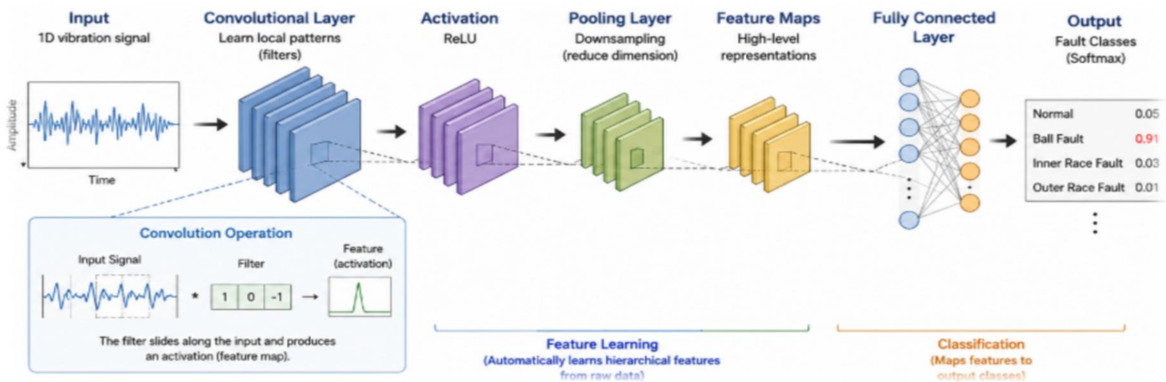
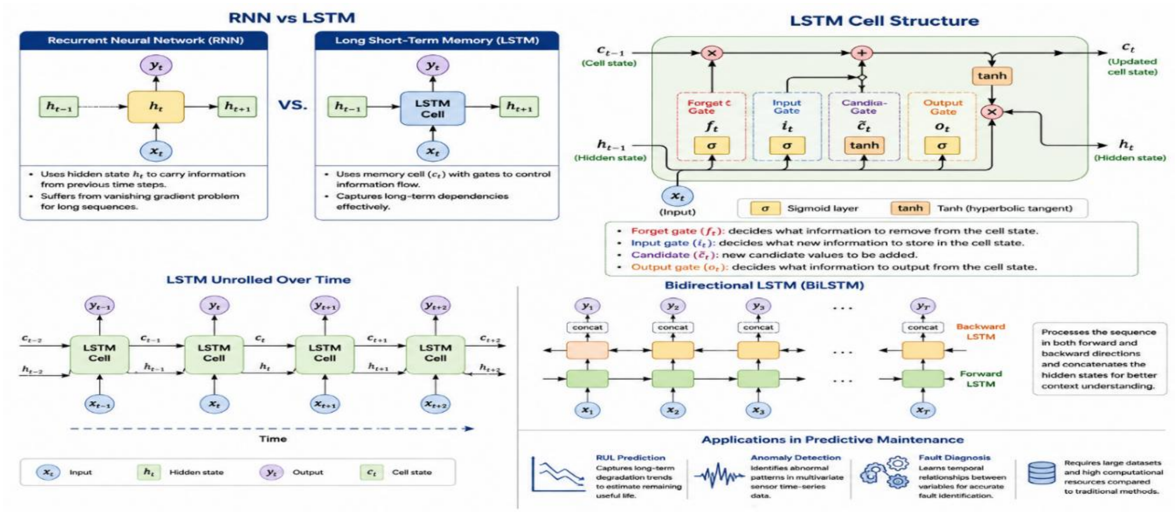


Figure 3.5: Convolutional Neural Networks (CNN)

3.2.2. Recurrent Neural Networks and Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) networks overcome the limitations of traditional RNNs by using memory cells and gates to capture long-term dependencies in sequential data.

They are widely used in predictive maintenance for RUL prediction, anomaly detection, and fault prediction, as well as in energy consumption forecasting based on historical and operational data. Although highly effective, LSTM models generally require large datasets



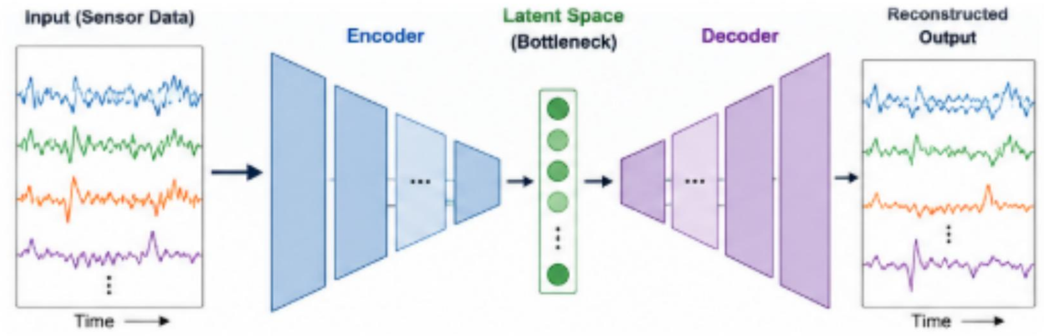
and significant computational resources. [33], [34].

Figure 3.6: Recurrent Neural Networks and Long Short-Term Memory (LSTM)

3.2.3. Autoencoders (AE) and Variational Autoencoders (VAE)

Autoencoders are neural networks that compress input data into a low-dimensional latent representation using an encoder, then reconstruct the original input through a decoder.

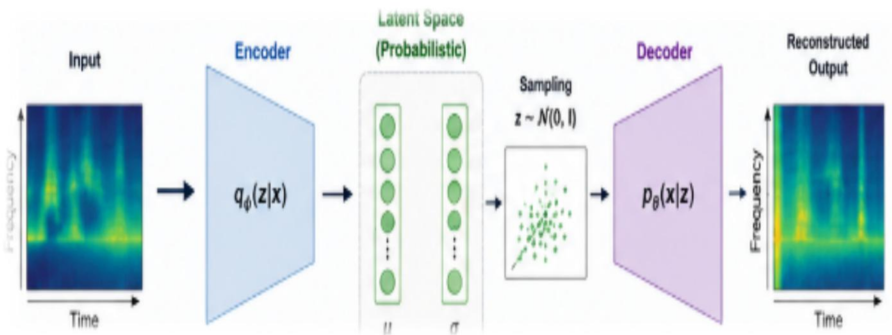
When trained on normal operating data, they learn the patterns of healthy behavior, and reconstruction errors are used to detect anomalies since faulty signals cannot be accurately



rebuilt. [35]

Figure 3.7: Autoencoders (AE)

Variational Autoencoders (VAEs) use a probabilistic latent space to enable both reconstruction and data generation, making them useful for synthetic fault data creation. Variants include CNN-based models for spectrograms and LSTM-based models for sequential data, while semi-supervised approaches combine normal unlabeled data with limited labeled



faults. [36]

Figure 3.8: Variational Autoencoders (VAE)

3.2.4. Generative Adversarial Networks (GAN)

A Generative Adversarial Network (GAN) is composed of a generator that creates synthetic data from noise and a discriminator that distinguishes real from generated samples, trained in a competitive minimax process. In predictive maintenance, GANs help mitigate data imbalance by generating realistic fault data, improving model robustness and classification performance. [37]

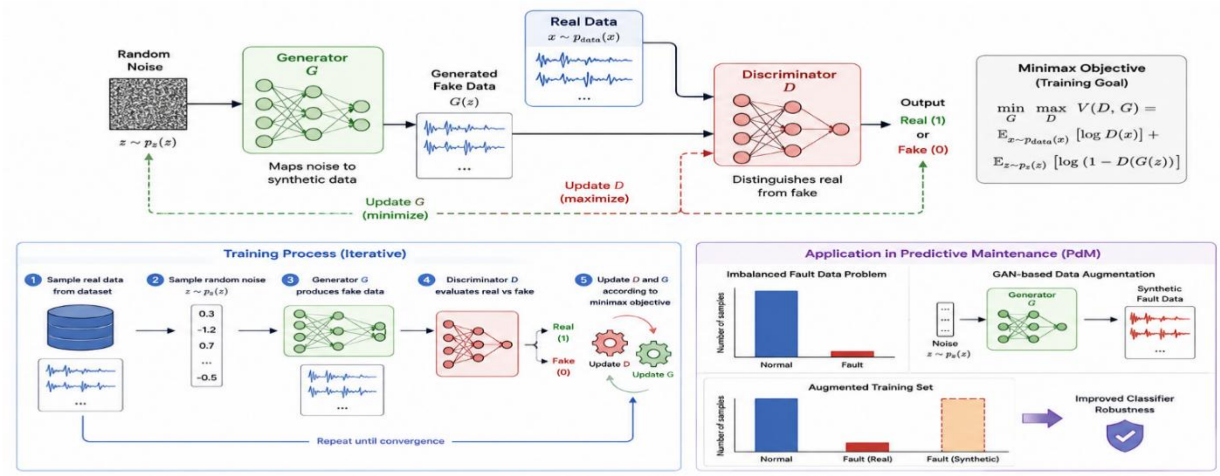


Figure 3.9: Generative Adversarial Networks (GAN)

3.2.5. Graph Neural Networks (GNN)

When sensor data is structured as a graph for example, a power network where nodes are substations and edges are transmission lines GNNs propagate and aggregate feature information along graph edges through successive neighborhood aggregation layers.

In smart grid applications, GNNs model spatial correlations between nodes for fault location in distribution networks and for coordinated energy optimization across multi-bus systems. [38]

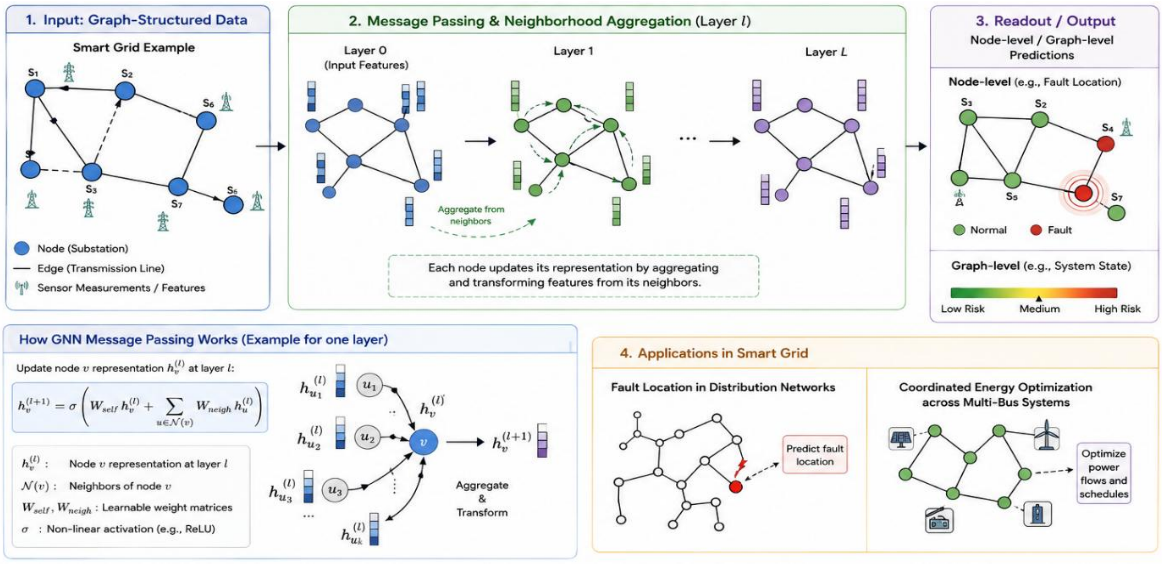


Figure 3.10: Graph Neural Networks (GNN)

3.3. Unsupervised and Semi-Supervised Methods

In this field, methods that do not rely fully on labeled information are also commonly used, including:

3.3.1. Principal Component Analysis (PCA)

Principal Component Analysis (PCA) reduces data dimensionality by projecting it onto directions of maximum variance. In predictive maintenance, it builds a normal behavior baseline, and anomalies are detected using T^2 for variations within the model and Q-statistic for reconstruction errors. Extensions like Kernel PCA and Robust PCA handle non-linear and noisy data. [39]

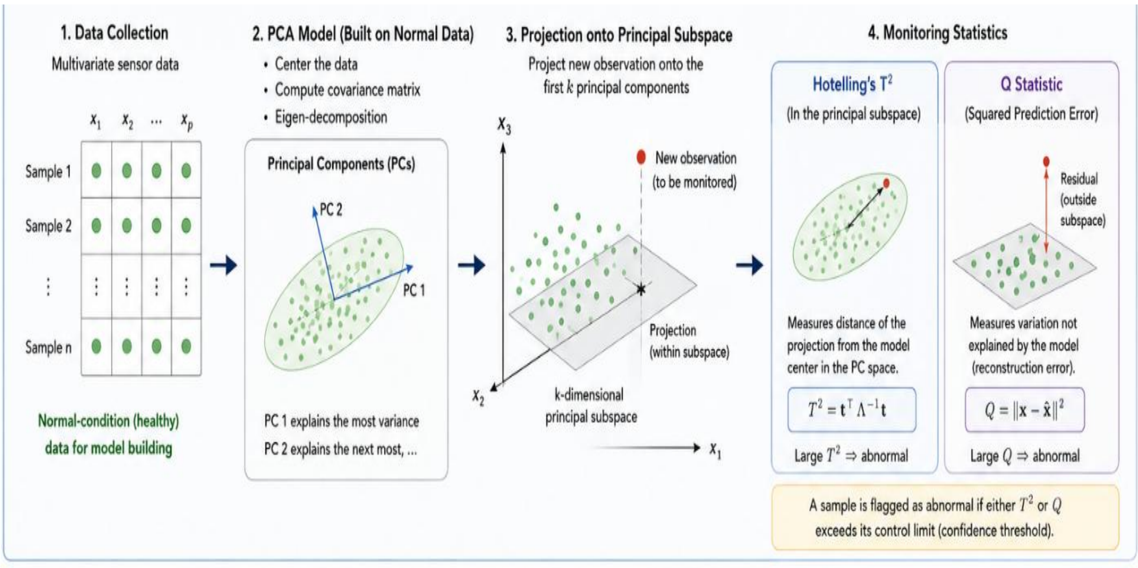


Figure 3.11: Principal Component Analysis (PCA)

3.3.2. Isolation Forest (IF)

Isolation Forest detects anomalies by repeatedly partitioning the feature space using random axis-aligned splits. Data points that lie in sparse or unusual regions are isolated more quickly, resulting in shorter path lengths compared to normal points located within dense clusters. The anomaly score is derived from the inverse of the average path length across a collection of isolation trees. This approach is computationally efficient, with a training complexity of O(n log n), and performs well in high-dimensional spaces without requiring an explicit model of normal behavior. [40]

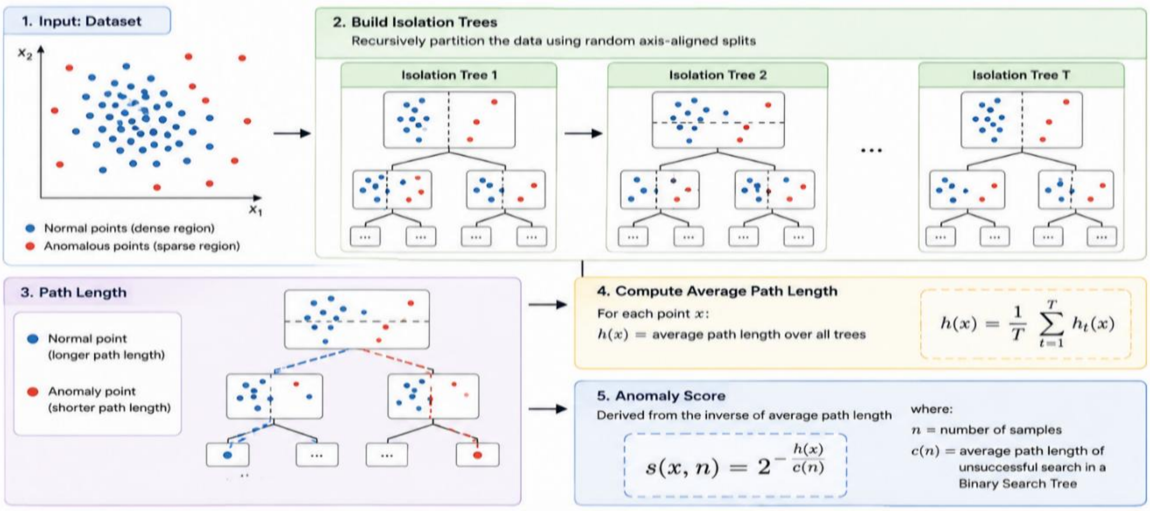
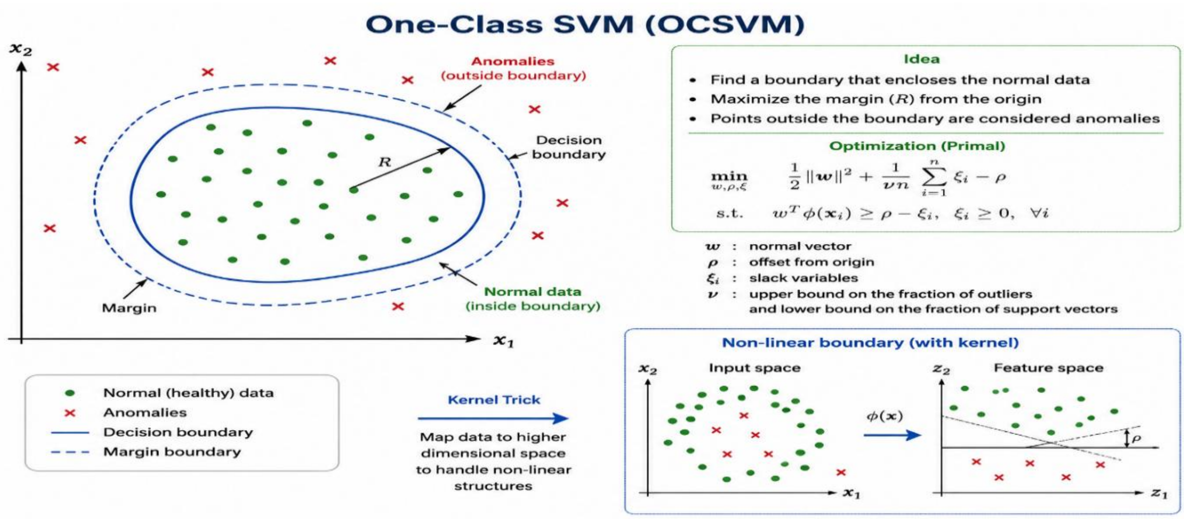


Figure 3.12: Isolation Forest (IF)

3.3.3. One-Class SVM (OCSVM)

OCSVM constructs a decision boundary in the feature space that tightly encloses normal data while maximizing the margin from the origin. Data points falling outside this boundary are considered anomalies. By applying the kernel trick, the method can model non-linear separation boundaries.

OCSVM is particularly useful in predictive maintenance scenarios where only healthy operating data is available for training, which is common in industrial systems due to the



rarity of fault conditions. [41]

Figure 3.13: One-Class SVM (OCSVM)

4. Evaluation Metrics

4.1. Classification Metrics for Fault Detection

Fault detection and identification are classification tasks, and their evaluation must account for class imbalance faults are typically rare relative to normal operation. The corresponding details are shown in Table 3.2. [42]

Metric	Formula	Interpretation
Accuracy	TP + TN / (TP + TN + FP + FN)	Fraction of all correct predictions. Misleading under class imbalance.
Precision	TP / (TP + FP)	Among all predicted faults, how many were true faults. High precision means few false alarms.
Recall (Sensitivity)	TP / (TP + FN)	Among all actual faults, how many were detected. High recall means few missed faults.
F1-Score	2 × Precision × Recall / (Precision + Recall)	Harmonic mean of Precision and Recall. Preferred for imbalanced datasets.
Macro-F1	Mean F1 across all classes	Weights all fault classes equally regardless of their frequency.
Weighted-F1	F1 weighted by class support	Accounts for class frequency; preferred when imbalance is informative.
Matthews Correlation Coefficient (MCC)	(TP×TN − FP×FN) / √((TP+FP)(TP+FN)(TN+FP)(TN+FN))	Balanced measure robust to imbalance; ranges from −1 to +1.
AUC-ROC	Area under Receiver Operating Characteristic curve	Measures discriminability across all classification thresholds.
Cohen's Kappa	(Po − Pe) / (1 − Pe)	Agreement corrected for chance; useful for multi-class scenarios.

Table 3.2. Illustrates the most frequent metrics used to evaluate fault detection classification.

4.2. Anomaly Detection Metrics

Anomaly detection is often evaluated in the absence of complete ground-truth labels. Common metrics include: [43]

Metric	Formula	Description	Main Purpose
Precision@K	Precision@K = |{top-K samples} ∩ {true anomalies}| / K	The fraction of the top-K highest-scoring samples that are true anomalies.	Evaluate ranking quality of anomaly scores at a fixed cutoff K.
Average Precision (AP)	$AP = \Sigma_n (R_n - R_{n-1}) \cdot P_n$ where P_n and R_n are precision and recall at the n-th threshold	The area under the Precision-Recall curve, computed point-by-point as the threshold varies.	Summarize model performance across all classification thresholds.
F1-Score at Optimal Threshold	$F1 = 2 \cdot (Precision \cdot Recall) / (Precision + Recall)$ threshold $t^* = \arg\max F1(t)$ on validation set	The F1 score achieved when the threshold is chosen to maximise F1 on a validation set.	Balance precision and recall at the best operating point.
Reconstruction Error Distribution (AUROC of error)	$error(x) = \lVert x - decoder(encoder(x))\rVert^2$ $AUROC = P(score(x_a) > score(x_n))$ x_a = anomalous sample, x_n = normal sample	For autoencoder-based detectors, the separation between reconstruction error distributions of normal and anomalous samples is visualized via histograms and quantified by the AUROC of the error as a scoring function.	Evaluate the discriminative power of reconstruction error for anomaly detection.

Table 3. 3. summarizes the most used metrics for anomaly detection.

4.3. Energy Forecasting Metrics

Energy consumption and generation forecasting is evaluated with regression metrics adapted to the probabilistic and intermittent nature of energy data: [44]

Metric	Formula	Description	Main Purpose
RMSE (Root Mean Square Error)	$RMSE = \sqrt{(1/n \cdot \Sigma(y_i - \hat{y}_i)^2)}$	Measures the square root of the average squared differences between actual and predicted values.	Evaluate overall prediction accuracy while penalizing large errors.
MAE (Mean Absolute Error)	$MAE = (1/n) \cdot \Sigma\|y_i - \hat{y}_i\|$	Computes the average absolute difference between actual and predicted values.	Measure average prediction error.
Normalised RMSE (nRMSE)	$nRMSE = RMSE / (ymax - ymin)$ or $RMSE / \bar{y}$	RMSE normalized by the range or mean of the target variable.	Compare forecasting performance across datasets with different scales.
Mean Absolute Scaled Error (MASE)	$MASE = MAE / [(1/(n-1)) \cdot \Sigma\|y_i - y_{i-1}\|]$	Compares the model MAE with the MAE of a naïve forecasting baseline.	Determine whether the model outperforms a simple baseline forecast.
Pinball Loss (Quantile Loss)	$Lq(y, \hat{y}) = \max(q(y - \hat{y}), (q-1)(y - \hat{y}))$	Evaluates prediction accuracy across different quantiles.	Assess probabilistic forecasting models.
Energy Score	$ES(F, y) = E\\|X - y\\| - ½E\\|X - X'\\|$	Measures calibration and sharpness in multivariate probabilistic forecasts.	Evaluate the quality of probabilistic energy forecasting.

Table 3.4. summarizes the most used metrics for Energy Forecasting.

5. Methodological Pipeline for AI-Based PdM and OEC Systems

This pipeline defines a standardized workflow for developing robust Predictive Maintenance (PdM) and Optimized Energy Consumption (OEC) systems, ensuring a structured transition from raw industrial data to deployable AI models

5.1. Data Acquisition and Quality Control:

Industrial sensor data is collected with properly configured sampling rates and synchronized across systems. Data quality is ensured by handling issues such as missing values, sensor drift, calibration errors, and noise from electrical interference.

a) **Signal Preprocessing:** Raw signals are cleaned and standardized through normalization, detrending, filtering (e.g., band-pass or anti-aliasing filters), and resampling to ensure consistent temporal resolution across all measurements.

b) **Feature Extraction:** Relevant information is extracted from the processed signals using time-domain statistics, frequency-domain representations (FFT), wavelet transforms, or directly through raw window-based inputs in deep learning models. [45]

c) **Dataset Construction:** The dataset is structured into training, validation, and test sets, ensuring proper representation of different operating conditions and fault types. Data augmentation is often applied to address class imbalance and rare failure events.

d) **Model Selection and Training:** Appropriate AI architectures are selected and optimized through hyperparameter tuning methods such as grid search or Bayesian optimization, followed by cross-validation to ensure generalization performance.

e) **Evaluation and Benchmarking:** Models are evaluated on unseen test data using relevant performance metrics and compared against baseline approaches to validate improvement in prediction or detection capability.

f) **Deployment and Monitoring:** The final model is deployed in edge or cloud environments for real-time inference. Continuous monitoring is implemented to detect data drift and performance degradation, triggering retraining when necessary. [46]

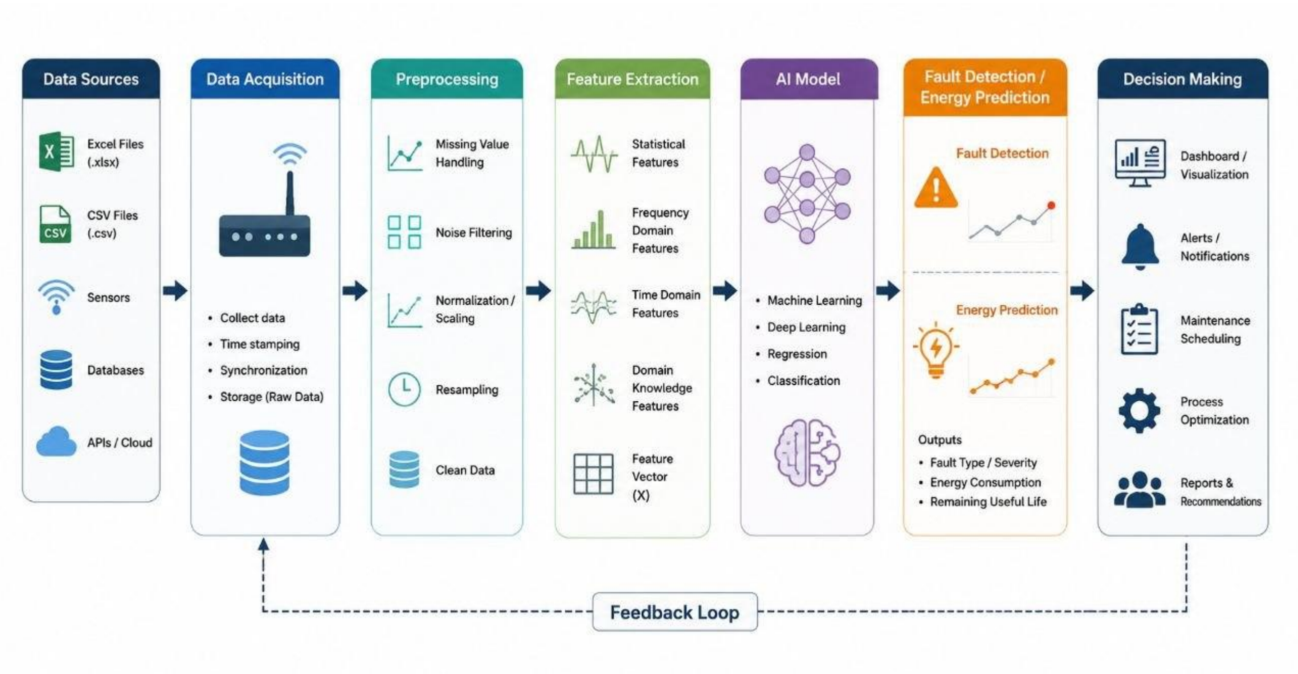


Figure 3.14: Methodological Pipeline for AI-Based PdM and OEC Systems

6. Literature Review and Related Work

This section reviews related work on anomaly detection for predictive maintenance, fault detection and classification, and smart grid energy optimization. It summarizes recent methodologies, datasets, and results to provide the theoretical and practical foundation for the proposed methods in this thesis.

6.1. Anomaly Detection for Predictive Maintenance in Smart Grids

TranAD, proposed by Tuli et al. (2022), is considered one of the most effective Transformer-based models for anomaly detection in multivariate time series. The model is highly relevant to Smart Grid predictive maintenance because electrical systems generate continuous streams of correlated sensor data such as voltage, current, temperature, and power consumption.

The authors designed TranAD using an encoder-decoder Transformer architecture combined with an adversarial training strategy. The model operates in two phases: first, it reconstructs the normal behaviour of the time series, and then it refines anomaly detection by focusing on reconstruction errors. This mechanism allows the system to distinguish abnormal operational patterns from normal fluctuations in grid data.

A major contribution of TranAD is its ability to capture long-range temporal dependencies more effectively than traditional recurrent models such as LSTM. In addition, the model supports online and incremental learning, making it suitable for real-time monitoring environments where Smart Grid data changes continuously.

The study demonstrated that Transformer-based approaches outperform many classical and recurrent deep learning methods in detecting subtle anomalies in multivariate industrial systems. The authors also highlighted that attention mechanisms improve interpretability by identifying which variables contribute most to abnormal behaviour.

For evaluation, TranAD was tested on widely used benchmark datasets including SMAP, MSL, SMD, and SWaT, where it achieved high detection accuracy and low false alarm rates compared to previous anomaly detection models. [47]

6.2. Fault Detection and Classification in Smart Grids Using Machine Learning

Alhanaf et al. (2023) proposed a deep learning framework for fault detection and classification in smart grids.

They simulated an IEEE 6-bus system using MATLAB/Simulink 2022b and generated three-phase voltage and current signals under multiple fault types: SLG, LL, DLG, and 3-phase faults.

Signals were sampled at 2.5 kHz (41 samples per cycle). The dataset also included variations in fault resistance, inception angle, and fault location.

Two models were evaluated: ANN and 1D-CNN. Both were trained on raw signals without feature engineering. Results showed that 1D-CNN achieved near-perfect accuracy:

- 99.99% for line identification
- 99.75% for fault classification
- 98.25% for fault localization

The study concludes that CNNs outperform fully connected networks due to their ability to capture local temporal patterns in electrical waveforms.

This thesis extends this work by integrating classical ML models including CatBoost, Random Forest, SVM, KNN, and Decision Trees for comparative analysis. [48]

6.3. Optimized Energy Consumption in Smart Grids

Huang et al. (2023) proposed an LSTM-based forecasting model combined with an improved Dynamic Programming Algorithm (DPA).

The system includes batteries and supercapacitors, and uses real-world data from the State Grid Fujian (China), including meteorological variables.

The approach follows a forecast-then-optimize strategy:

- LSTM predicts future demand
- DPA optimizes energy scheduling based on predictions

Results show significant energy savings compared to baseline methods, especially under high demand variability. [49]

The reviewed works establish the foundation of this thesis.

This thesis integrates these three domains into a unified smart grid intelligence framework combining predictive maintenance, fault detection, and energy optimization.

7. Conclusion:

This chapter provided an overview of AI techniques used in predictive maintenance and optimized energy consumption in industrial and smart grid systems. It emphasized the role of data processing and feature engineering in preparing sensor data for AI models.

It also reviewed key machine learning, deep learning, and anomaly detection methods, along with datasets and evaluation metrics. Finally, it highlighted how recent research supports the development of intelligent systems that improve reliability, reduce energy use, and enhance decision-making in industrial environments.

Chapter4: Analyse, Design and implementation of the proposed solution

1. Introduction

This chapter outlines the experimental setup of the study, covering three main components: the implementation environment, the datasets used, and the machine learning models applied. It presents the tools and libraries employed in system development, along with the characteristics and preprocessing of the datasets.

It further introduces the different models explored in this work, each associated with its respective dataset and experimental configuration. In addition, the chapter details the design of the platform database, highlighting its overall structure and key components.

2. Tools and Work Environment

This study was conducted in a Python based development environment designed to support both model implementation and system development. The setup includes the following components:

- **Operating System**: Windows 11
- **Programming Language**: Python 3.12.0 (latest major release with performance improvements and new features)
- **Development Stack**: Python libraries and tools for machine learning and web interface development

This environment provided a stable and efficient framework for building and executing the proposed system.

- **JupyterLab:** is an interactive environment used in this project for data exploration and model development within a notebook-based workflow. [50]
- **Google Colab:** was used for training models, benefiting from cloud based execution and GPU support to speed up computations. [51]
- **Visual Studio Code Environment:** was used as the main code editor for developing and organizing the Streamlit application and project files. [52]
- **Reason for Selection:** VS Code was selected for its simplicity, flexibility, and strong support for Python and Streamlit development.

2.1. Python Tools, Libraries and Frameworks

The proposed system was developed using a collection of Python libraries and frameworks that support data processing, machine learning, visualization, optimization, and web application development. These tools were selected to facilitate the implementation of predictive models as well as the deployment of the platform interface.

2.2. Web Framework

-Streamlit: was used to develop the web based interface of the platform, providing an interactive environment for data input, prediction, and result visualization.

2.3. Machine Learning and Artificial Intelligence Libraries

- **Scikit-learn:** Used for data preprocessing, model development, and performance analysis.
- **TensorFlow/Keras:** Used for implementing and training deep learning models, particularly the Autoencoder architecture.
- **CatBoost, XGBoost, and LightGBM:** Gradient boosting frameworks employed for prediction and forecasting tasks.

3. Data Processing Libraries

- **SciPy**: Provided additional scientific computing and statistical analysis functionalities.

3.1. Data Visualization Libraries

-Matplotlib, Seaborn, and Plotly: Used to visualize data distributions, model performance, and analytical results through static and interactive charts.

3.2. Utility Libraries

- **Joblib:** Used for saving and loading trained models efficiently.
- **OS and Datetime:** Used for file management and handling date and time operations within the system.
- **Database Module (db_queries):** A custom module developed to manage database interactions and support data retrieval and storage operations within the platform.

4. Implementation of the Anomaly Detection Framework

4.1. Dataset Description:

The dataset was collected from a real gas turbine facility operated by Sonelgaz in M'sila, Algeria, using a Testo 350 analyzer between 2022 and 2023, resulting in about 3,200 sensor readings. All measurements were recorded while the turbine was in idle or shutdown conditions, with oxygen levels close to 21%, indicating no active combustion. Due to the absence of labeled fault data, supervised classification could not be applied, leading to the use of a one-class learning approach combined with physics-informed synthetic data generation.

4.2. Measurement Equipment

Gas measurements were collected using the Testo 350 flue gas analyzer, a portable industrial combustion analysis instrument capable of simultaneously measuring six exhaust gas parameters from the gas turbine. Table 4.1 presents the measured variables and their corresponding units.

Parameter	Symbol	Unit	Physical Meaning
Flue Gas Temperature	°C TF	°C	Temperature of exhaust gases at measurement point
Sulfur Dioxide	ppm SO_2	ppm	Combustion by-product; indicator of fuel quality
Oxygen Content	% O_2	%	Residual oxygen; reflects combustion air excess
Carbon Monoxide	ppm CO	ppm	Incomplete combustion indicator; early fault signal
Draft Pressure	mbar Tir.	mbar	Chimney suction; always negative in normal operation
Ambient Temperature	°C TA	°C	Reference temperature for normalization

Table 4.1: Measured parameters from the testo 350 flue gas analyzer.

4.3. Physics-Informed Synthetic Data Generation

Due to the lack of labeled fault data and the absence of active combustion measurements, a physics-informed synthetic data generation approach was used. A dataset of 50,000 synthetic samples was created to represent four normal operating scenarios, with all samples calibrated according to the statistical characteristics of the real Sonelgaz measurements.

➢ **Scenario 1: Stable Combustion (60%, 30,000 samples)**

This scenario represents the most common and critical operating state: the turbine running at full load with stable, complete combustion. able 4.2 presents the parameter distributions, which were set as follows:

Parameter	**Mean**	**Std. Dev.**	**Physical Justification**
°C TF	540°C	20°C	Efficient energy extraction; residual exhaust heat after turbine work
% O_2	20.97%	0.05%	Lean mixture with large air excess characteristic of low-NOx burners
ppm CO	~1 ppm	0.3 ppm	Near-zero CO confirms complete combustion; slight measurement noise
mbar Tir.	-0.18 mbar	0.02	Moderate negative draft, confirms proper chimney suction

Table 4.2. Parameter distributions for Scenario 1 (Stable Combustion).

- **Scenario 2: Ideal Combustion (20%, 10,000 samples)**

Peak efficiency operation where the turbine reaches its optimal performance point maximum energy extraction per unit of fuel. This state is less frequent but represents the best-case operational benchmark.

- **Scenario 3: Ambient Air Measurement (10%, 5,000 samples)**

The sensor measures ambient air during pre-ignition functional checks, post shutdown cooling periods, or maintenance intervals. O_2 approaches 21% and TF approaches ambient temperature. Including this scenario prevents false alarms during startup and shutdown transitions.

- **Scenario 4: Transitional State (10%, 5,000 samples)**

Cold start or ramp-down periods during which all parameters change continuously. This state typically lasts 10-30 minutes for a large industrial gas turbine. Parameters evolve between ambient and full-load values according to documented turbine dynamics.

4.4. Data Preprocessing and Feature Engineering

4.4.1. File Reading and Data Cleaning

The raw data were imported from Excel (.xlsx) files using Pandas and underwent a preprocessing stage to clean and standardize the measurements. This process included removing unit suffixes, converting values to numerical format, correcting invalid or physically impossible values, handling missing data through median imputation, and converting timestamps into datetime format for further temporal analysis.

4.4.2. Feature Engineering

The raw sensor readings are not fed directly into the Autoencoder. Two additional engineered features are derived from the timestamp to capture daily operational patterns. The final input vector has dimension 8.

➢ **Features 1-6: Raw Sensor Readings**

The model relies on six key cleaned sensor measurements from the Testo 350 analyzer to describe the turbine's operating condition:

- **Feature 1: Flue Gas Temperature (°C TF):** Primary indicator of combustion health and thermal performance.
- **Feature 2: Sulfur Dioxide (ppm SO_2)**: Reflects fuel quality and combustion completeness.
- **Feature 3: Oxygen Level (% O_2)**: Indicates air–fuel ratio and combustion stoichiometry.
- **Feature 4: Carbon Monoxide (ppm CO)**: Early indicator of incomplete combustion and potential faults.
- **Feature 5: Draft Pressure (mbar Tir.)**: Represents exhaust flow and system ventilation efficiency.
- **Feature 6: Ambient Temperature (°C TA)** : Provides environmental reference for normalization.

➢ **Features 7-8: Cyclic Time Encoding**

Time information is encoded using sine and cosine transformations of the hour of day to preserve its cyclic nature. This representation maps time onto a unit circle, ensuring that adjacent hours (e.g., 23 and 0) remain close in the feature space and improving the model's ability to capture time-dependent behavior.

➢ **Complete Feature Vector**

After raw sensor extraction and feature engineering, every sample is represented as an 8-dimensional vector:

x = [Feature 1: °C TF**, Feature 2:** ppm SO_2, **Feature 3**: % O_2, **Feature 4:** ppm CO**, Feature 5:** mbar Tir**, Feature 6:** °C TA**, Feature 7:** hour_sin**, Feature 8:** hour_cos**]**

This vector is simultaneously the input to and the reconstruction target of the Autoencoder.

4.4.3. Normalization-MinMax Scaling

All eight features are scaled to the range [0, 1] using MinMax Scaling. This choice ensures compatibility with the Autoencoder's output layer and better reflects the physical bounds of gas sensor measurements.

To avoid data leakage, MinMax Scaler is fitted only on the training set and then applied to the test set without refitting, preventing any influence of test data statistics on model training and ensuring unbiased performance evaluation.

4.5. Autoencoder Architecture and Training

4.5.1. Model Architecture

The proposed anomaly detection model is a symmetric deep Autoencoder implemented using the TensorFlow/Keras framework. The architecture consists of an encoder path that progressively compresses the 8-dimensional input into a 2-dimensional latent representation, followed by a symmetric decoder path that reconstructs the original input from this bottleneck representation. he architecture of the proposed Autoencoder is summarized in Table 4.4:

Layer	Type	Units	Activation	Role
Input	Input Layer	8	--	8-dimensional feature vector
Encoder 1	Dense	6	ReLU	First compression stage
Encoder 2	Dense	4	ReLU	Second compression stage
Bottleneck	Dense	2	ReLU	Latent representation (most compressed)
Decoder 1	Dense	4	ReLU	First reconstruction stage
Decoder 2	Dense	6	ReLU	Second reconstruction stage
Output	Dense	8	Sigmoid	Reconstructed feature vector ∈ [0, 1]

Table 4.4. Autoencoder layer-by-layer architecture summary.

The Autoencoder uses ReLU in hidden layers for stable non-linear learning and a Sigmoid output layer to ensure outputs stay within the [0, 1] range. A 2-dimensional bottleneck is used to strongly compress the input, forcing the model to learn only normal operating patterns and resulting in high reconstruction errors for anomalous data.

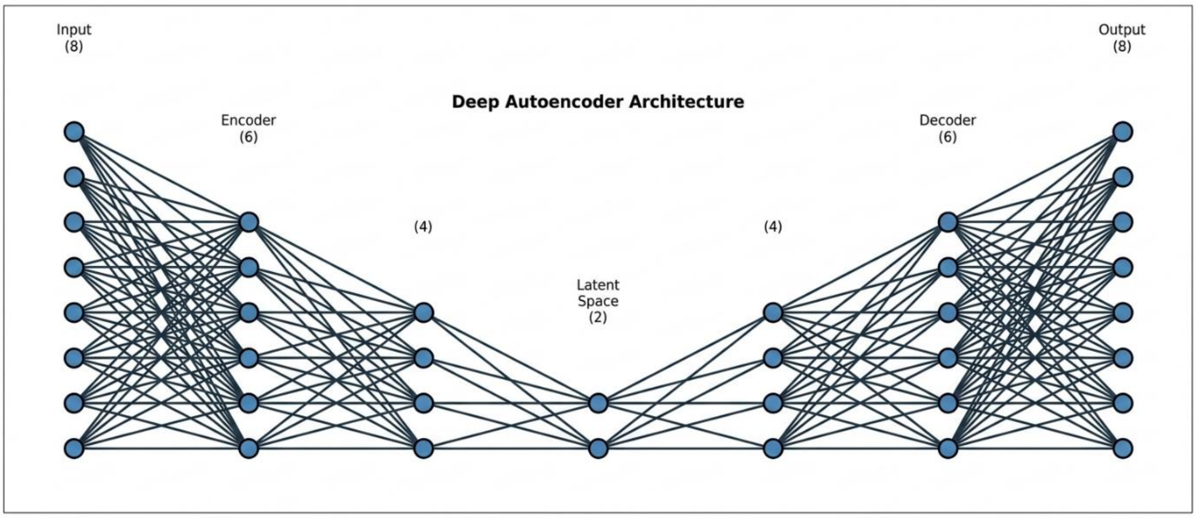


Figure 4.1: Deep Autoencoder architecture

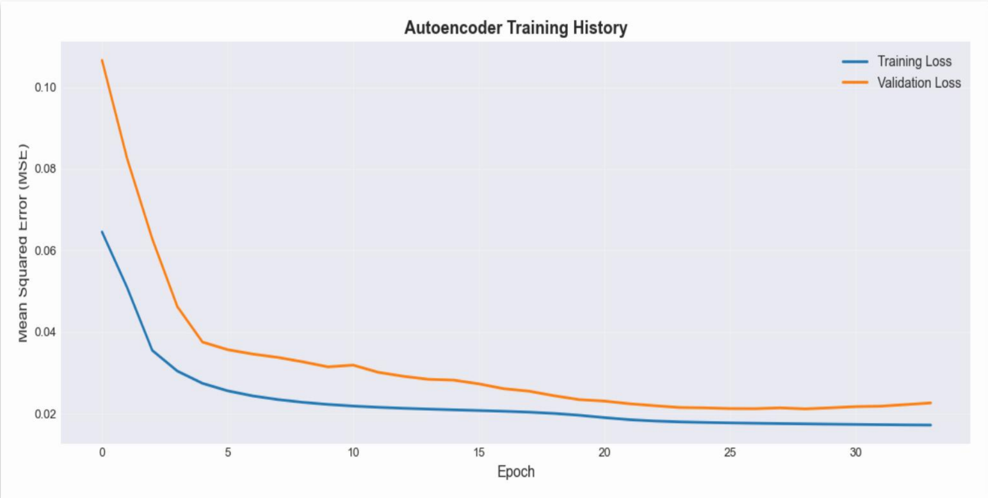
4.5.2. Training Procedure

The Autoencoder is trained using the Adam optimizer with Mean Squared Error (MSE) as the loss function. The training configuration is summarized in Table 4.5 :

Hyperparameter	**Value**	**Justification**
Optimizer	Adam	Adaptive learning rate; robust to noisy gradients
Loss Function	MSE (Mean Squared Error)	Penalizes large reconstruction deviations
Epochs	50 (with early stopping)	Prevents overfitting; training halts at optimal point
Batch Size	128	Balance between gradient accuracy and training speed
Validation Split	10%	Monitor generalization without data leakage
Early Stopping Patience	5 epochs	Restores best weights when val_loss stagnates
Training Samples	50,000	Physics-informed synthetic normal data

Table 4.5: Autoencoder training hyperparameters.

Early stopping monitors the validation loss at the end of each epoch and halts training when no improvement is observed for 5 consecutive epochs. The **restore_best_weights=True** flag ensures that the final model retains the weights from the



epoch with the lowest validation loss, not the weights from the last epoch.

Figure 4.2: Training history plot: training loss and validation loss vs. epoch

4.5.3. Anomaly Threshold and Anomaly Type Classification

1) Anomaly Threshold Determination

After training, reconstruction errors are computed on the normal dataset to establish a baseline distribution. An anomaly threshold is then defined using the 95th percentile of these errors, allowing only a small fraction of normal samples to exceed it and be considered anomalous.

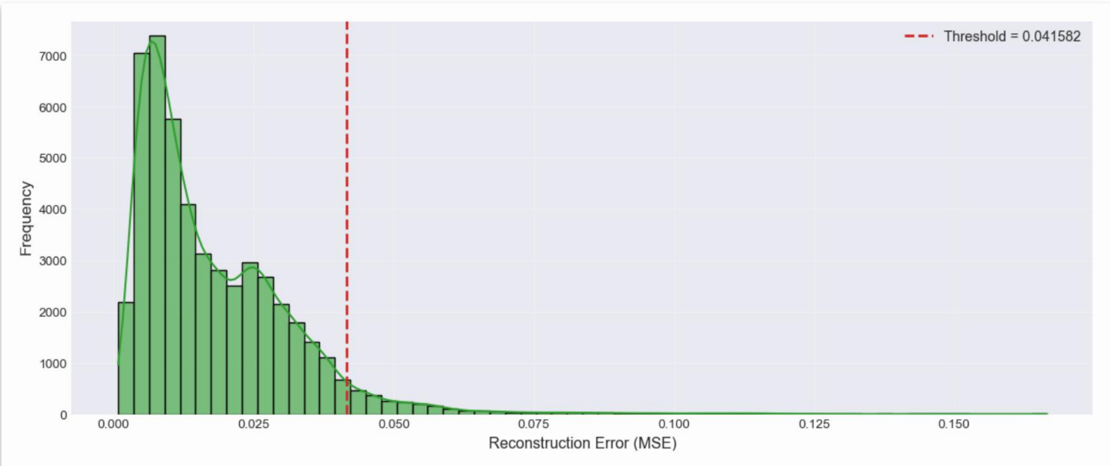


Figure 4.3: Histogram of training reconstruction errors with threshold line

2) Anomaly Type Classification

Before building the test dataset, anomaly types were defined based on combustion physics and statistical patterns observed in the Sonelgaz data. Three anomaly categories were identified:

- **Anomaly Type 1 :CO_eleve (Carbon Monoxide Spike)**

Carbon monoxide is a key indicator of incomplete combustion. Under normal conditions, methane combusts fully to produce CO_2 and H_2O, but when oxygen is insufficient or flame stability is disturbed, partial combustion occurs and CO is generated. This makes CO a leading early-warning signal, often increasing before changes in temperature or oxygen are observed.

Typical real-world causes include:

- Partial blockage of the fuel injection nozzle
- Reduced fuel pressure or flow rate
- Lower combustion zone temperature below ignition threshold
- Degradation of flame stabilizer performance

- **Anomaly Type 2 :TF_bas (Temperature Drop)**

Flue gas temperature is typically between 400°C and 555°C under normal full-load operation. A sharp drop to 50-200°C while the turbine is still running indicates a major disturbance such as air leakage into the exhaust system, sensor malfunction, or partial fuel supply interruption due to control valve failure. This anomaly is highly detectable due to its strong deviation from normal operating ranges.

- **Anomaly Type 3: tirage_anormal (Draft Pressure Anomaly)**

Draft pressure reflects the suction force generated in the exhaust chimney and is an important indicator of proper exhaust gas flow.

- Draft pressure is normally negative during standard turbine operation.
- Positive or near-zero values indicate abnormal exhaust conditions.

Possible causes include:

- Chimney blockage
- Damaged exhaust duct seals
- Exhaust fan malfunction

These faults disrupt the natural draft flow and can affect overall turbine performance.

4.6. Experimental Results and Model Evaluation

4.6.1. Evaluation Metrics

The model is evaluated using a mixed test dataset containing both normal samples and injected anomaly samples with known ground-truth labels. The following metrics are computed:

Metric	**Value**
Precision	0.91
Recall	0.88
F1-Score	0.895
ROC-AUC	0.972
Accuracy	94.8%

Table 4.6: Model Performance Metrics

The obtained results demonstrate that the proposed Autoencoder-based anomaly detection system achieves a high level of performance in distinguishing normal and abnormal operating conditions of the gas turbine.

- **Accuracy (94.8%):** Indicates that the model correctly classifies most samples, demonstrating strong overall performance.
- **Precision (0.91):** Shows a low false-positive rate, meaning most detected anomalies are genuine faults.
- **Recall (0.88):** Confirms the model's ability to identify the majority of anomalous conditions.
- **F1-Score (0.895):** Reflects a good balance between anomaly detection capability and prediction reliability.
- **ROC-AUC (0.972):** Demonstrates excellent discrimination between normal and anomalous operating states.

4.6.2. Autoencoder Detection Results

The trained Autoencoder with the 95th percentile threshold achieved the following performance on the mixed test dataset.

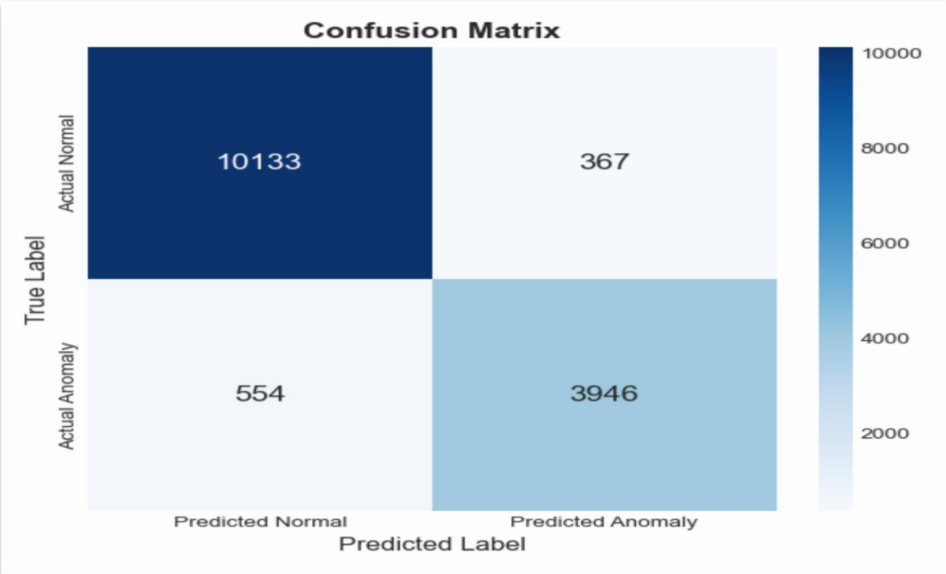


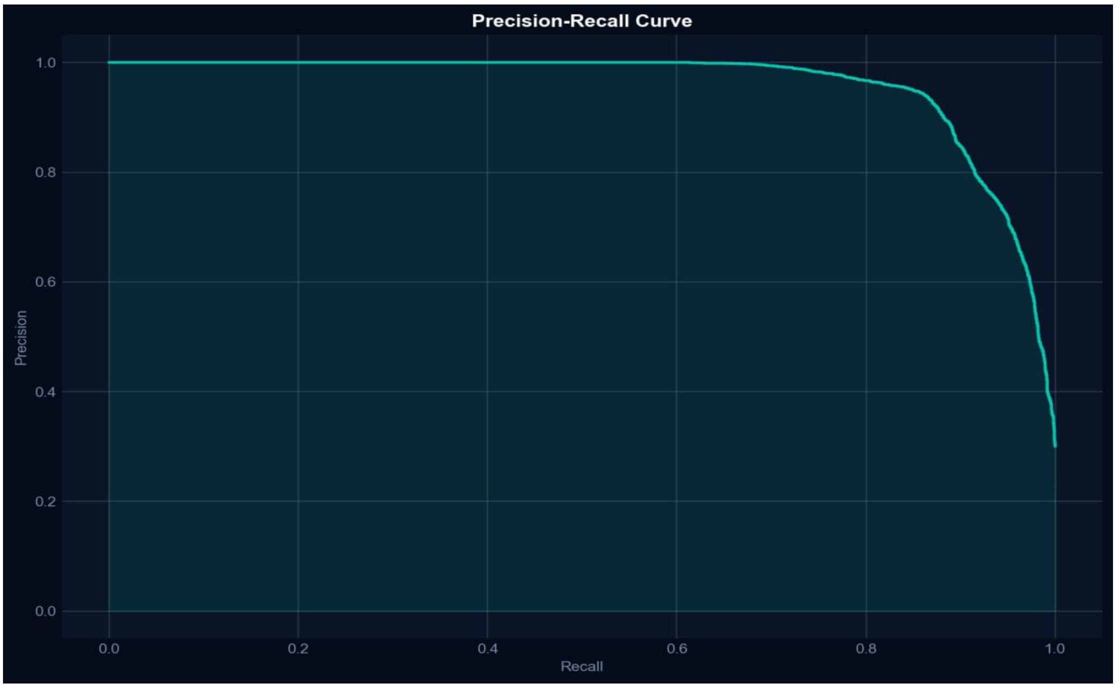
Figure 4.4: Confusion matrix for Autoencoder anomaly detection

The confusion matrix was used to assess the model's classification performance by comparing correct and incorrect predictions for normal and anomalous samples. And the result was as follows :

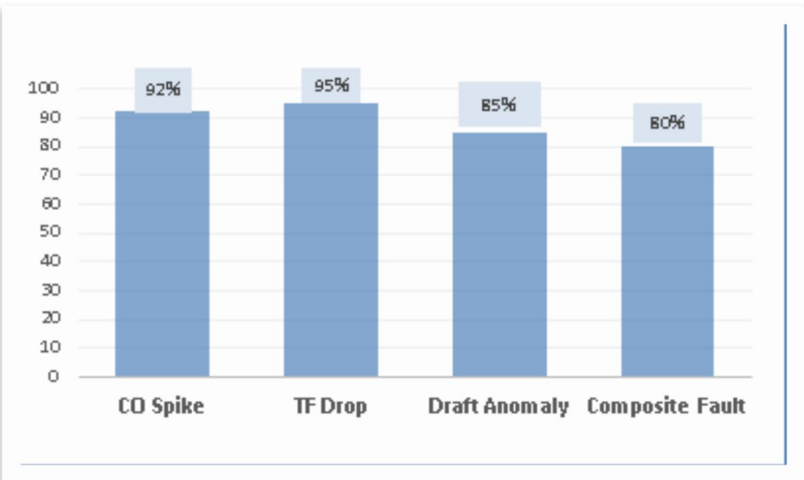
- **True Negatives (TN = 10,133):** The model correctly identified 10,133 normal samples, demonstrating a strong ability to learn and recognize normal turbine behavior.
- **True Positives (TP = 3,946):** The model successfully detected 3,946 anomalous samples, confirming its effectiveness in identifying abnormal operating conditions.
- **False Positives (FP = 367):** Only 367 normal samples were incorrectly classified as anomalies, indicating a relatively low false-alarm rate and reducing unnecessary maintenance interventions.
- **False Negatives (FN = 554):** A total of 554 anomalies were classified as normal. Although this represents a small proportion of the dataset, further optimization could improve the detection of subtle fault patterns.

The confusion matrix shows a clear dominance of correct classifications (TN and TP) over misclassifications (FP and FN). This confirms that the proposed Autoencoder model provides reliable anomaly detection with high accuracy, strong fault recognition capability, and a low rate of false alarms, making it suitable for predictive maintenance applications in gas turbine systems.

Figure 4.5: Precision-Recall Curve Analysis of Deep Autoencoder for Anomaly Detection



The figure presents the Precision–Recall curve of a Deep Autoencoder used for anomaly detection, showing strong performance with near-perfect precision up to 0.85 recall, followed by a gradual decline as recall increases. At the optimal threshold, the model achieves 0.91 precision and 0.88 recall, indicating a good balance between detecting anomalies and minimizing false alarms. Overall, the curve confirms high discriminative ability, with reduced



performance only when attempting to capture highly subtle and complex fault cases.

Figure 4.6: Model Performance Across Different Anomaly and Fault Categories

The figure presents the classification performance of the proposed model in terms of detection accuracy across four distinct fault categories:

- **CO Spike (92%):** The model achieves a high accuracy of 92% for CO Spike events, indicating strong sensitivity to abrupt variations in CO-related signals. This suggests that the learned features effectively capture sudden emission-related disturbances, which are typically well-defined and easier to discriminate.
- **TF Drop (95%):** The highest performance is observed for TF Drop, with an accuracy of 95%. This indicates that the model is particularly effective at identifying temperature or flow-related drops, likely due to their consistent temporal patterns and clear deviation from normal operating behavior.
- **Draft Anomaly (85%):** For Draft Anomaly, the accuracy decreases to 85%, reflecting moderate difficulty in classification. This reduction suggests that draft-related anomalies may exhibit less distinct patterns or greater overlap with normal operational variability, making them harder for the model to separate.
- **Composite Fault (80%):** The lowest performance is recorded for Composite Faults at 80%. This is expected, as composite faults typically involve multiple simultaneous abnormal conditions, increasing the complexity of the feature space and making accurate classification more challenging.

4.6.3. Comparison with Baseline Anomaly Detection Methods

To contextualize the Autoencoder's performance, three baseline one-class and anomaly detection algorithms were evaluated under identical conditions on the same dataset.

Model	Precision	Recall	F1-Score	ROC-AUC
Autoencoder	0.91	0.88	0.895	0.972
Isolation Forest	0.83	0.49	0.62	0.894
One-Class SVM	0.93	0.69	0.79	0.87

Table 4.7: Comparative performance of anomaly detection methods

The results show clear performance differences between the three models:

Autoencoder achieves the best overall performance with high Precision (0.91), Recall (0.88), F1-Score (0.895), and the highest ROC-AUC (0.972), indicating strong and balanced detection capability.

Isolation Forest performs the weakest, especially in Recall (0.49), meaning it misses many anomalies despite acceptable Precision (0.83). Its overall F1-Score is low (0.62).

One-Class SVM shows high Precision (0.93) but moderate Recall (0.69), resulting in a medium F1-Score (0.79) and lower ROC-AUC (0.87), indicating a conservative detection behavior.

The Autoencoder is the most effective model, providing the best balance between detecting anomalies and minimizing errors.

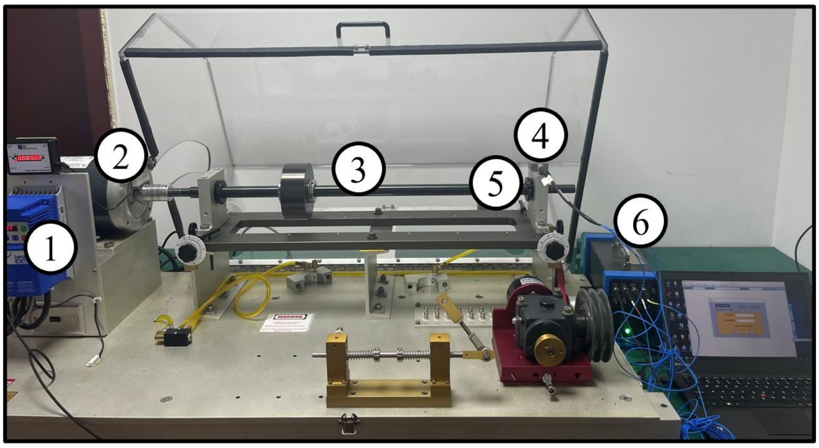
5. Implementation of the Intelligent Fault Detection Framework

5.1. Dataset Description: HUSTbearing

The HUSTbearing dataset is a public bearing vibration dataset from Huazhong University of Science and Technology (HUST), containing 99 files covering 9 health states under 11 operating conditions, and is widely used for fault diagnosis. ‘Domain Generalization for Cross Domain Fault Diagnosis: an application-oriented Perspective and a Benchmark Study,” Reliability Eng.Syst.Saf.,2024, [https://doi.org/10.1016/ j.ress.2024.109964](https://doi.org/10.1016/j.ress.2024.109964)

5.1.1. Experimental Test Rig

All bearing fault experiments were conducted using a Spectra Quest Mechanical Fault



1: Speed control, 2:Motor, 3: Shaft, 4: Acceleration sensor, 5: Bearing, 6: Data acquisition board

Simulator. The test rig consists, from left to right:

Figure 4.7: Test rig of HUSTbearing dataset.

5.1.2. Bearing Specifications

The tested bearing type is the ER-16K. The principal parameters of the bearing, along with key acquisition settings.

5.1.3. Fault Types and Health States

The dataset contains 9 bearing health states, including inner race, outer race, ball, and combined faults with medium or severe damage levels. Files with the prefix “0.5X” indicate medium faults, while the others represent severe faults.

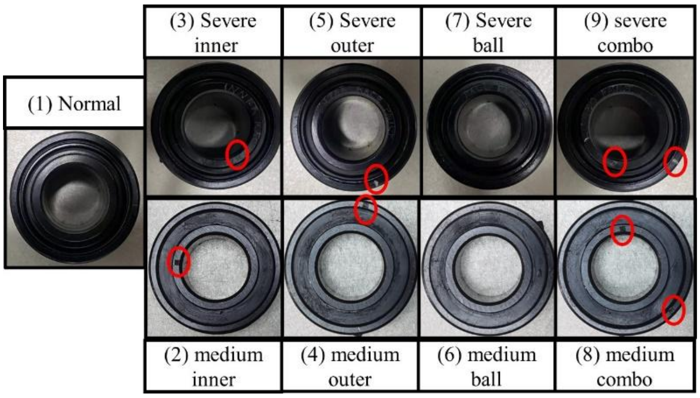


Figure 4.8: Photographs of the failure bearings.

Code	Fault Description	Severity
H	Healthy (Normal)	--
I (medium)	Inner Race Fault	Medium (0.5×)
I (severe)	Inner Race Fault	Severe
O (medium)	Outer Race Fault	Medium (0.5×)
O (severe)	Outer Race Fault	Severe
B (medium)	Ball Fault	Medium (0.5×)
B (severe)	Ball Fault	Severe
C (medium)	Combination Fault (inner + outer)	Medium (0.5×)
C (severe)	Combination Fault (inner + outer)	Severe

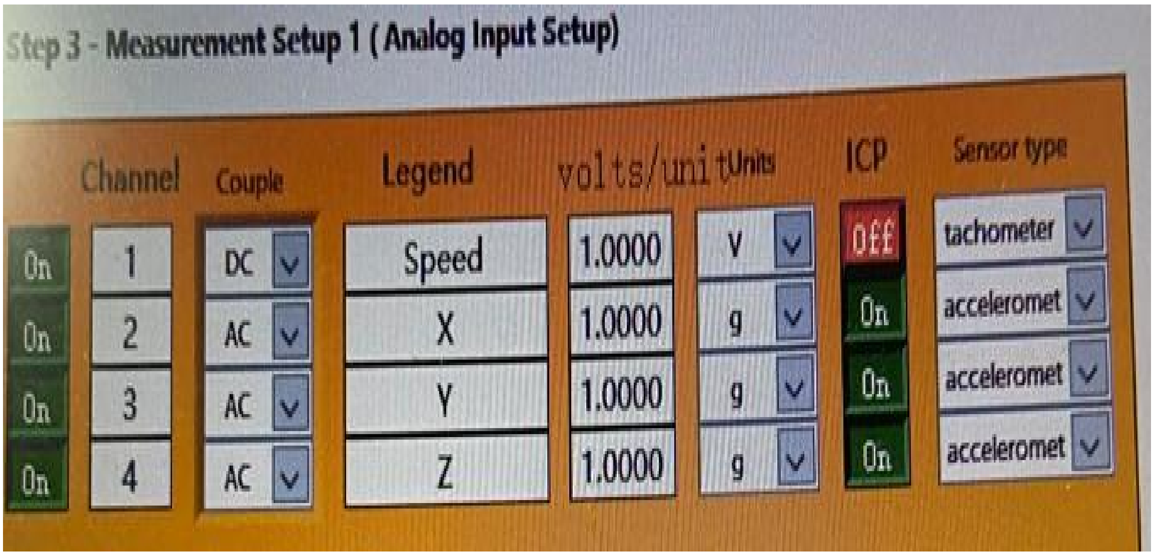
Table 4.8: Bearing Health States in the HUSTbearing Dataset

5.1.4. Filename convention:

0.5X_B_65Hz indicates a medium ball fault at 65 Hz, while O_80Hz indicates a severe outer race fault at 80 Hz. Healthy files use the prefix H_. This figure shows where each fault occurs in the bearing and its severity level.

5.1.5. Operating Conditions and Signal Acquisition

The dataset includes 11 operating conditions: 10 constant speeds from 20 Hz to 80 Hz, and one varying speed profile (0 to 40 to 0 Hz) simulating transient behavior. Vibration signals were captured along the X, Y, and Z axes using a three-axis



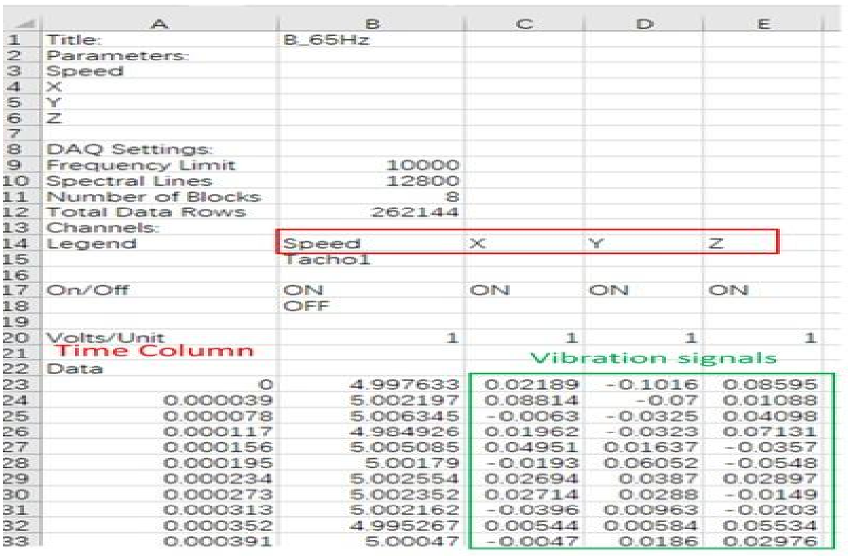
accelerometer at 25.6 kHz, with each file containing 262,144 samples (≈ 10.2 s)

Figure 4.9: Signal acquisition setting.

5.2. Data Preprocessing

5.2.1. File Reading

CSV and Excel files are supported. The first 22 metadata rows are skipped, and



	A	B	C	D	E
1	Title:	B_65Hz			
2	Parameters:				
3	Speed				
4	X				
5	Y				
6	Z				
7					
8	DAQ Settings:				
9	Frequency Limit	10000			
10	Spectral Lines	12800			
11	Number of Blocks	8			
12	Total Data Rows	262144			
13	Channels:				
14	Legend	Speed	X	Y	Z
15		Tacho1			
16					
17	On/Off	ON	ON	ON	ON
18		OFF			
19					
20	Volts/Unit	1	1	1	1
21	Time Column			Vibration signals	
22	Data				
23	0	4.997633	0.02189	-0.1016	0.08595
24	0.000039	5.002197	0.08814	-0.07	0.01088
25	0.000078	5.006345	-0.0063	-0.0325	0.04098
26	0.000117	4.984926	0.01962	-0.0323	0.07131
27	0.000156	5.005085	0.04951	0.01637	-0.0357
28	0.000195	5.00179	-0.0193	0.06052	-0.0548
29	0.000234	5.002554	0.02694	0.0387	0.02897
30	0.000273	5.002352	0.02714	0.0288	-0.0149
31	0.000313	5.002162	-0.0396	0.00963	-0.0203
32	0.000352	4.995267	0.00544	0.00584	0.05534
33	0.000391	5.00047	-0.0047	0.0186	0.02976

only the X, Y, and Z signal columns are extracted using pandas library.

Figure 4.10: Illustration of file structure.

5.2.2. Amplitude Normalization

Each axis signal is normalized by its peak absolute value to scale amplitudes to [−1, +1], with a small epsilon (1×10−10) added to avoid division by zero. This step reduces bias caused by amplitude variations due to sensor sensitivity and installation conditions.

5.2.3. Signal Windowing and Label Extraction

Signals are divided into overlapping windows (2048 samples, 50% overlap) to increase training samples and improve temporal analysis. Approximately 255 windows are generated per file. Fault type and severity labels are automatically extracted from filenames using pattern matching, while unrecognized files are ignored during training and testing.

5.2.4. Feature Extraction

A total of 33 statistical and spectral features are extracted from each windowed segment of each axis. Because three axes (X, Y, Z) are processed per window, the concatenated feature vector has a dimensionality of 3 × 33 = 99 features.

5.2.5. Time Domain Features

Time domain features were extracted because they reflect the raw vibration signal and capture changes in intensity, stability, and impact like behavior caused by faults.

- **Mean**: used to detect any shift or bias in the signal that may indicate measurement issues or changes in operating conditions.
- **Standard Deviation / Variance:** used to measure how much the vibration varies; this typically increases as mechanical faults develop.
- **Root Mean Square (RMS):** represents the effective energy of the vibration signal and is a strong indicator of bearing degradation.
- **Energy**: measures the overall mechanical activity, which increases with fault severity.
- Skewness: helps detect asymmetry in the signal distribution caused by abnormal behavior.
- **Kurtosis**: used to identify sharp peaks or impulsive events, which are key indicators of bearing defects.
- **Crest Factor:** highlights signals with high peaks relative to their overall energy and is very sensitive to localized faults.

5.2.6. Differential Features

Differential features (first order differences) were extracted because they highlight rapid changes and short duration shock events that are not clearly visible in the original signal.

- They help emphasize high frequency vibrations caused by surface cracks or defects.
- They increase the sensitivity of the system to sudden changes in the signal.
- The sum of absolute differences reflects the total variation of the signal, which increases when surface damage or abnormal friction is present.

5.2.7. Frequency Domain Features

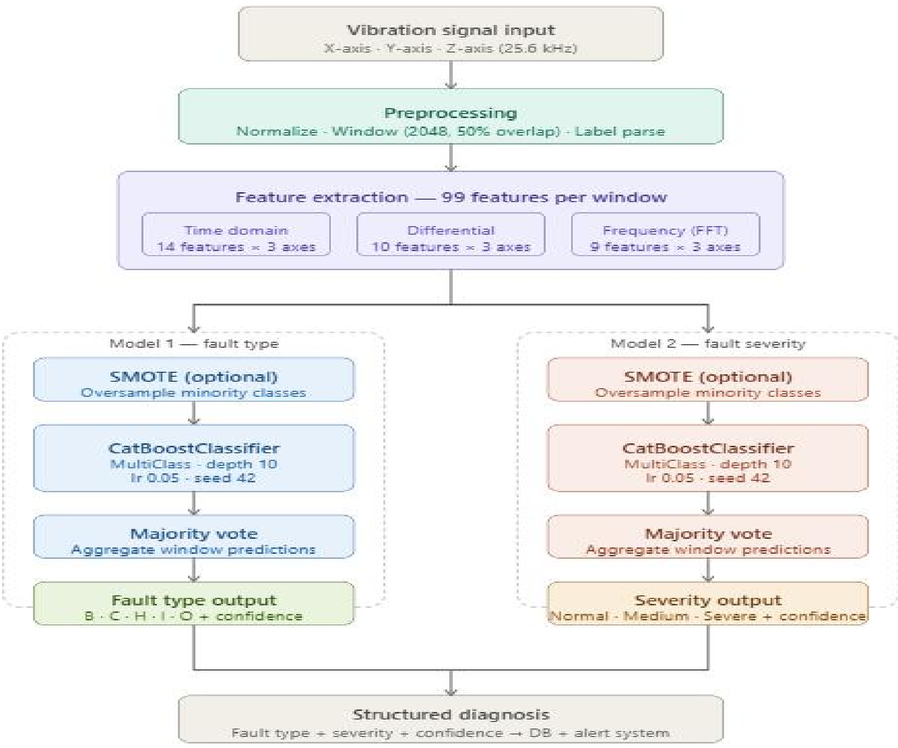
Frequency-domain features were extracted because they describe how the signal energy is distributed across different frequency components, which is essential for understanding the type of fault, not just its severity.

- Dominant Frequency: identifies the main vibration frequency and reflects the overall dynamic behavior of the system.
- Low Frequency Energy: represents normal mechanical operation and stable system behavior.
- High Frequency Energy: captures impacts and fault related vibrations, which are strong indicators of defects.

5.3. Model Training and Evaluation

5.3.1. Model Architecture

Two independent CatBoost gradient boosted classifiers are trained: one for fault type and one for fault severity. CatBoost was selected over other ensemble methods because of its superior handling of tabular data, built-in regularization that reduces overfitting, and efficient training without extensive hyperparameter tuning. The MultiClass loss function is employed



in both models since the prediction tasks involve more than two classes.

Figure 4.11: CatBoost dual model architecture.

5.3.2. Class Imbalance Handling with SMOTE

Because some operating conditions contain fewer healthy samples, SMOTE is option-ally applied to the training set to balance class distribution before model training. It generates synthetic samples for minority classes by interpolating between existing feature vectors and is applied separately to fault type and severity datasets.

5.3.3. Training Procedure

The full feature matrix is partitioned into training and test sets using stratified random splitting, with the test fraction configurable between 10% and 40% (default: 20%). Stratification ensures that each class is proportionally represented in both partitions.

The training procedure for both models follows identical steps:

- Feature extraction is performed across all valid files in the dataset folder.
- Labels are encoded as integers using scikit-learn's LabelEncoder, which also retains the mapping for inverse transformation during inference.
- SMOTE is applied to the training split if enabled.
- The CatBoostClassifier is trained on the (possibly oversampled) training data.
- The trained model is evaluated on the held-out test set using accuracy and the confusion matrix.
- Models and label encoders are serialized to disk using joblib for subsequent deployment.

5.3.4. Majority Vote Aggregation and Confidence Scoring

Each file is split into multiple windows, and the final label is obtained by majority voting across window predictions. CatBoost models output class probabilities using predict_proba, where the maximum probability is used as the confidence score.

5.4. Testing Modules and Evaluation

The Testing tab processes labeled vibration files, performs window-based inference, and aggregates results using majority voting. It then computes accuracy and confusion matrices for both fault type and severity.

5.4.1. Fault Type Classification Results

➢ **Overall Accuracy:**

The fault type prediction model achieved an accuracy of 99.98%, demonstrating an extremely high ability to correctly identify the fault category among the five considered classes (B, C, H, I, and O).

➢ **Confusion Matrix Analysis:**

The confusion matrix further supports the strong performance of the model. Almost all samples were classified correctly, with only a single misclassification observed between the C and O classes. This indicates that the learned decision boundaries are highly discriminative, and the model is capable of accurately separating different fault conditions.

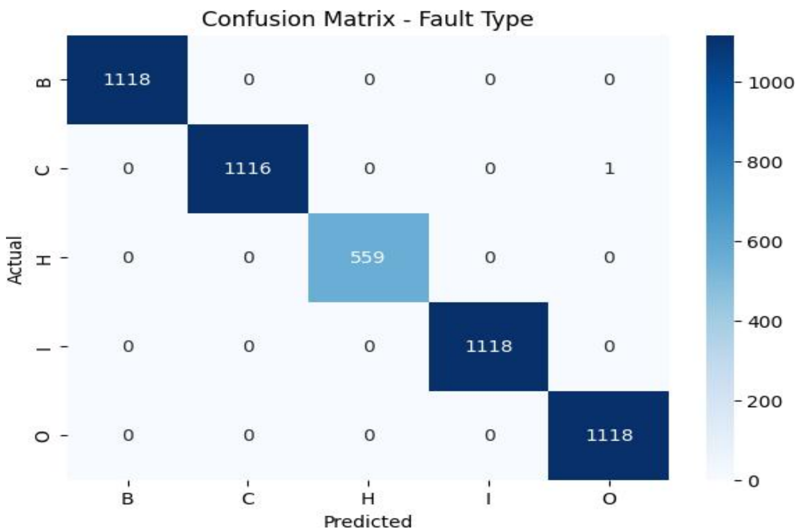


Figure 4.12 : Confusion Matrix for Fault Type Classification.

5.4.2. Fault Severity Classification Results

➢ **Overall Accuracy:**

For fault severity prediction, the model achieved an accuracy of 99.96%, showing excellent performance in distinguishing between the three severity levels (Normal, Medium, and Severe).

➢ **Confusion Matrix Analysis:**

The confusion matrix indicates that the severity model produces highly accurate predictions. Only a very small number of errors occurred, where two samples belonging to the Severe class were predicted as Medium. Despite this minor confusion, the overall performance remains highly stable and confirms the effectiveness of the trained model.

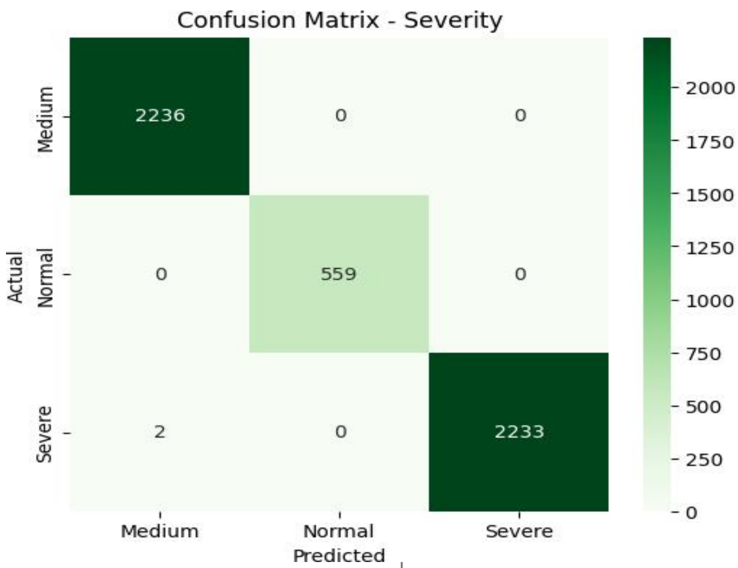


Figure 4.13: Confusion Matrix for Fault Severity Classification

6. Implementation of the electricity load forecasting and optimization energy

6.1. Dataset Description

The Electricity Load Forecasting dataset is a public hourly time-series dataset from Kaggle (https://www.kaggle.com/datasets/saurabhshahane/electricity-load-forecasting) used for energy demand forecasting.

Its main target variable, nat_demand, represents national electricity consumption in Megawatts (MW). The dataset also includes weather related features such as temperature, precipitation, and wind speed that influence electricity demand.

6.1.1. Data Source and Acquisition

The dataset was collected from a national electricity grid operator and published on Kaggle as an open-access CSV file named continuous dataset.csv. Each row represents one hourly observation indexed by datetime. The data spans multiple years, making it suitable for analyzing seasonal and long-term electricity demand patterns.

6.1.2. Dataset Structure and Variables

The raw dataset contains the following principal columns:

Variable	**Type**	**Description**
datetime	DateTime Index	Hourly timestamp (index after loading)
nat_demand	Float (MW)	National electricity demand — forecasting target
T2M_toc	Float (°C)	Air temperature at 2m height
QV2M_toc	Float (g/kg)	Specific humidity at 2m

TQL_toc	Float (kg/m²)	Total column liquid water
W2M_toc	Float (m/s)	Wind speed at 2m height

Table 4.9: Principal Variables of the Electricity Load Forecasting Dataset

After loading the dataset, the datetime column was converted into a panda DatetimeIndex and the data was sorted chronologically. The nat_demand variable contained no missing values, providing a clean continuous time series suitable for feature extraction.

6.1.3. Statistical Characteristics of the Target Variable

The national electricity demand exhibits clear daily, weekly, and annual seasonal patterns. Demand is higher during daytime working hours, increases on weekdays compared to weekends, and rises during winter and summer due to heating and cooling usage. These recurring patterns support the use of cyclical time encoding and lag-based feature engineering.

6.2. Data Preprocessing and Feature Engineering

6.2.1. Temporal Indexing and Sorting

The CSV file is loaded using pandas with parse_dates=["datetime"], and the datetime column is set as the DataFrame index. The dataset is then sorted chronologically to ensure strict time ordering, which is essential for lag-based and rolling-window feature construction. Maintaining correct temporal order prevents data leakage and ensures valid forecasting evaluation.

6.2.2. Cyclical Time Encoding

Time variables such as hour, day of week, and month are encoded using sine and cosine transformations to preserve their cyclical nature. This allows the model to capture periodic patterns more effectively than raw integer encoding. Additional features like is_weekend, quarter, and hour_of_week are included to represent broader temporal and business-cycle effects.

6.2.3. Lag Features

Electricity demand is highly dependent on recent and seasonally aligned past values. To capture this dependency, lag features are created by shifting nat_demand by 1, 2, 3, 6, 12, 24, 48, 72, 168, and 336 hours. In particular, the 24 hour and 168-hour lags capture daily and weekly seasonality, while the 336-hour lag reflects bi-weekly repeating patterns.

6.2.4. Rolling-Window Statistical Features

Rolling statistics are computed over 24 hour and 168-hour windows to capture short- and medium-term trends and variability. For each window, the mean, standard deviation, minimum, and maximum are extracted, providing a compact representation of recent demand dynamics without explicit autoregressive modeling.

6.2.5. Exponential Moving Averages

Three exponential moving averages (EMA) with spans of 12, 24, and 168 hours are added. Unlike simple rolling means, EMAs apply exponentially decreasing weights to older observations (adjust =False), making them more responsive to recent demand shifts. They complement the rolling statistics by providing smoother trend estimates that react faster to structural changes in the series.

6.2.6. Demand Velocity Features

First order differences of the demand signal are computed at lags of 1 hour (demand_diff_1) and 24 hours (demand_diff_24). These velocity features capture the rate and direction of demand change and help the model anticipate sudden load variations that are not encoded in the level-based lag features.

6.2.7. Interaction Features

Two cross-product interaction terms are constructed. The feature hour_x_dow, defined as the product of the integer hour and day-of-week, encodes the joint distribution of daily and weekly demand peaks for example: Monday morning rush-hour versus Sunday evening which is not fully captured by either component individually.

The feature lag1_x_roll24 multiplies the one-hour lag by the 24-hour rolling mean, providing a non-linear amplification signal that penalizes predictions when both the immediate past and the recent average are simultaneously elevated.

6.2.8. Missing Value Removal

All feature construction operations that look backwards in time (lags, rolling windows, EMAs) introduce NaN values at the beginning of the series where insufficient history exists. A single dropna() call is applied after all feature columns have been computed, removing the initial rows uniformly across all columns. This ensures that the feature matrix passed to the models is complete and consistent with no imputation artifacts.

6.2.9. Final Feature Vector

After preprocessing, each observation is represented by a feature vector comprising cyclical time encodings, lag features, rolling statistics, exponential moving averages, velocity features, and interaction terms. Together with the meteorological covariates present in the original dataset, the final input dimensionality is sufficient to capture the dominant sources of variation in national electricity demand.

6.3. Dataset Partitioning

6.3.1. Temporal Train / Validation / Test Split:

Because electricity demand is a time series, random shuffling of observations would violate temporal causality and introduce future information leakage into the training set. A strict chronological split is therefore applied. The full post-preprocessing dataset of N observations is divided into three non-overlapping, ordered partitions:

Partition	Proportion	Purpose
Training Set	70% (first 0.70 × N)	Hyperparameter search and base-model fitting
Validation Set	15% (0.70–0.85 × N)	Optuna objective evaluation; meta-learner training
Test Set	15% (last 0.15 × N)	Final unbiased model evaluation

Table 4.10: Dataset Partitioning Strategy

A combined train + validation set (train-val, covering the first 85% of the series) is assembled for final base-model fitting after hyperparameter selection, following the standard practice of using the full available history before the test horizon. The test set is held out and never used during any stage of model selection or tuning.

6.3.2. Time-Series Cross-Validation

Within the training set, 5-fold time-series cross-validation (TimeSeriesSplit, n_splits=5) is employed during Bayesian hyperparameter optimization. Each fold strictly preserves temporal ordering: the validation fold is always later in time than the training fold, preventing any form of look-ahead bias. The objective minimized across folds is the mean Root Mean Squared Error (RMSE), which penalizes large prediction errors more heavily than the mean absolute error.

6.4. Model Architecture: Stacked Gradient Boosting Ensemble

6.4.1. Architecture Overview

The forecasting system uses a two-level stacking ensemble. The first level consists of three gradient boosting models: XGBoost, LightGBM, and CatBoost. Their outputs are combined by a Ridge regression model at the second level to produce the final prediction. A bias-correction step is applied to reduce systematic errors, leveraging the strengths of each model while limiting overfitting through regularization.

6.4.2. Base Estimator 1-XGBoost:

XGBoost (eXtreme Gradient Boosting) is a regularized gradient boosting framework that constructs an ensemble of decision trees using second-order optimization. The model is configured for regression with a squared error objective and uses histogram-based tree building for efficiency. Regularization is applied through L2 penalty and minimum child weight constraints, while subsampling of rows and features helps reduce overfitting and improve generalization.

6.4.3. Base Estimator 2-LightGBM:

LightGBM (Light Gradient Boosting Machine) is a gradient boosting framework that uses a leaf-wise tree growth strategy, enabling deeper and more efficient trees compared to level-wise approaches. It improves efficiency through Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB). Model complexity is mainly controlled via the number of leaves, and early stopping is used to prevent overfitting during training.

6.4.4. Base Estimator 3-CatBoost:

CatBoost is a gradient boosting framework based on symmetric (oblivious) decision trees and ordered boosting, which helps reduce target leakage during training. It is configured with fixed hyperparameters (iterations=1200, learning_rate=0.02, depth=7) and uses subsampling of both samples and features. Early stopping is applied to avoid overfitting. It complements XGBoost and LightGBM by providing a different tree-building strategy and improving model diversity.

6.4.5. Hyperparameter Optimization with Optuna

Hyperparameter optimization for XGBoost and LightGBM is performed using the Optuna framework with the Tree-structured Parzen Estimator (TPE) sampler. Each model is trained over 40 trials and evaluated using 5-fold time-series cross-validation, where the mean RMSE is minimized to select the best parameter configuration. The search space is defined over the parameters listed in Table 4.11.

Parameter	**XGBoost Range**	**LightGBM Range**
n_estimators	600 - 1500	600 -1500
max_depth	4 - 9	4 - 9
learning_rate	0.005 - 0.05 (log)	0.005 - 0.05 (log)

subsample	0.70 - 1.00	0.70 - 1.00
colsample_bytree	0.60 - 1.00	0.60 – 1.00
reg_lambda	0.50 - 5.00	0.50 - 5.00
gamma	0.00 - 0.50	--
min_child_weight	1 - 10	--
num_leaves	--	31 - 127
min_child_samples	--	10 - 50

Table 4.11: Hyperparameter Search Spaces for XGBoost and LightGBM

6.4.6. Meta-Learner and Bias Correction

The predictions of XGBoost, LightGBM, and CatBoost are combined into a meta-feature matrix and used to train a Ridge regression meta-learner with L2 regularization. The model learns optimal weights for combining the base predictions while reducing systematic errors. A final bias-correction step is applied by adding the mean validation residual to the test predictions, helping reduce consistent over- or under-prediction.

6.5. Experimental Results

6.5.1. Model Performance Comparison

Table 4.12 and Figure 4.14 present the test-set performance of all five model variants across the four evaluation metrics.

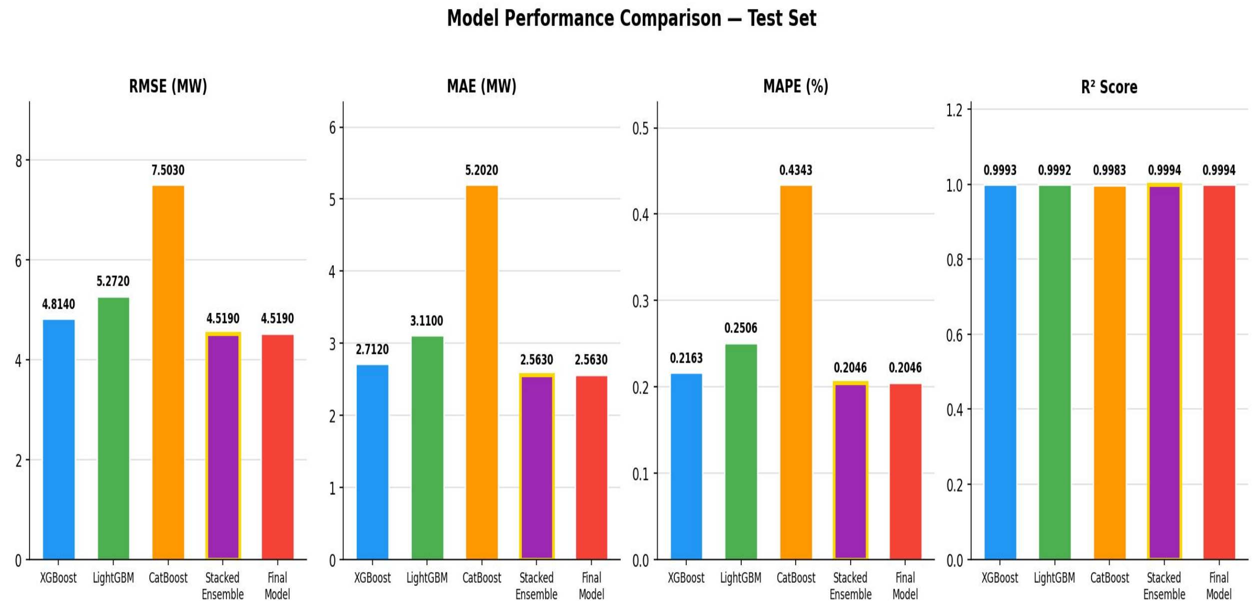
Model	RMSE (MW)	MAE (MW)	MAPE (%)	R^2 Score
XGBoost	4.814	2.712	0.2163	0.99932
LightGBM	5.272	3.110	0.2506	0.99918

CatBoost	7.503	5.202	0.4343	0.99835
Stacked Ensemble	4.519	2.563	0.2046	0.99940
Final Model	4.519	2.563	0.2046	0.99940

Table 4.12: Test-Set Performance Comparison of All Model Variants

Figure 4.14: Bar chart comparison of RMSE, MAE, MAPE, and R^2 across all models. Gold-bordered bars indicate best values.

Several key observations emerge from the comparison:



- XGBoost achieves the best single model RMSE of 4.814 MW and R^2 of 0.99932, confirming that its level-wise tree growth with second order gradient regularization is well suited to the smooth, autocorrelated demand signal.
- LightGBM scores 5.272 MW RMSE, 9.5% higher than XGBoost suggesting that its leaf wise growth, while faster, introduces slightly more variance on this dataset. However, its R^2 of 0.99918 remains exceptional.
- CatBoost, trained with fixed hyperparameters rather than Optuna optimization, is the weakest individual model (RMSE=7.503, MAPE=0.4343%). This is expected and does not diminish its role as a diversifying third perspective in the ensemble.
- The Stacked Ensemble reduces RMSE to 4.519 MW a 6.1% improvement over the best individual model (XGBoost). This reduction confirms that LightGBM and CatBoost capture complementary variance that XGBoost alone misses.

- All five models achieve $R^2 > 0.998$, confirming that gradient boosting with temporal feature engineering captures over 99.8% of demand variance a result consistent with state of the art load forecasting literature.

6.5.2. Meta-Learner Blend Weights

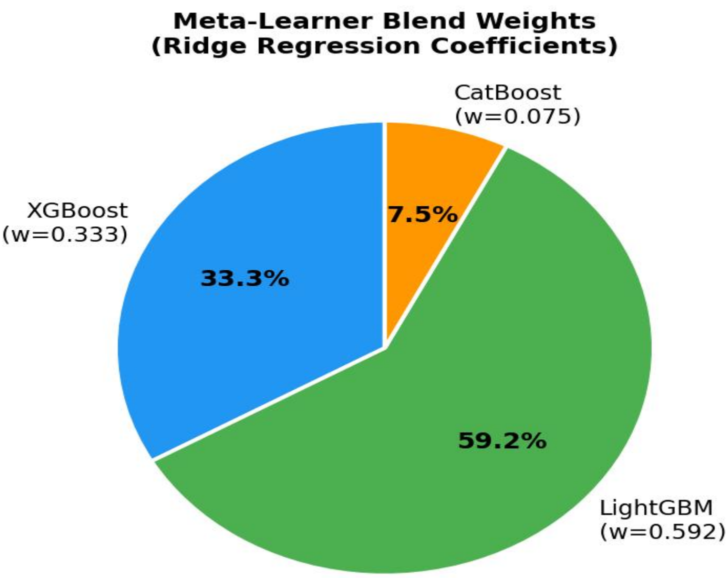
The Ridge meta-learner assigns the following coefficients to the three base models on the validation partition:

Base Model	Blend Weight	Contribution (%)
XGBoost	0.333	33.3%
LightGBM	0.592	59.2%
CatBoost	0.075	7.5%

Table 4.13: Ridge Meta-Learner Coefficients

Figure 4.15: Pie chart of meta-learner blend weights showing the relative contribution of each base model.

LightGBM receives the highest ensemble weight (59.2%), followed by XGBoost (33.3%) and CatBoost (7.5%). Although XGBoost achieves a lower individual RMSE, the



meta-learner gives more importance to LightGBM because its predictions better complement the errors of the other models. CatBoost receives a smaller weight due to its lower standalone performance, but its different architecture still contributes useful corrective information to the ensemble.

6.5.3. Generalisation Analysis

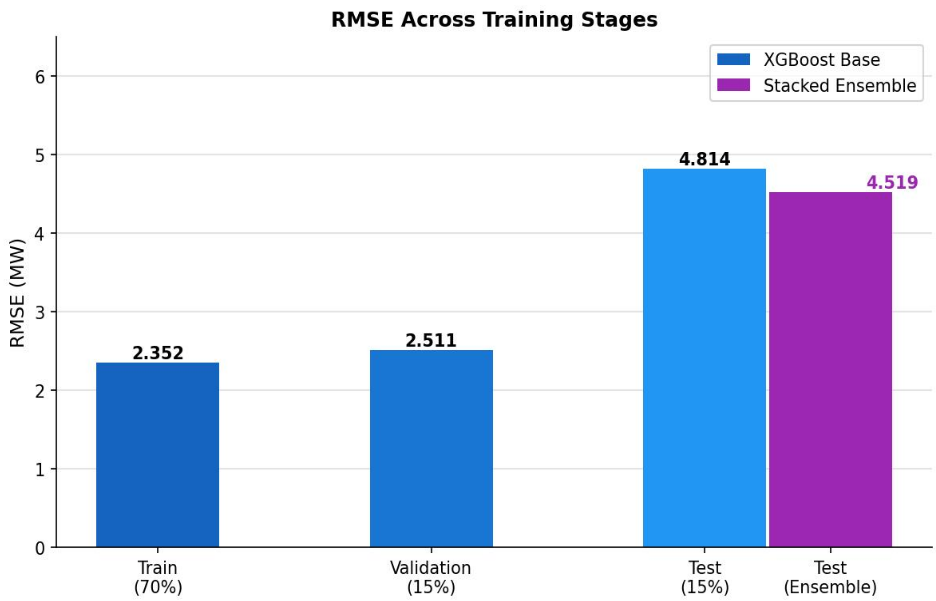


Figure 4.16: RMSE progression from training to validation to test sets for XGBoost base model and the stacked ensemble.

Figure 4.16 shows the generalization performance of the XGBoost model and the stacked ensemble. The train RMSE is 2.352 MW, increasing slightly to 2.511 MW on validation, indicating effective overfitting control during hyperparameter optimization. The test RMSE rises to 4.814 MW due to temporal distribution changes in the final portion of the dataset. The stacked ensemble improves performance by reducing test RMSE to 4.519 MW. Considering the average demand of about 1198 MW, the relative prediction error remains very low (≈0.38%), demonstrating strong generalization capability.

6.5.4. Advanced Error Metrics -Final Ensemble Model

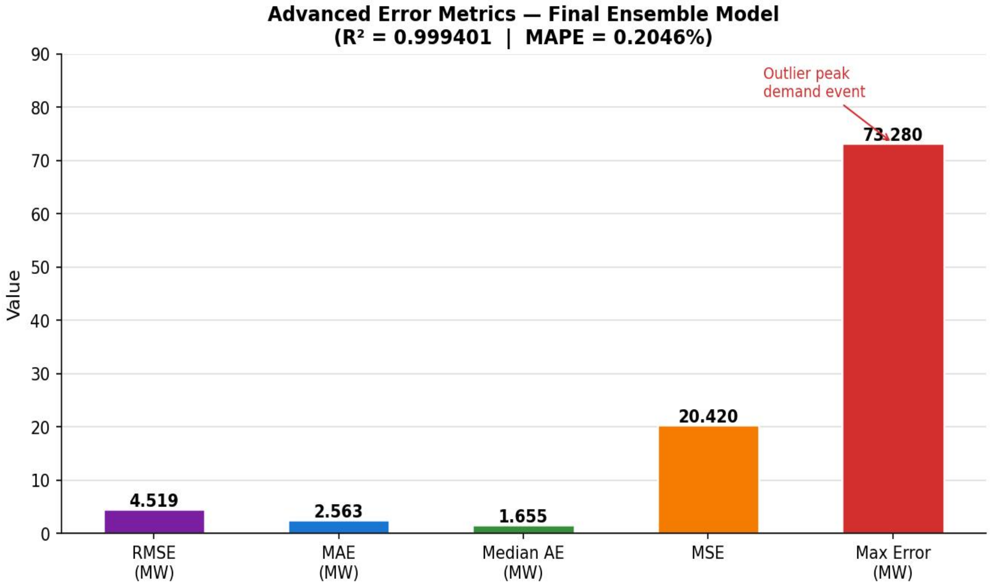


Figure 4.17: Advanced error metrics for the final ensemble model. The red bar (Max Error = 73.28 MW) corresponds to rare peak-demand events.

The advanced metrics provide a detailed evaluation of the final model's errors. The median absolute error of 1.655 MW is lower than the MAE of 2.563 MW, indicating that most predictions are very close to the true demand, while a small number of larger errors increase the mean error. The maximum error of 73.28 MW occurs during rare peak-demand events, likely caused by sudden weather changes or unusual load conditions that were not well represented in the training data. The model achieved a MAPE of 0.2046%, which indicates excellent forecasting accuracy according to industry standards. In addition, the R^2 score of 0.999401 shows that the model explains 99.94% of the variance in national electricity demand through its feature engineering and ensemble architecture.

Metric	Value	Interpretation
RMSE	4.519 MW	Primary accuracy metric < 0.38% of mean demand
MAE	2.563 MW	Average prediction error < 0.21% of mean demand
Median Absolute Error	1.655 MW	Typical error for 50% of predictions
MAPE	0.2046%	Excellent industry benchmark < 0.5%
Max Error	73.28 MW	Extreme peak events; 6.1% of mean demand
R^2 Score	0.999401	99.94% of demand variance explained
Explained Variance	0.999401	Confirms absence of systematic bias after correction

Table 4.14: Complete Error Metrics for the Final Ensemble Model

6.5.5. Grid Optimization Results

The forecasted demand values from the final ensemble model are fed into a Unit Commitment and Economic Dispatch module to demonstrate the operational value of accurate load prediction. The generation fleet totals 2450 MW across nine units ranging from base-load (150 MW, 28 $/MWh) to peaking (450 MW, 90 $/MWh), with a mandatory 15% spinning reserve margin.

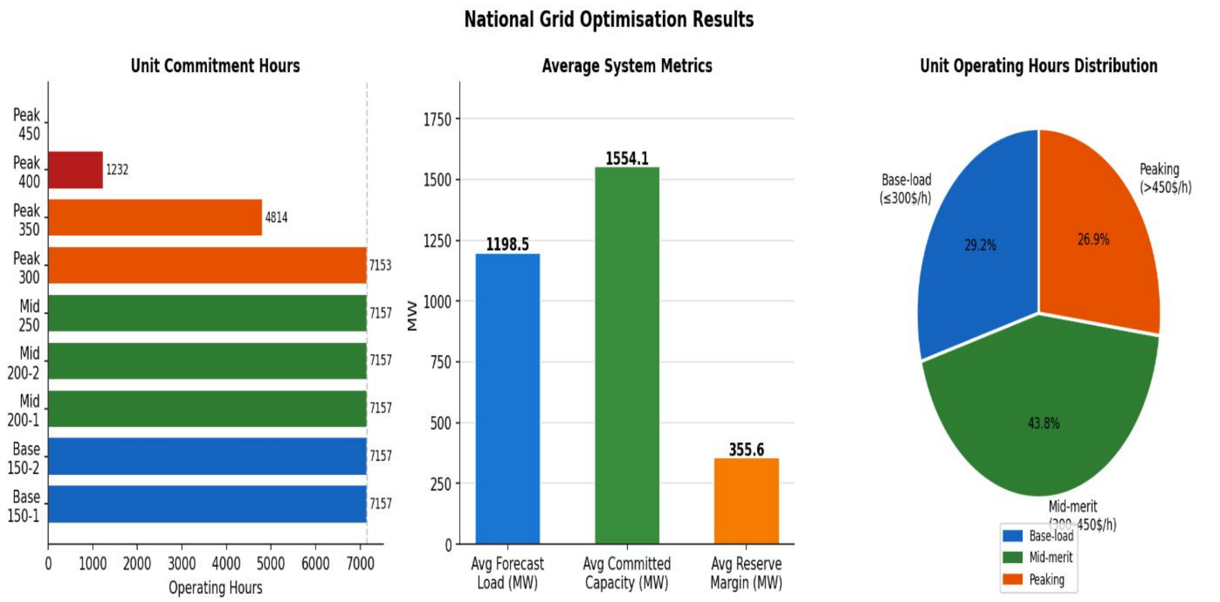


Figure 4.18: Grid optimization results (left) operating hours per unit, (center) average system metrics, (right) unit operating hour distribution by category.

6.5.6. Operational Metrics

Metric	Value	Significance
Average Forecasted Load	1,198.54 MW	48.9% of total installed capacity (2,450 MW)
Average Committed Capacity	1,554.11 MW	Includes 15% reserve margin above forecast
Average Reserve Margin	355.57 MW	29.7% above average load reliable buffer
Average System Utilisation	77.45%	High utilization efficient fleet loading
Maximum Power Shortage	0.00 MW	Zero curtailment across entire test period
Average Operating Cost	69,857 $/h	Weighted average of committed unit costs
Maximum Operating Cost	102,150 $/h	Peak-demand hours requiring all 8 large units

Table 4.15: National Grid Optimization Performance Metrics

6.5.7. Unit Commitment Analysis

The analysis of Figure 4.18 shows the generation commitment hierarchy driven by demand levels. The cheapest base-load units operate almost continuously (7,157 hours), as the average demand (1,198 MW) exceeds their combined capacity (950 MW). Mid-cost units are also frequently dispatched, while peak units are activated only during higher demand periods, with the most expensive unit never being used, indicating that demand rarely reaches extreme capacity limits.

The system maintains a strong safety margin, with an average reserve of 355.57 MW (29.7%), exceeding the required threshold. No supply shortages occur throughout the test period, confirming the reliability of the demand forecasts. Overall, base and mid-merit units dominate operation (63%), while peaking units cover demand variability (37%), ensuring both economic efficiency and system stability.

6.6. Discussion

6.6.1. Key Findings

The experimental results lead to three main conclusions. First, the stacked gradient boosting ensemble achieves very high predictive performance on the national electricity demand dataset, with a MAPE of 0.2046% and an R^2 of 0.99940. This confirms that tree-based ensemble models combined with effective temporal feature engineering can match or exceed the performance of more complex deep learning approaches for hourly load forecasting.

Second, the stacking strategy consistently improves performance over individual models, achieving a 6.1% reduction in RMSE compared to the best single estimator. This demonstrates that XGBoost, LightGBM, and CatBoost capture complementary patterns in the demand signal, which are effectively integrated by the meta-learner.

Third, the high forecasting accuracy translates directly into operational reliability in power system management. When the model's predictions are used for unit commitment decisions, no power shortages occur across the test period, confirming its suitability for grid operation support.

6.6.2. Limitations

The proposed approach has several limitations. It relies heavily on lag-based features, which require continuous availability of historical demand data in real-time applications. Although rare, the observed maximum error of 73.28 MW indicates reduced robustness under extreme peak-demand conditions that are underrepresented in the training data. Furthermore, the model was validated on a single national grid dataset; therefore, its generalization to other systems with different demand structures, such as high renewable penetration or distinct climatic conditions, may require retraining or domain adaptation.

7. Unified Modeling Language (UML)

The Unified Modeling Language (UML) is a standardized visual modeling language used to specify, analyze, design, and document software systems. It provides a common graph- ical notation that facilitates communication among software engineers, developers, re- searchers, and stakeholders throughout the software development lifecycle. UML enables the representation of both the structural and behavioral aspects of a system, thereby improving understanding, maintainability, and system quality.

7.1. Objectives of UML

The main objectives of UML are summarized as follows:

- To understand and specify system requirements.
- To represent the dynamic behavior of the system.
- To model the internal structure and architecture of the system.
- To facilitate communication among developers, analysts, researchers, and stake- holders.

7.2. Types of UML Diagrams

UML diagrams are generally classified into two major categories:

- **Behavioral Diagrams**
  - Use Case Diagram
  - Sequence Diagram
- **Structural Diagrams**
  - Class Diagram

7.3. Diagrams Used in the Implementation

To model and document the proposed intelligent framework, three UML diagrams were employed during the design phase. These diagrams provide complementary perspectives of the system, enabling a comprehensive understanding of its functional requirements, dynamic interactions, and structural organization.

- Use Case Diagram
- Sequence Diagram
- Class Diagram

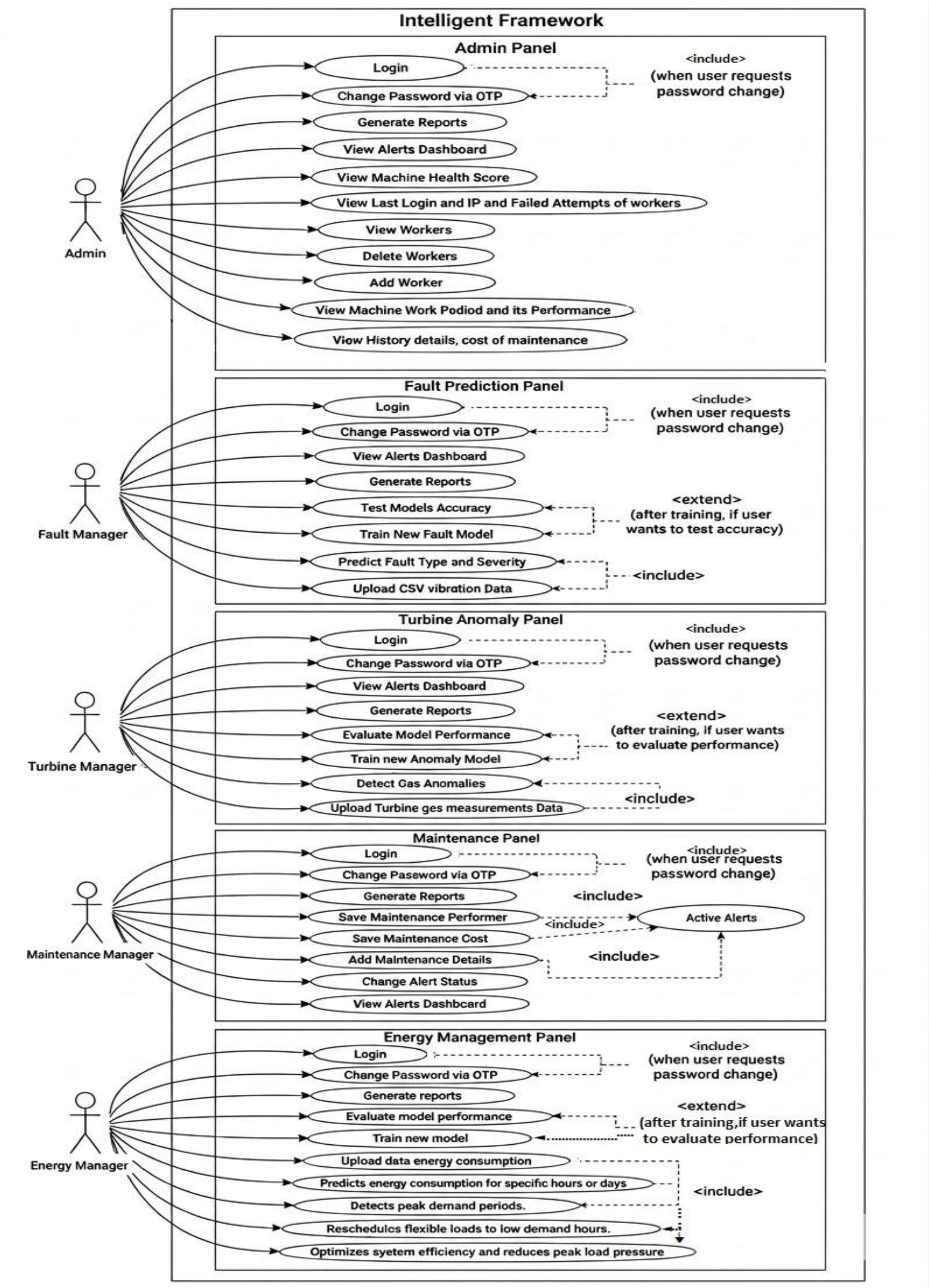
7.3.1. Use Case Diagram

A Use Case Diagram is a behavioral UML diagram that captures the functional requirements of the intelligent framework from the perspective of its external actors. It describes what the system does rather than how it performs its functionalities by illustrating the interactions between users and the services provided by the system.

Two important relationships are used within the diagram:

- **«include»**: Represents a mandatory functionality that is always executed as part of another use case.

- **«extend»**: Represents an optional functionality that is executed only under specific



conditions.

Figure 4.19: Use Case Diagram of the Intelligent Framework

Actors and Their Roles

The intelligent framework involves five distinct actors, each associated with a dedicated panel and a specific set of responsibilities.

Actor / Role	Primary Responsibilities
Admin	Manages workers (add, delete, and view), monitors machine health scores, views login history and failed authentication attempts, monitors machine work periods, performance and maintenance cost history, generates reports, and accesses the global alerts dashboard.
Fault Manager	Uploads CSV vibration data, predicts fault type and severity using machine learning models, train new fault detection models, evaluates model accuracy, and generates prediction reports.
Turbine Manager	Uploads turbine gas measurement data, detects gas anomalies using a deep learning model, trains anomaly detection models, evaluates model performance, and generates analytical reports.
Maintenance Manager	Views active alerts, records maintenance interventions including performer, cost and description, updates alert status (Pending, In Progress, Resolved), and generates maintenance reports.
Energy Manager	Uploads electricity load data, trains forecasting models, predicts future energy consumption, identifies peak demand periods, reschedules flexible loads to off-peak periods, optimizes energy usage, and generates energy reports.

Table 4.16: Actors and Their Roles in the System Architecture

All actors share three common use cases across the system:

- Login
- Change Password via OTP Verification
- Generate Report

The password recovery process is modeled as an *include* relationship since it is only executed when explicitly requested by the user. [53]

7.3.2. Sequence Diagram

A Sequence Diagram is a UML interaction diagram that models the dynamic behavior of a system by showing how actors and system components communicate over time. Messages are represented chronologically, allowing the visualization of interactions between users, system modules, and intelligent models during the execution of system processes.

1) Actors

The sequence diagram involves the following participants:

- Admin
- Worker
- Maintenance Staff
- Intelligent System
- Machine Learning and Deep Learning Models

2) Interaction Flows

➢ **Authentication Flow**

Every actor initiates a session by submitting login credentials or requesting a password reset through OTP verification. The Intelligent System validates the provided information and returns an access confirmation upon successful authentication.

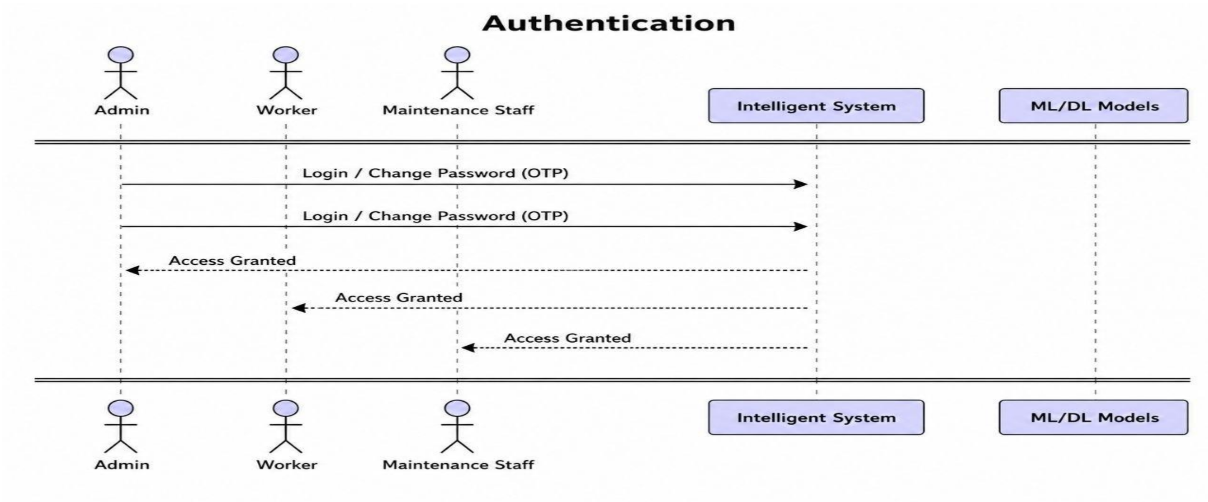
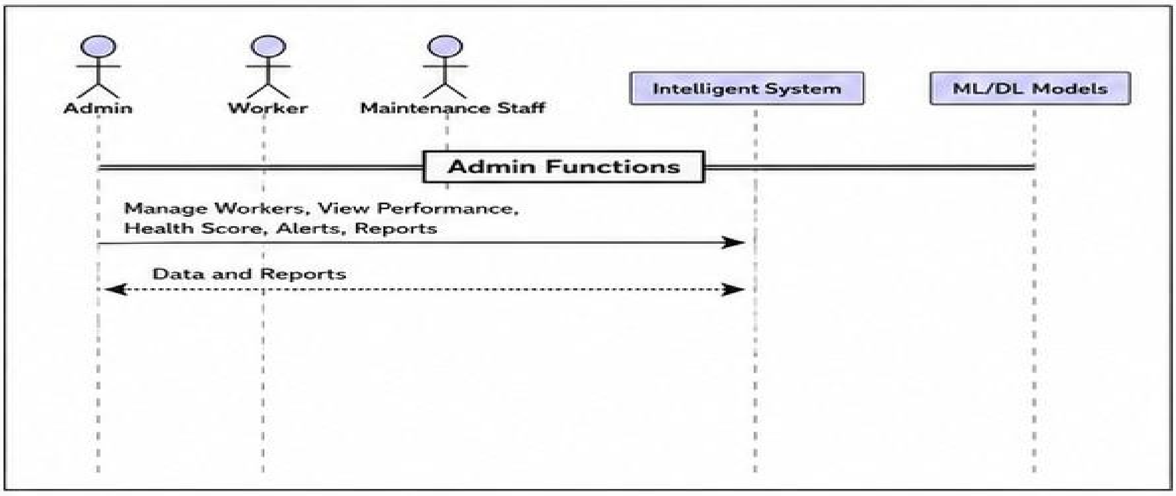


Figure 4.20: Authentication Sequence Diagram

➢ **Administrative Monitoring Flow**

The admin requests system information from the Intelligent System. The system retrieves machine health scores, maintenance history, worker activity logs, and generated reports. The retrieved information is then presented to the admin through the monitoring

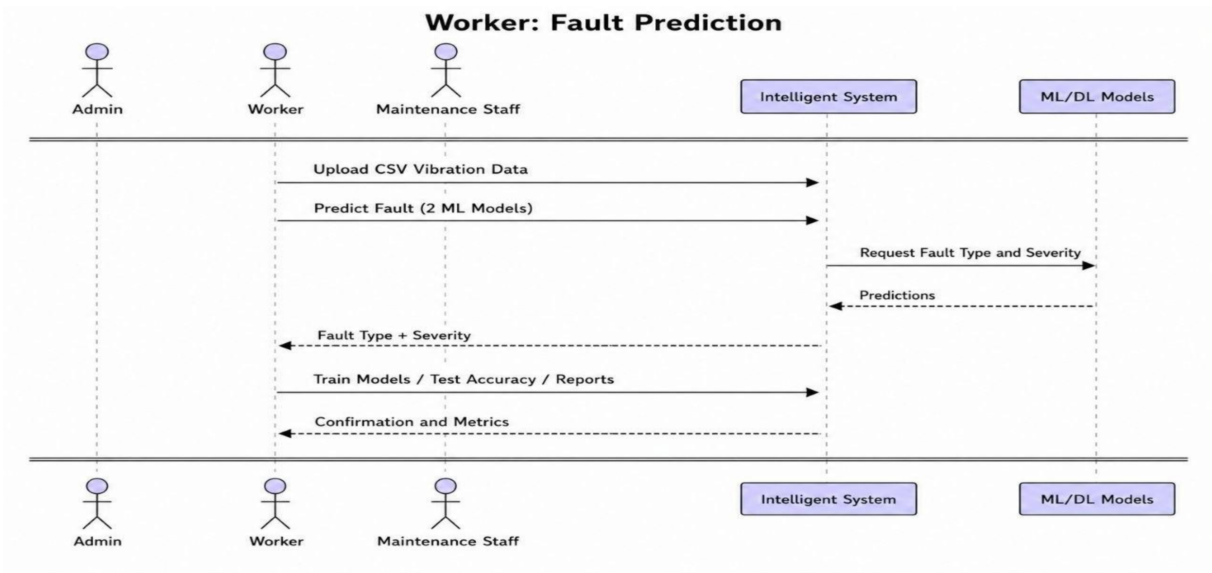


dashboard, enabling system supervision and decision making.

Figure 4.21: Administrative Monitoring Sequence Diagram

➢ **Fault Prediction Flow**

The worker uploads a CSV file containing vibration sensor readings. The Intelligent System forwards the data to two machine learning models responsible for fault prediction. The models return the predicted fault type and severity level. The worker may also initiate



model training or model evaluation and receive performance metrics as feedback.

Figure 4.22: Fault Prediction Sequence Diagram

➢ Energy Prediction and Optimization Flow

The worker uploads electricity consumption data. The Intelligent System requests future load predictions from the energy forecasting model. Based on the obtained results, peak demand periods are identified, and flexible loads are shifted to off-peak periods. The optimized schedule is returned to improve energy efficiency and reduce operational costs.

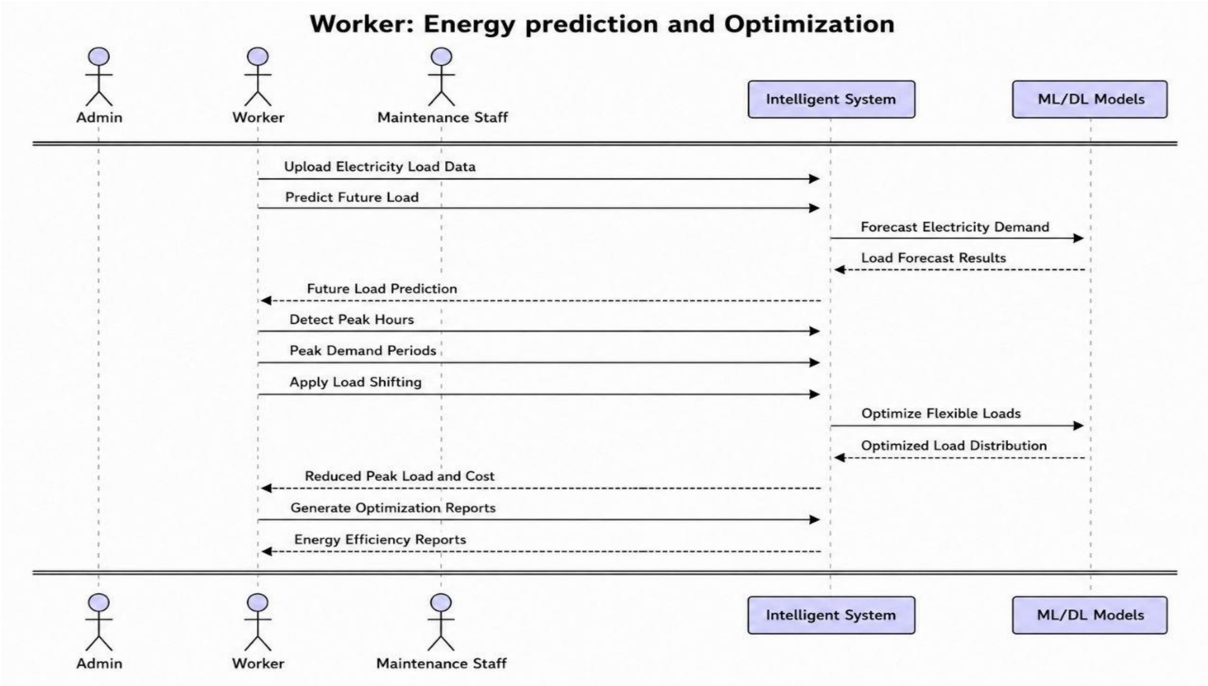


Figure 4.23: Energy Prediction and Optimization Sequence Diagram

➢ Turbine Anomaly Detection Flow

The worker uploads turbine gas measurement data. The Intelligent System forwards the measurements to the deep learning anomaly detection model. The model returns an anomaly score and anomaly status. The worker may additionally train or evaluate the model and receive corresponding performance metrics.

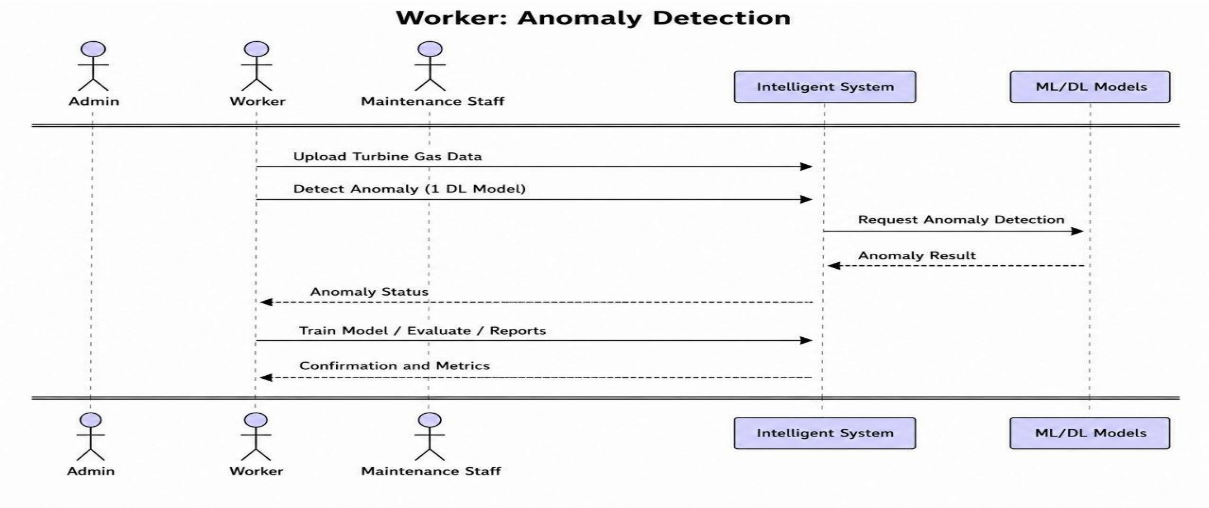


Figure 4.24: Turbine Anomaly Detection Sequence Diagram

- **Maintenance Management Flow**

Maintenance staff members view active alerts generated by the intelligent models, update alert statuses, record intervention details, and generate maintenance reports. This process is purely operational and does not involve direct interaction with machine learning or deep learning models. [53]

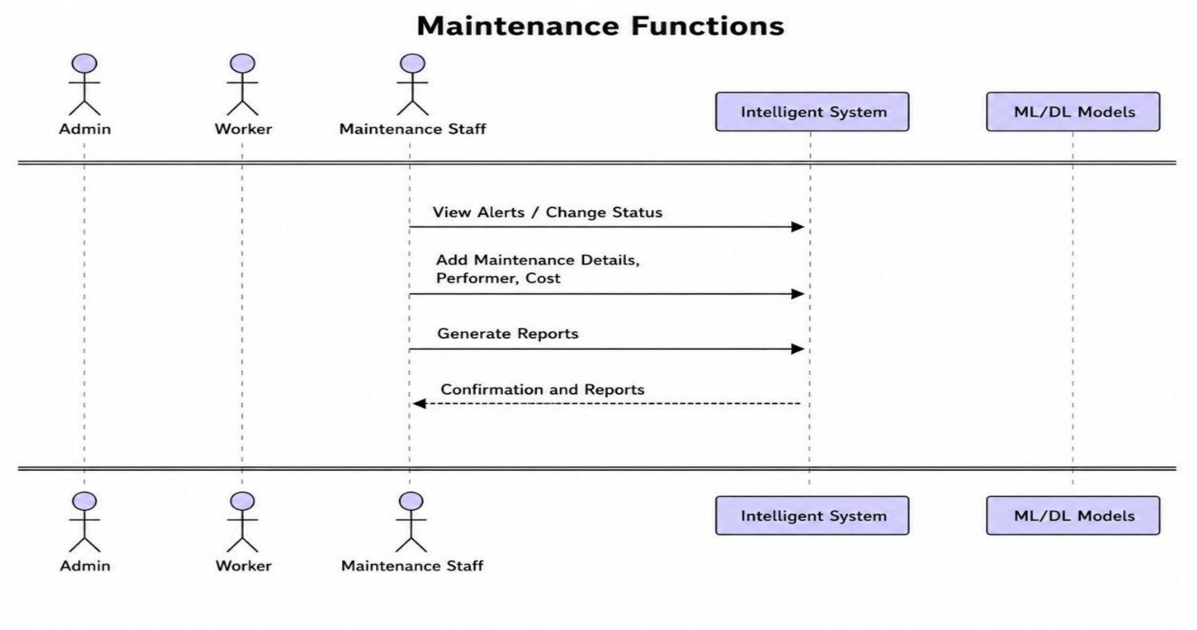
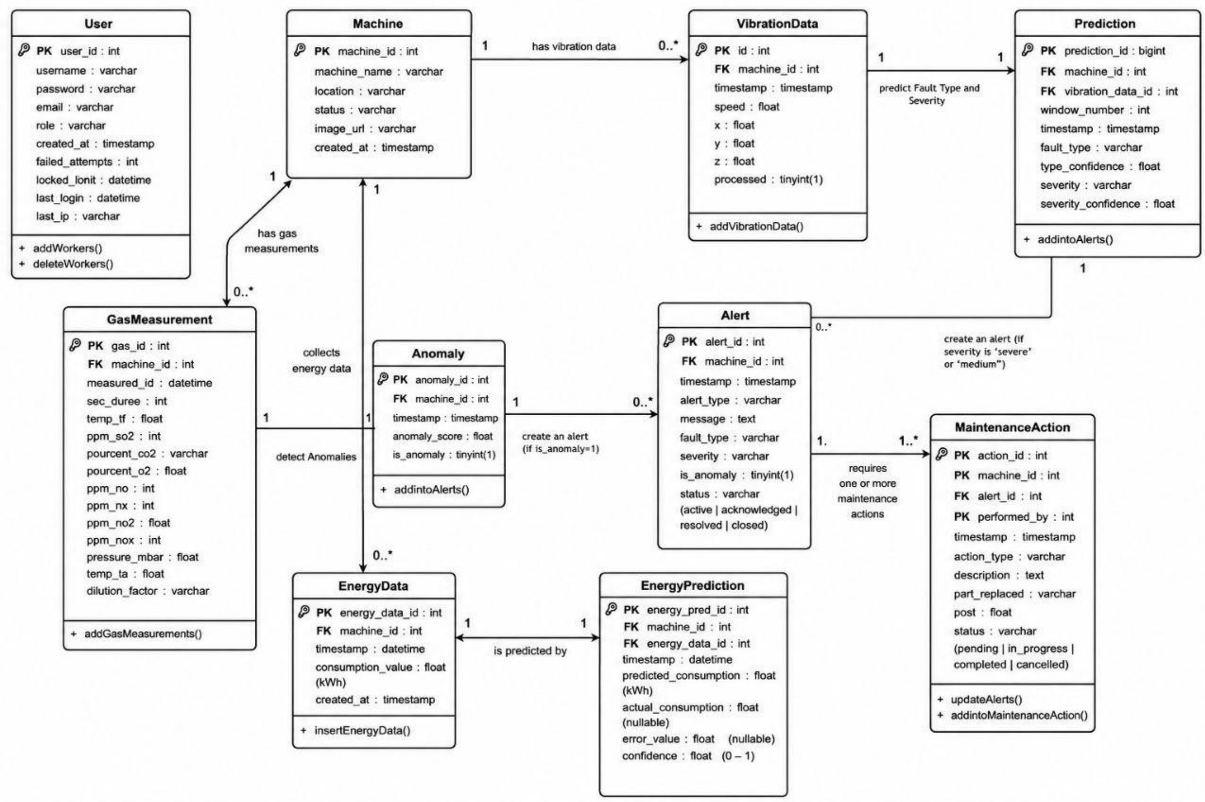


Figure 4.25: Maintenance Management Sequence Diagram

7.3.3. Class Diagram

A Class Diagram is a structural UML diagram that describes the static architecture of a system. It defines the classes composing the system, their attributes, methods, and relationships, thereby serving as the foundation of the software architecture and database



design.

Figure 4.26: Class Diagram of the Intelligent Framework

- **System Classes**

The intelligent framework is structured around ten main classes distributed across users, machines, sensor data, prediction results, alerts, and maintenance operations.

Class Name	Description
User	Represents all system users. Stores authentication credentials, roles, login history, failed attempts, and lock status. Provides worker management functionalities.
Machine	Represents an industrial machine and stores machine information, including name, location, operational status, image URL, and creation timestamp.
VibrationData	Stores raw vibration sensor readings, including speed values and X, Y, and Z axis measurements collected from industrial equipment.
Prediction	Stores fault prediction results generated by machine learning models, including fault type, severity level, confidence scores, and vibration window number.
GasMeasurement	Stores turbine gas sensor readings, including temperatures, gas concentrations, pressure values, and dilution factors.
Anomaly	Stores anomaly detection results produced by the deep learning model, including anomaly score and anomaly status.
EnergyData	Stores electricity consumption measurements and timestamps collected from monitored machines.
EnergyPrediction	Stores energy forecasting outputs, including predicted consumption, actual consumption, prediction error, confidence score, and execution status.
Alert	Represents alerts automatically generated when prediction severity or anomaly thresholds are exceeded. Stores machine references, timestamps, alert messages, and lifecycle status information.
MaintenanceAction	Records maintenance interventions linked to alerts. Stores action type, description, replaced components, intervention cost, performer, and execution status.

Table 4.16: Class Diagram Overview: System Entities and Their Responsibilities

- **Key Design Decisions**

- Role based access control is implemented through a single role attribute within the User class, simplifying authentication and authorization management.
- Each sensor domain follows a consistent architecture where raw data is processed by intelligent models that generate prediction or anomaly results.
- The Alert class serves as the bridge between the intelligent analysis layer and the maintenance management layer.
- Maintenance actions ensure complete traceability of incidents from detection to resolution.
- Both Alert and Maintenance Action maintain explicit status attributes to support lifecycle monitoring and auditing capabilities. [53]

8. Conclusion

Chapter Four demonstrated the effectiveness of the proposed framework through the design and implementation of three interconnected artificial intelligence systems.

The Autoencoder model for turbine anomaly detection achieved an accuracy of 94.8% with a ROC-AUC of 0.972, outperforming Isolation Forest and One-Class SVM. For bearing fault detection and severity classification, the dual CatBoost model achieved exceptional performance with an accuracy exceeding 99.96% on the HUSTbearing dataset. In energy forecasting, the ensemble model (XGBoost + LightGBM + CatBoost) achieved a MAPE of 0.2046% and an R^2 score of 0.9994, while ensuring zero under-supply in operational optimization results.

The chapter also presented a complete UML-based system modeling and a well-structured platform architecture, establishing a strong foundation for efficient predictive maintenance and energy management in industrial environments.

Chapter 5: FactoryNova Platform Implementation and Deployment

1. Introduction

This chapter presents the practical implementation of the FactoryNova system through its graphical user interface and workflow.

It highlights core functions including authentication, worker management, alert monitoring, maintenance tracking, and fault detection.

It also covers electricity demand forecasting and energy optimization, organized by user roles: Administrator, Maintenance Staff, and Manager.

Finally, it shows real-time outputs from AI models and retraining features, demonstrating system efficiency in predictive maintenance and energy management.

2. Application Description

FactoryNova is a web-based AI platform for industrial monitoring, predictive maintenance, and energy management. It uses machine learning and deep learning for fault detection, forecasting, and energy optimization. The system integrates multiple AI models within a role-based environment for different users. It provides dashboards and analytics to support efficient and informed decision-making.

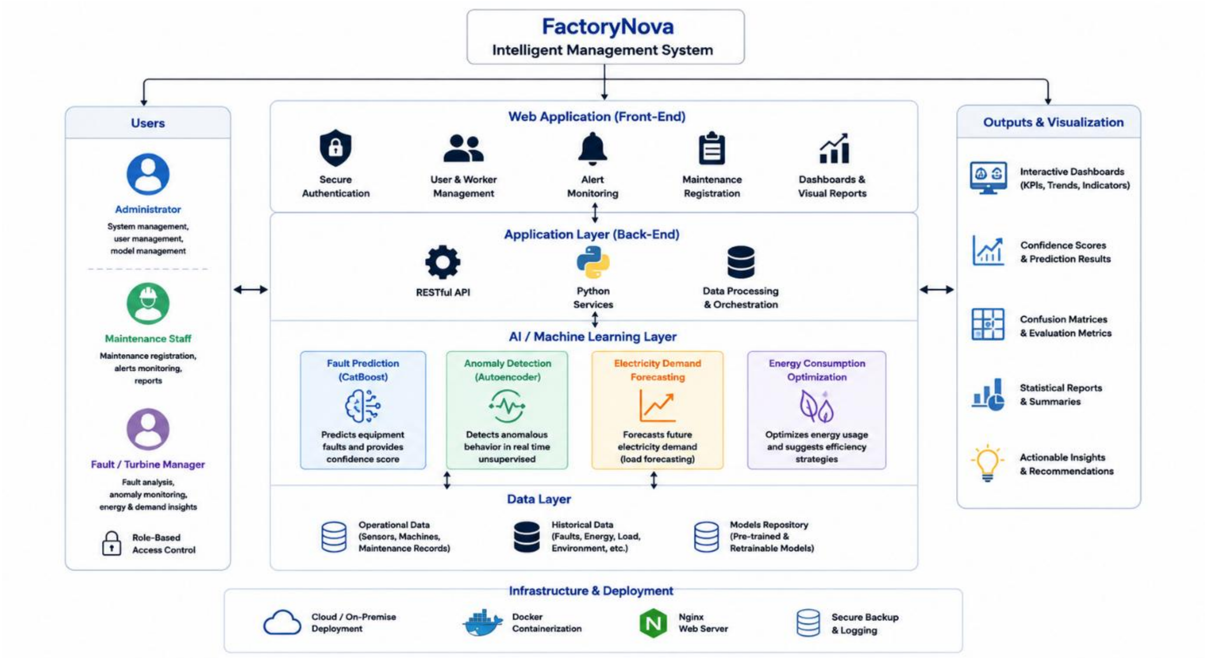


Figure 5.1: FactoryNova System Architecture Overview

3. Application Objective

The application provides a unified AI platform for industrial decision-making and automation. It manages the full AI lifecycle to improve operational efficiency. It offers a centralized environment for model development and evaluation. It is designed to be accessible for all user levels, from basic to advanced.

4. Application Features

The application provides the following key features:

- Role-based authentication with secure password recovery and user access control.
- Administration tools for managing worker accounts and system users.
- Real-time monitoring dashboards displaying alerts, equipment status, and operational indicators.
- Maintenance management interfaces for recording interventions, repair activities, and maintenance history.
- Fault Prediction module for identifying equipment faults and their severity levels using machine learning models.
- Anomaly Detection module for detecting abnormal operational behavior from industrial sensor data.
- Electricity Demand Forecasting module for predicting future energy demand based on historical consumption patterns.
- Energy Consumption Optimization module for supporting efficient energy utilization and reducing operational costs.
- Model training and retraining interfaces allowing AI models to be updated with new datasets.
- Performance evaluation and visualization tools, including prediction results, confidence scores, statistical analyses, and graphical reports.
- Automated report generation and data export functionalities for decision support and documentation purposes.

5. Application Interface

This section presents the detailed graphical user interfaces of the FactoryNova system. Each interface is described according to its purpose, the user role it serves, and the outputs it produces.

5.1. Common Interfaces

The system provides several interfaces accessible to all users regardless of their role. These interfaces ensure secure access, account management, and efficient navigation throughout the platform.

5.1.1. Login Interface

This interface allows users to securely authenticate and access the system according to their assigned permissions. The login page displays the FactoryNova branding along with a security notice indicating that unauthorized access attempts are monitored and logged. Users enter their username and password to sign in.

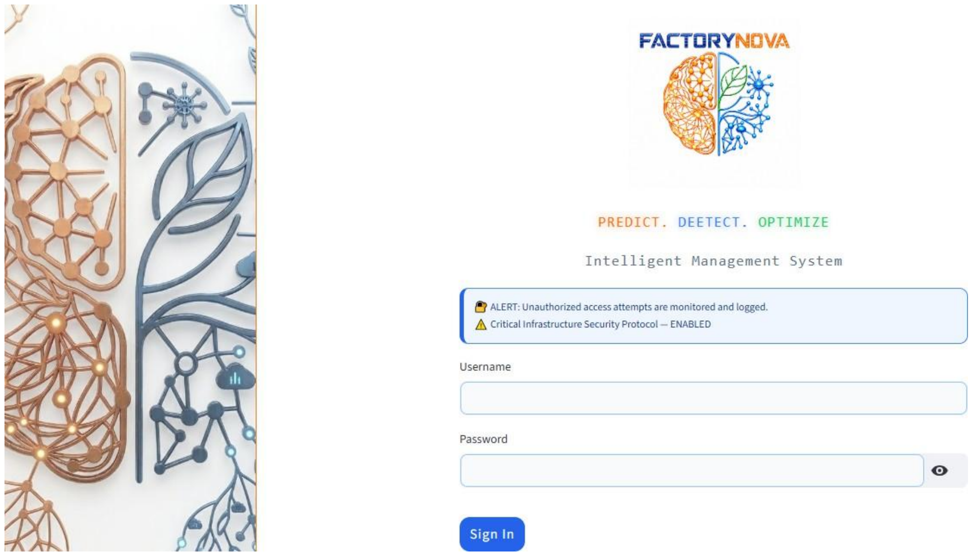


Figure 5.2: Login Interface

5.1.2. Change Password via OTP Interface

This interface enables all users to reset their password through their registered email address. The user enters their username, requests an OTP code sent to their email, verifies the code, and sets a new password. This mechanism ensures secure credential recovery without requiring administrator intervention.

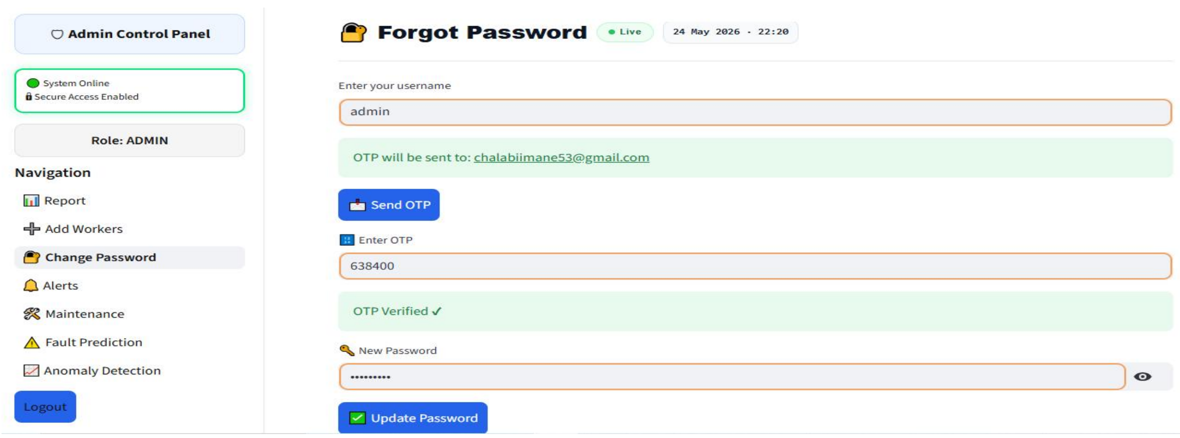
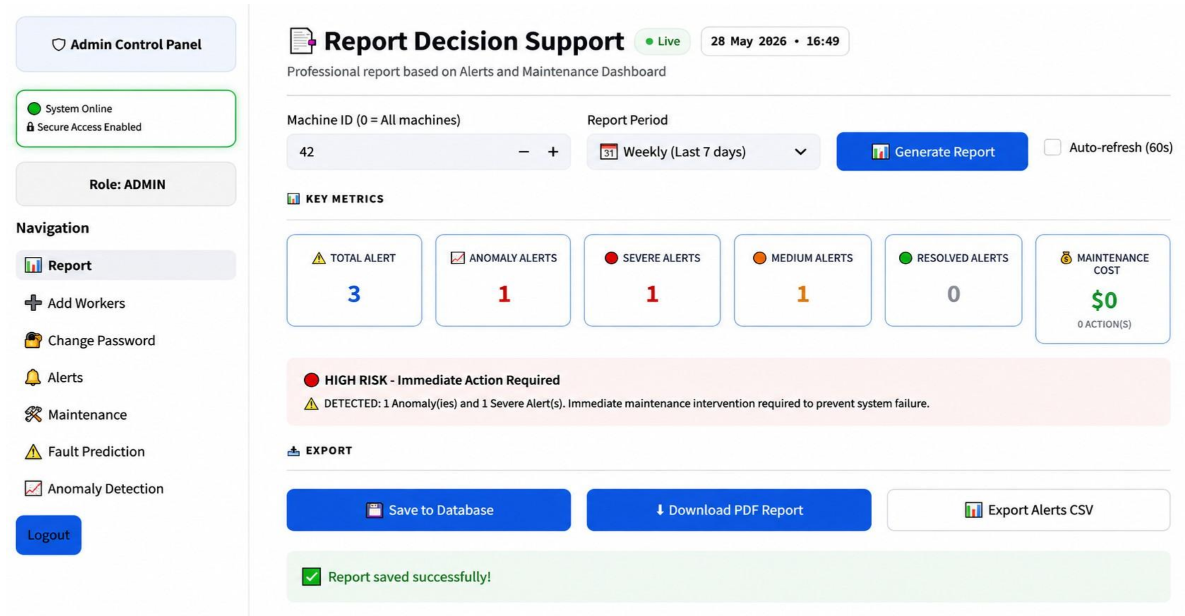


Figure 3: Change Password via OTP Interface

5.1.3. Report Interface

The Report interface converts system alerts and maintenance records into clear, structured decision-support reports. Managers can select a machine ID and a report period (e.g., weekly), then generate a report that displays key metrics total alerts, anomaly alerts, severe alerts, medium alerts, resolved alerts, and total maintenance cost. The system evaluates overall risk level and recommends immediate action when necessary. Reports can be saved to



the database, downloaded as PDF, or exported as a CSV file.

Figure 5.4: Report Interface

5.2. Admin Interfaces

The Administrator holds full system privileges, including the ability to manage user accounts and access all monitoring and reporting modules.

5.2.1. Manage Workers Interface

This interface allows the Administrator to add new worker accounts by specifying their username, password, role (maintenance, faults_manager, or Anomaly_manager….), and registered email address. It also displays the complete list of existing workers with their roles, creation dates, failed login attempts, last login IP, and email. Workers can be deleted from the system using the Delete Worker section. A confirmation message is displayed upon successful addition.

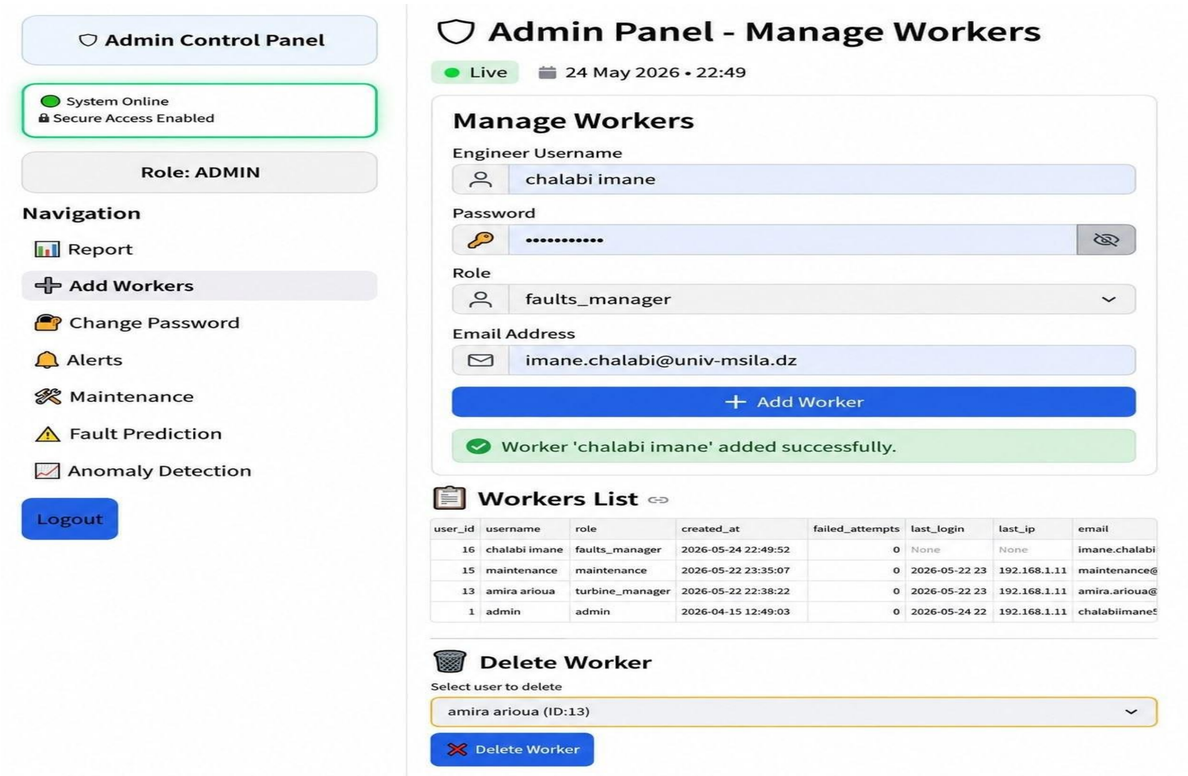


Figure 5.5: Manage Workers Interface

5.2.2. Maintenance Staff Interfaces

Maintenance staff can access the Alerts Interface to monitor active fault alerts and perform maintenance actions to resolve them.

5.2.3. Alerts Dashboard Interface

The Alerts Dashboard provides maintenance staff with a comprehensive overview of the system's alert status. It displays the total number of alerts, active alerts, resolved alerts, severe alerts, medium alerts, and anomaly alerts as summary cards. A table lists current active alerts with their machine ID, alert type, fault type, severity level, anomaly flag, and last detection timestamp. The dashboard also includes a Machine Health Score bar chart showing health percentages per machine, a Severity Distribution bar chart, and an Alerts Status Overview pie chart distinguishing active from resolved alerts.

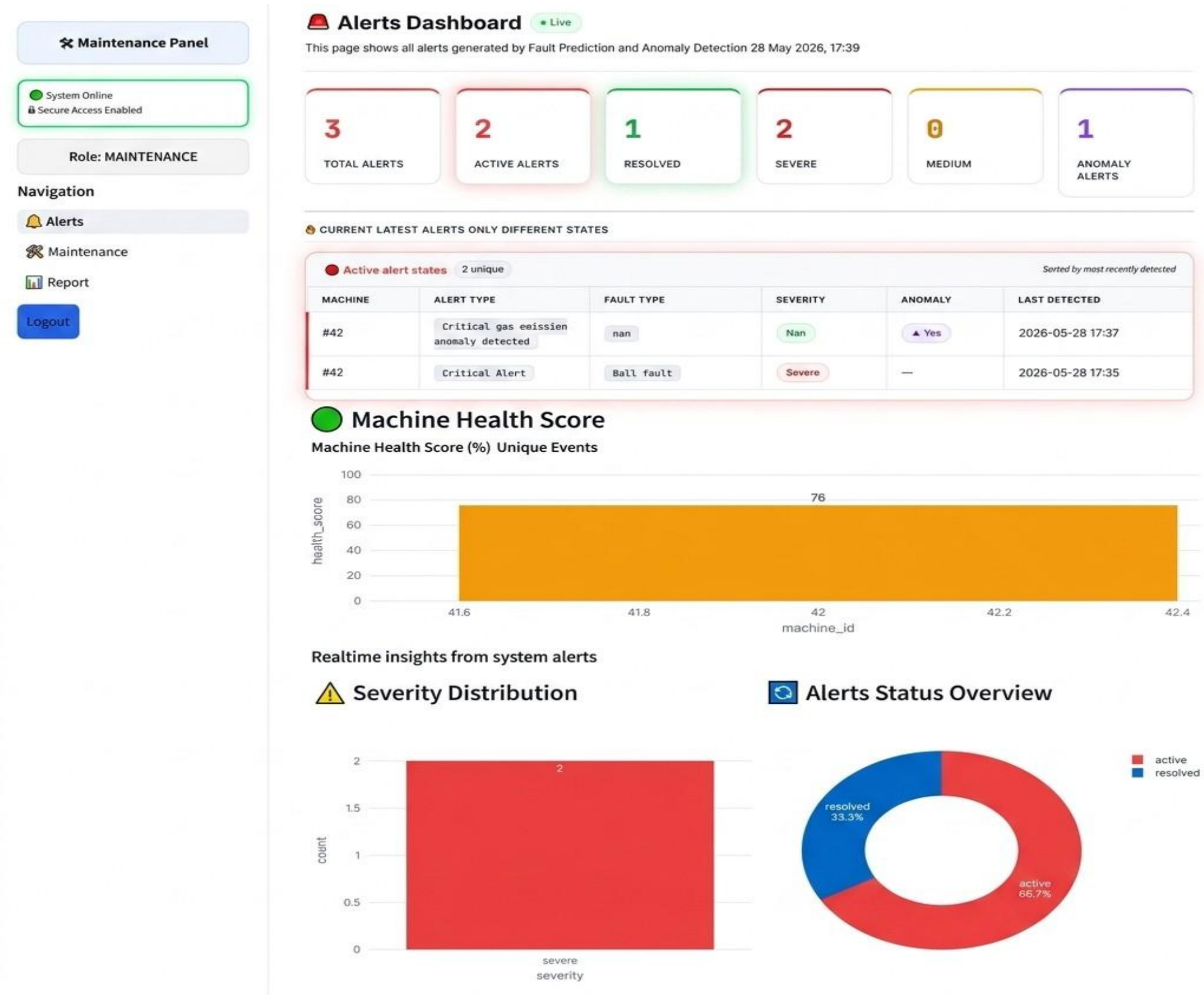


Figure 5.6: Alerts Dashboard Interface

5.3. Maintenance Interface

This interface allows maintenance staff to act on active alerts by selecting a specific alert from a dropdown menu. Once an alert is selected, its details machine ID, alert type, fault type, severity, and anomaly flag are displayed as a summary banner. The staff member then fills in the maintenance action form specifying:

- Action type (e.g., Repair, Replacement, Inspection).
- Maintenance status (e.g., completed, in progress).
- Technician name responsible for the intervention.
- Replaced part name and associated cost in dollars.
- Description or notes providing contextual details about the intervention.

After submission, the maintenance record is saved and the alert is marked as resolved. A confirmation message is shown, and the record is automatically added to the Maintenance History.

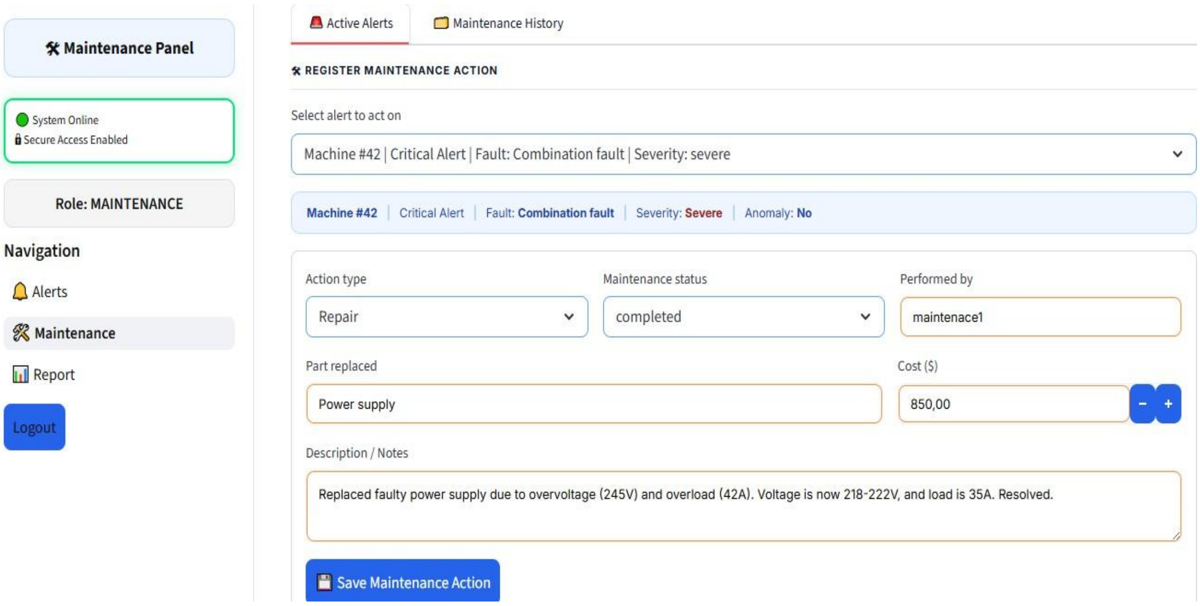


Figure 5.7: Maintenance Interface

5.3.1. Maintenance History Interface

The Maintenance History tab summarizes all completed maintenance actions, displaying the total cost of all interventions and the total number of recorded actions. Each row shows the action ID, machine ID, alert ID, timestamp, action type, and a brief description. This historical log enables managers to track maintenance expenses and review past interventions for audit or reporting purposes.

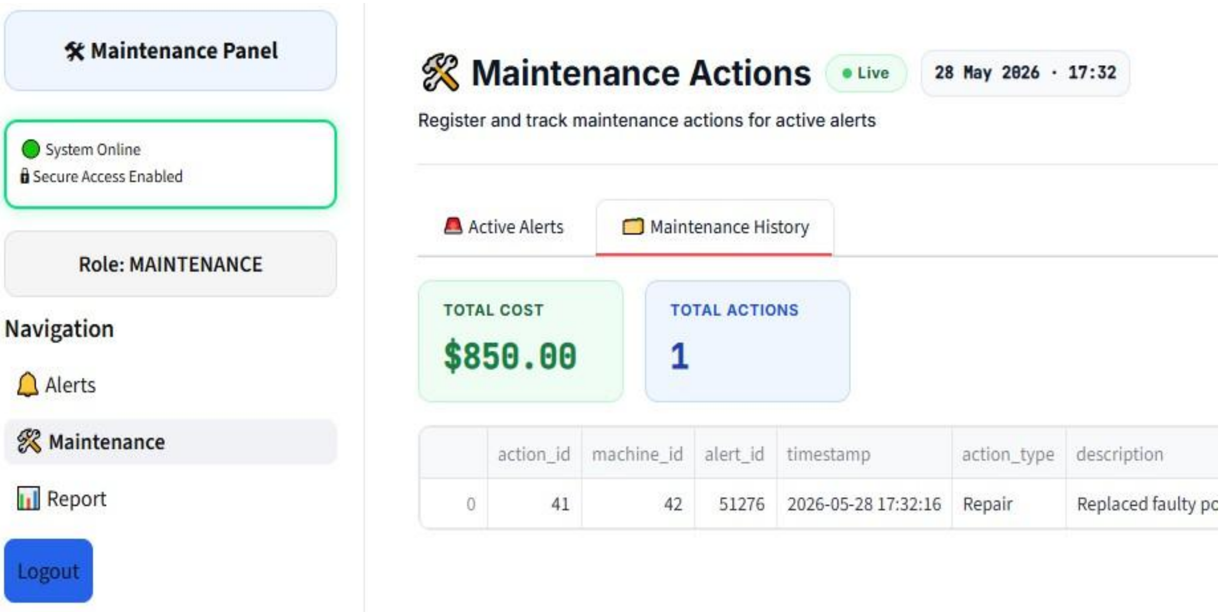


Figure 5.8: Maintenance History Interface

5.4. Fault Manager Interfaces

The Fault Manager is responsible for predicting potential fault types and severity levels from raw vibration data, monitoring generated alerts, and retraining the prediction model when new data becomes available.

5.4.1. Fault Prediction Interface

This interface allows the Fault Manager to upload one or more raw vibration data files in CSV or XLSX format. After clicking Predict All Files, the system processes each file using the trained CatBoost model and displays results in a table showing, for each signal window: the source file name, the window index, the predicted fault type, the type confidence score, the predicted fault severity, and the severity confidence score. Below the prediction table, two bar charts are displayed:

- Fault Type Confidence: average confidence per predicted fault type (e.g., C, H).
- Fault Severity Confidence: average confidence per severity level (Normal, Severe).

The user can download the complete prediction results as a CSV file for further analysis or reporting.

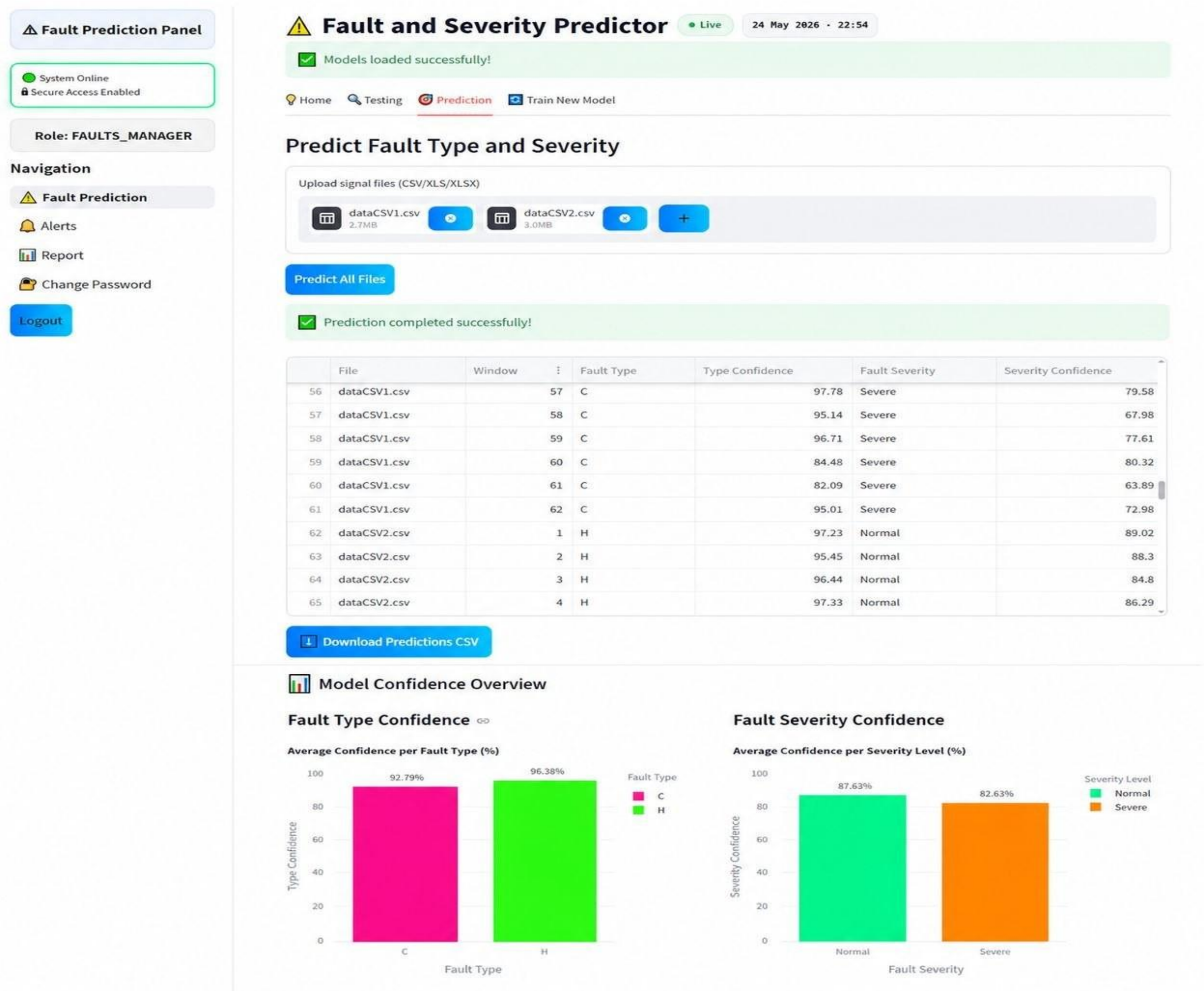


Figure 5.9: Fault Prediction Interface

5.4.2. Training New Fault Prediction Model Interface

This interface enables the Fault Manager to retrain the fault prediction model using a new local dataset. The user specifies the dataset folder path on the server, adjusts the number of CatBoost iterations (default: 500), sets the test size percentage (default: 20%), and optionally enables SMOTE oversampling to handle class imbalance. After clicking Train New Models Now, the system trains updated classifiers for both fault type and severity level, replacing the existing models. This workflow ensures that the system can adapt to new machinery data or evolving fault patterns without requiring developer intervention.

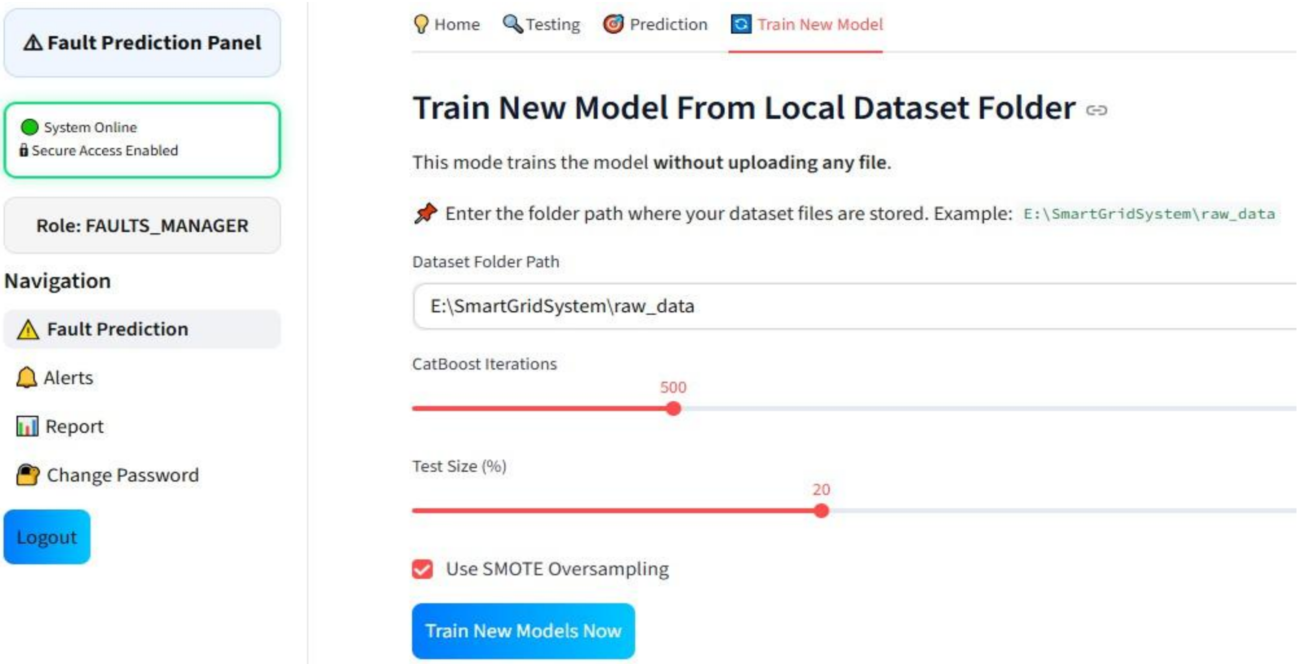


Figure 5.10: Training New Fault Prediction Model Interface

5.5. Anomaly Manager Interfaces

The Turbine Manager oversees turbine operations and evaluates performance indicators by running the anomaly detection module on sensor data.

5.5.1. Anomaly Detection Interface

The Anomaly Detection Prediction interface analyzes new turbine sensor datasets using the trained autoencoder model. The user uploads one or more CSV or Excel files and clicks Run Prediction. The system computes the reconstruction error for each sample and classifies it as normal or anomaly by comparing the error against a predefined threshold. For each uploaded file, the interface displays:

- Total number of samples, normal count, and anomaly count as summary cards.
- A reconstruction error time-series plot with the anomaly threshold marked as a dashed line and anomaly points highlighted.

- A scatter plot of reconstruction errors over sample indices, with anomaly points color-coded.
- A Reconstruction Error Distribution histogram separating normal and anomaly samples.
- Each file's results can be exported individually as a CSV, and a combined export of all files is also available.

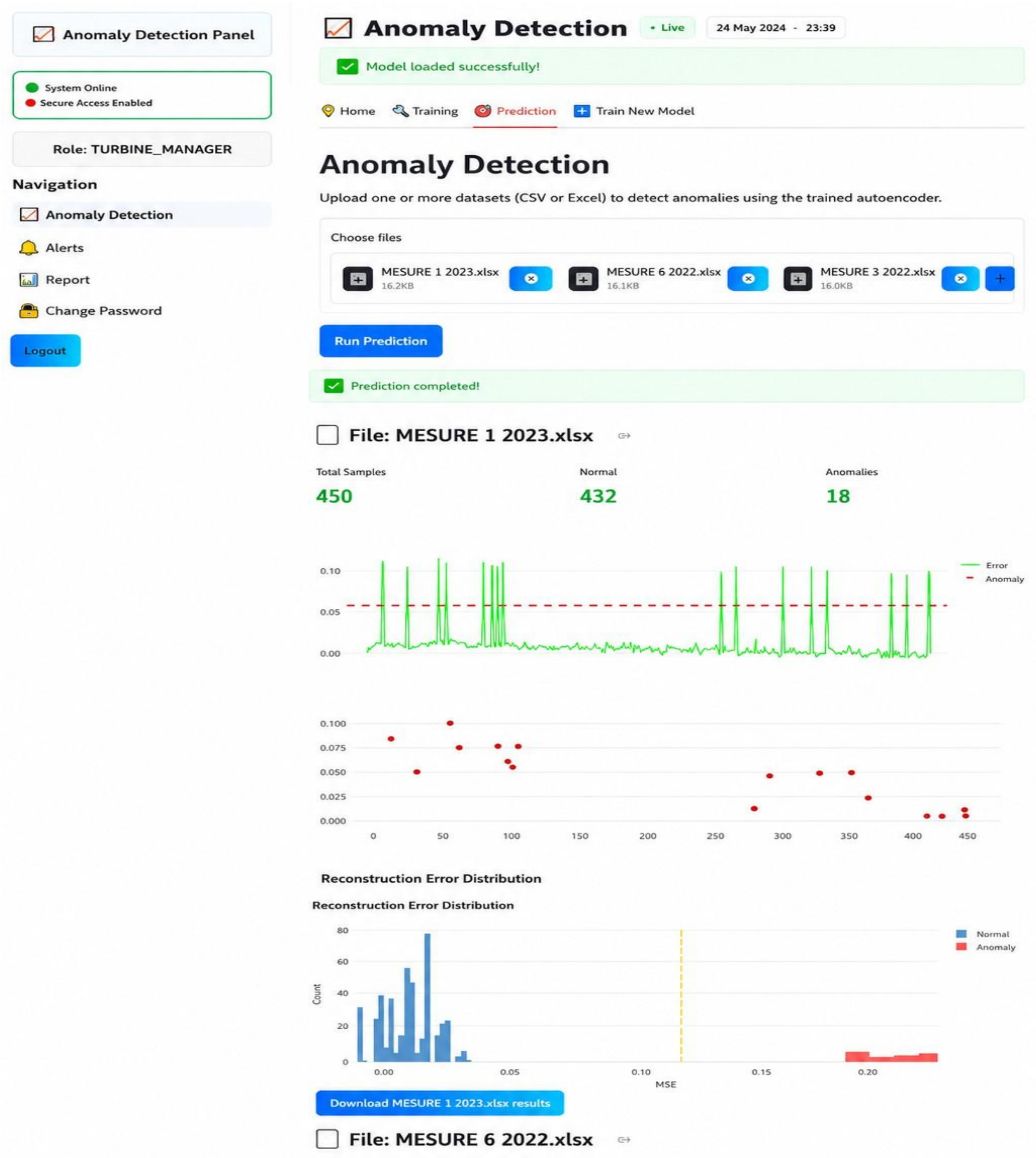


Figure 5.11: Anomaly Detection Interface Part 1 (File: MESURE 1 2023.xlsx)

Figure 12: Anomaly Detection Interface Part 2 (Files: MESURE 6 2022.xlsx and MESURE 5 2022.xlsx)

5.5.2. Testing the Anomaly Detection Model

The Testing interface evaluates the performance of the trained autoencoder model using labeled datasets, enabling a thorough comparison between model predictions and true class labels. The user uploads a labeled CSV or Excel test file and clicks Run Testing. The system then computes and displays the following performance metrics:

- **Accuracy**: overall proportion of correctly classified samples.
- **Precision**: proportion of detected anomalies that are true anomalies.
- **Recall**: proportion of actual anomalies successfully detected by the model.
- **F1-Score**: harmonic mean of precision and recall.
- **ROC-AUC**: ability of the model to discriminate between normal and anomaly classes.
- **Average Precision**: area under the Precision-Recall curve.

The interface also displays a Confusion Matrix showing the counts of True Positives, True Negatives, False Positives, and False Negatives, a ROC Curve, a Precision, Recall Curve, a Reconstruction Error by True Class histogram; and a full Classification Report table with per class precision, recall, F1-score, and support values.

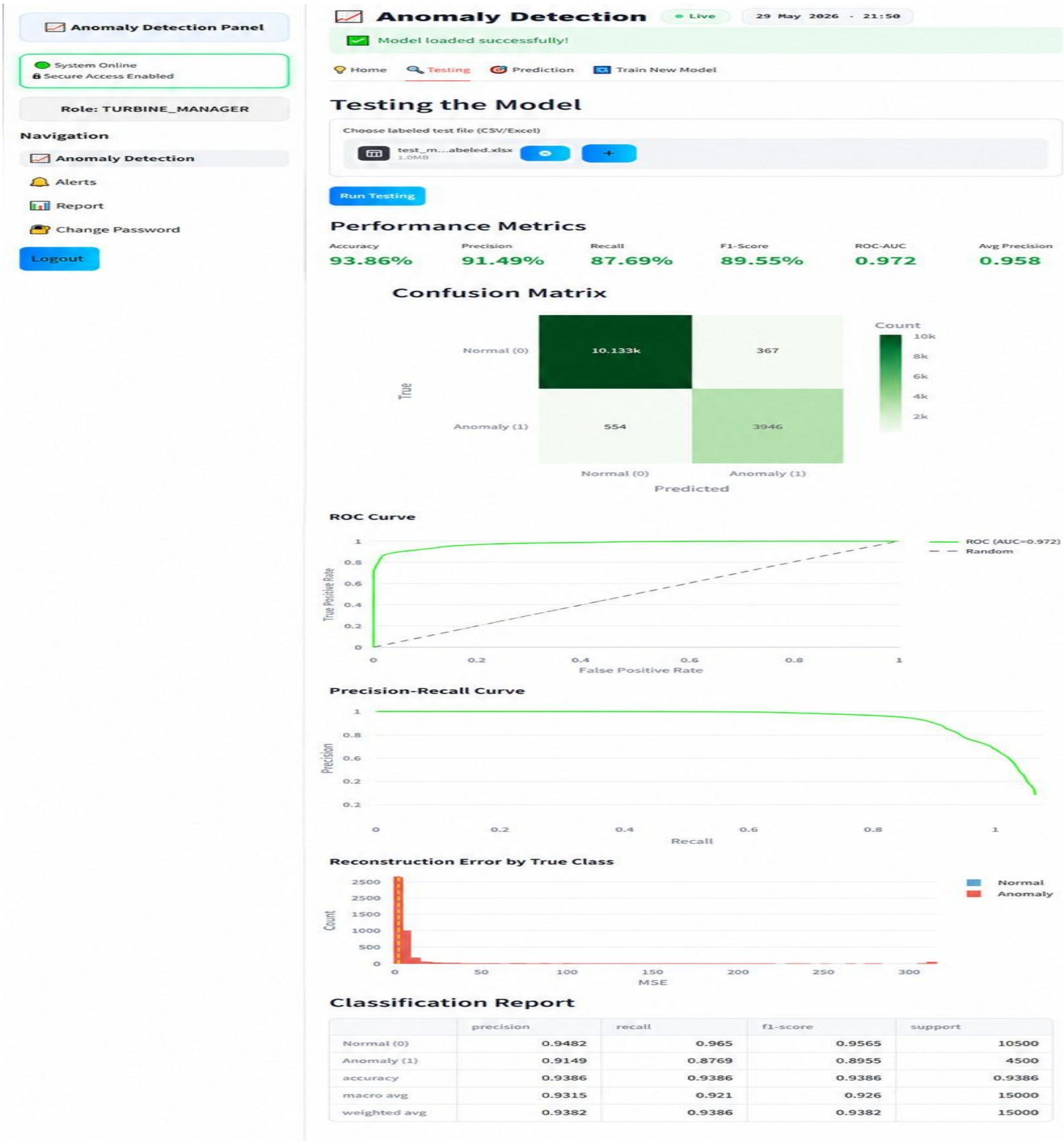


Figure 5.13: Testing Anomaly Detection Model Interface

5.5.3. Training New Anomaly Detection Model Interface

The Train New Model interface allows the Turbine Manager to retrain the autoencoder using a new set of normal operational data. The user uploads a training dataset (CSV or Excel containing only normal samples), previews the data in a tabular format, configures the number of training epochs and batch size, and clicks Run Training. The system retrains the autoencoder, updates the anomaly detection threshold based on the new reconstruction error distribution, and saves the updated model for future predictions. This ensures the system remains accurate as turbine behavior evolves over time.

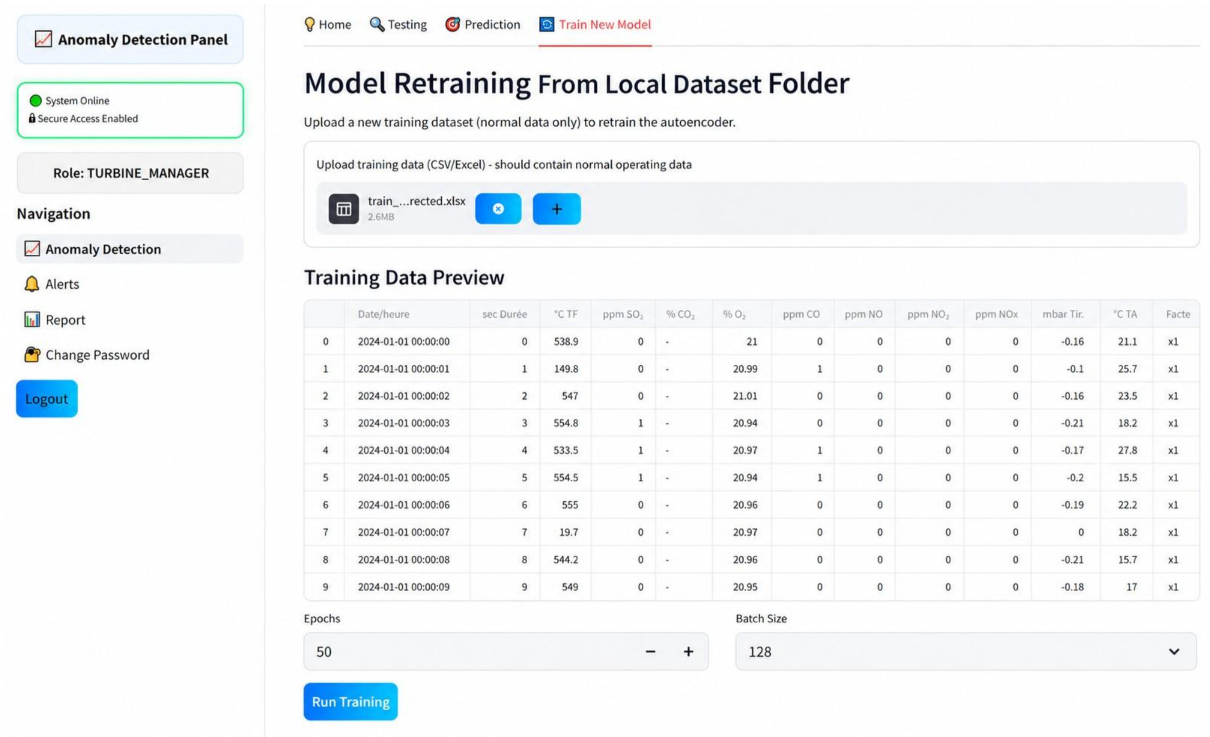


Figure 5.14: Training New Anomaly Detection Model Interface

6. Conclusion

In conclusion, this chapter presented the implementation and user interface of the FactoryNova Intelligent Management System. Through its role-based architecture, the platform supports predictive maintenance and intelligent energy management by integrating real-time monitoring, maintenance tracking, fault prediction, anomaly detection, electricity demand forecasting, and energy consumption optimization. The interfaces were designed to be intuitive and accessible to users with different technical backgrounds, while the underlying AI models provide accurate and reliable predictions.

Furthermore, the system supports model retraining, ensuring adaptability to changing industrial and energy conditions. The obtained results demonstrate the effectiveness of the proposed approach and provide a solid foundation for future enhancements, including real-time sensor integration, mobile notifications, and advanced multimachine and energy monitoring capabilities.

Chapter 6: Business model and detailed financial study

1. Introduction

This chapter presents the business model and financial feasibility of FactoryNova, an intelligent platform designed for predictive maintenance, fault diagnosis, anomaly detection, energy forecasting, and energy optimization in industrial environments.

The financial projections are developed over a five-year horizon based on realistic assumptions regarding market growth, customer acquisition, operational costs, and startup financing. The objective is to demonstrate the economic viability and scalability potential of the project.

2. Startup Presentation

2.1. Commercial Name

FactoryNova

Figure 6 .1: FactoryNova: Commercial Name and Logo

2.2. Mission

FactoryNova aims to help industrial companies reduce equipment failures, optimize maintenance operations, and improve energy efficiency through Artificial Intelligence and Machine Learning technologies.

2.3. Vision

To become the leading Algerian AI-powered industrial intelligence platform and expand across the North African energy and manufacturing sectors within five years.

2.4. Core Values

- Innovation
- Reliability

- Sustainability
- Operational Excellence
- Data-Driven Decision Making

2.5. Current Stage

MVP (Minimum Viable Product)

The platform has already been developed and tested using industrial datasets and includes functional modules for predictive maintenance, anomaly detection, and energy forecasting.

3. Business Model Canvas

3.1. Customer Segments

- **Public Industrial Sector:** Power generation and distribution companies.
  - Petrochemical and natural gas companies
  - Water utilities and treatment facilities.
  - Infrastructure and facility management organizations.
- **Private Industrial Sector:**
  - Cement factories.
  - Food and beverage processing plants.
  - Textile and garment factories.
  - Chemical industries.
  - Automotive and component manufacturing.
- **Academic and Research Sector:**
  - Universities.
  - Research centers.
  - AI and energy research laboratories.

3.2. Value Proposition

- **Predictive Maintenance:** Early fault detection to reduce unplanned downtime and transition from traditional maintenance to predictive industrial environments.

- **Advanced Predictive Techniques:** Comprehensive data collection, data quality management, and technical load analysis to support accurate, data-driven decisions.
- **Real-time Unified Platform:** An integrated digital platform that consolidates data, connects operations, and supports real-time decision-making instead of relying on fragmented data.
- **Accurate and Real-time Reporting:** Automated performance reporting with tailored optimization recommendations to improve responsiveness.
- **Global and Local Standards Compliance:** Platform alignment with international and local industrial standards, with full support for the Arabic language.

3.3. Channels

- **Direct B2B Transactions:** Direct corporate subscriptions and enterprise offers.
- **Direct Tech Channels:** Online platforms, digital portals, and direct technical support interfaces.
- **Institutional Partnerships:** University-industry collaborations and government research grants.

3.4. Customer Relationships

- **Training and Qualification:** Interactive user training sessions and workshops to enable technical staff to maximize platform utility efficiently.
- **Continuous Support:** Ongoing technical support, regular system maintenance, and periodic platform updates.
- **Collaborative Partnerships:** Joint research and development initiatives with universities to deploy the solution.
- **Premium After-Sales Services:** Providing continuous value through periodic reports, advanced analytics, strategic model development, and continuous smart-model updates.

3.5. Revenue Streams

- **Platform Licensing (Per production unit/capacity):** Annual licensing fees for various industrial sectors.
- **Customized Smart Models:** Developing and selling tailored predictive models for specific industries.
- **Government/Institutional Funding:** Research grants and project funding from the Ministry of Higher Education and Scientific Research.

- **Consulting Contracts:** Technical consulting services for factories and industrial projects on AI and energy integration.
- **Academic Licensing:** Special licensing for academic centers and universities to test and expand the predictive system.
- **Direct B2B Revenue Flows:**
  - Fixed annual subscriptions and tailored enterprise offers.
  - Implementation, setup, and post-execution service contracts.
  - Monthly usage licenses.
  - Online platform access and premium tech support fees.
  - Specialized consulting for performance optimization.
  - Revenue from standard national and international regulations compliance partnerships.

3.6. Key Resources

- **Human Resources:**
  - Engineering Team (Specialized in software, AI, industrial engineering, data analytics).
  - Electrical engineering consultants and technical experts.
- **Technical/Physical Resources:**
  - Databases (Kaggle).
  - Data Access (Sonelgaz datasets).
  - Digital simulation software.
  - Cloud hosting infrastructure.
  - Machine learning and data analytics tools.
- **Infrastructure Resources:** Main offices and industrial hardware infrastructure base.
- **Intellectual Property:** Platform proprietary rights and software copyrights.

3.7. Key Activities

- **Data Collection:** Continuous gathering of energy consumption and industrial operational data.
- **Data Analysis:** Processing industrial data using artificial intelligence.

- **Predictive Model Development:** Building models for fault prediction and energy/production optimization.
- **Smart Maintenance Implementation:** Deploying and executing predictive maintenance frameworks.
- **Reporting:** Issuing performance reports and optimization recommendations.
- **Support:** Continuous user training and technical support operations.

3.8. Key Partnerships

- **Data Companies:** Sonelgaz.
- **Industrial Big Data Analytics Companies.**
- **Technology Partners:** Cloud computing companies and research centers.
- **Solution Developers & Business Partners:** Algeria Venture.
- **Academic Partners:** University of M'Sila.

3.9. Cost Structure

- **Fixed Costs:**
  - **Salaries and Benefits** (Development team and experts): 2,400,000 DA
  - **Cloud Hosting & Infrastructure Costs:** 500,000 DA
  - **Training and Technical Support** for Clients: 250,000 DA
  - **Server and Data Storage Costs:** 400,000 DA
  - **Marketing and Business Development Budget:** 300,000 DA
  - **Platform Security, Licenses and Maintenance:** 200,000 DA
- **Variable Costs:**
  - **Digital Marketing and Campaigns:** 500,000 DA
  - **Research and Development** (New features): 600,000 DA
  - **Scaling Infrastructure** (Based on user base growth): 350,000 DA

4. Financial Projections

4.1. General Assumptions

The financial model is based on the following assumptions:

- Average annual market growth: 20%
- Inflation rate: 5%
- Customer payment period: 30 days
- Subscription-based SaaS business model
- Progressive customer acquisition through partnerships and direct sales

4.2. Initial Investment Plan

Category	Amount (DZD)
Platform & AI Development	2,121,227
Technical Equipment	433,001
Total Initial Investment	**2,554,228**

Table 6.1 : Initial Investment Plan

4.3. Revenue Projections

Revenue Stream	Year 1	Year 2	Year 3	Year 4	Year 5
Annual Licenses	7,500,000	15,000,000	22,500,000	30,000,000	45,000,000
Maintenance Contracts	1,050,000	2,100,000	6,800,000	6,000,000	9,000,000
Custom Development	500,000	500,000	1,000,000	1,000,000	1,500,000
Government Grants	200,000	200,000	0	0	0
Consulting Services	0	100,000	400,000	900,000	1,600,000
Academic Licensing	50,000	200,000	450,000	800,000	1,250,000
Total Revenue	**9,300,000**	**18,100,000**	**31,150,000**	**38,700,000**	**58,350,000**

Table 6.2: Projected Revenue by Source (DZD)

4.4. Projected Income Statement

Year	Revenue (DZD)	Total Costs (DZD)
Year 1	9,300,000	34,652,000
Year 2	18,100,000	34,942,000
Year 3	31,150,000	35,261,000
Year 4	38,700,000	35,611,900
Year 5	58,350,000	35,997,890

Table 6.3 : Summary Income Statement

5. Operating Expenses Breakdown

5.1. Payroll Expenses

Position	Monthly Salary (DZD)
AI Developer	150,000
Full Stack Developer	150,000
Industrial Expert	180,000
Technical Manager	155,000
Technical Support Officer	100,000
Total Annual Payroll	**31,752,000 DZD**

Table 6.4: Summary Annual Payroll

5.2. Other Operating Expenses

Expense Category	Year 1	Year 2	Year 3	Year 4	Year 5
Cloud Hosting & Database Infrastructure	400,000	440,000	484,000	532,400	585,640
Platform Maintenance & Cybersecurity	300,000	330,000	363,000	399,300	439,230
Software Licenses & Development Tools	200,000	220,000	242,000	266,200	292,820
Digital Marketing & Sales	300,000	330,000	363,000	399,300	439,230
Research & Development	400,000	440,000	484,000	532,400	585,640
Infrastructure Expansion	250,000	275,000	302,500	332,750	366,025
Office Rent / Incubator Space	420,000	462,000	508,200	559,020	614,922
Utilities (Electricity & Services)	60,000	66,000	72,600	79,860	87,846
Internet & Telecommunications	24,000	26,400	29,040	31,944	35,138
Training & Professional Development	250,000	275,000	302,500	332,750	366,025
Professional Fees (Accounting, Legal, Notary)	100,000	110,000	121,000	133,100	146,410
Travel & Field Operations	96,000	105,600	116,160	127,776	140,554
Miscellaneous Expenses	100,000	110,000	121,000	133,100	146,410
Total Operating Expenses	2,900,000	3,190,000	3,509,000	3,859,900	4,245,890

Table 6.5 : Projected Operating Expenses (DZD)

The operating expenses are expected to increase gradually over the five-year planning horizon, reflecting the startup's growth strategy and expansion of activities. The largest cost categories are cloud infrastructure, research and development, office facilities, and cybersecurity maintenance. An average annual increase of approximately 10% has been applied to most expense categories to account for business expansion, inflation, and increased operational requirements. Despite this growth, the projected increase in revenue significantly exceeds the growth in operating expenses, supporting the long-term financial sustainability of FactoryNova.

5.3. Cash-Flow Projection

Year	Cash Balance (DZD)
Year 1	2,000,002
Year 2	2,270,002
Year 3	3,676,002
Year 4	5,923,702
Year 5	9,220,632

Table 6.6: Projected Cash Position

The projected cash position remains positive throughout the planning horizon, indicating adequate liquidity and financial sustainability.

5.4. Financing Plan and Capital Structure

Financing Source	Amount (DZD)
Founders' Contribution	1,054,229
Algeria Venture Funding	3,000,000
Government Grants	500,000
Total Funding	**4,554,229**

Table 6.7 : Financing Sources

5.5. Financial Risk Analysis

Risk	Probability	Impact	Mitigation Strategy
Lower-than-expected sales	Medium	High	Strategic partnerships and marketing
Cost overruns	Medium	Medium	Budget control and monitoring
Cash-flow shortages	Low	High	Reserve funding and grants
Regulatory changes	Low	Medium	Continuous compliance monitoring
Dependence on key customers	Medium	High	Customer diversification

Table 6.8: Financial Risk Matrix and Mitigation Strategies

6. Development Roadmap and Key Milestones

Period	Milestone
M0–M3	Finalization of MVP
M3–M6	Pilot testing with industrial partners
M6–M12	Commercial launch and first contracts
M12–M18	Product enhancement and customer acquisition
M18–M24	Expansion and scaling phase

Table 6.9: Strategic Development Roadmap and Key Milestones for FactoryNova

7. Conclusion

FactoryNova provides an innovative and scalable AI-powered solution for predictive maintenance and energy optimization in industrial environments. Financial projections show strong growth, with revenues increasing from 9.3 million DZD in Year 1 to 58.35 million DZD in Year 5. The startup benefits from diversified revenue streams, strategic partnerships, and positive cash-flow forecasts, supporting its financial viability. The main priority is the commercialization of the MVP through pilot projects and industrial partnerships.

General Conclusion

This work addressed the challenges of reactive maintenance practices and the underutilization of industrial data at the M'sila Gas Turbine Production Unit of SONALGAZ. To overcome these limitations, an intelligent Platform named FactoryNova was designed and implemented to support predictive maintenance and energy optimization through the integration of advanced artificial intelligence techniques.

The proposed solution combines three complementary modules: an Autoencoder-based anomaly detection system for monitoring gas turbine operation, a CatBoost-based fault diagnosis and severity classification system for rotating machinery, and a stacked ensemble model for electricity demand forecasting and energy optimization. These modules were integrated into a unified web-based platform providing monitoring, predictive analytics, and decision-support functionalities.

The experimental results demonstrated the effectiveness of the proposed Platform. The anomaly detection model achieved an accuracy of 94.8% with a ROC-AUC of 0.972, while the CatBoost classifiers achieved accuracies exceeding 99.9% for fault type and severity prediction. In addition, the electricity demand forecasting model achieved a MAPE of 0.2046% and an R^2 of 0.9994, confirming its high predictive performance. These results highlight the potential of machine learning and deep learning techniques for improving equipment reliability, reducing maintenance costs, and supporting efficient energy management.

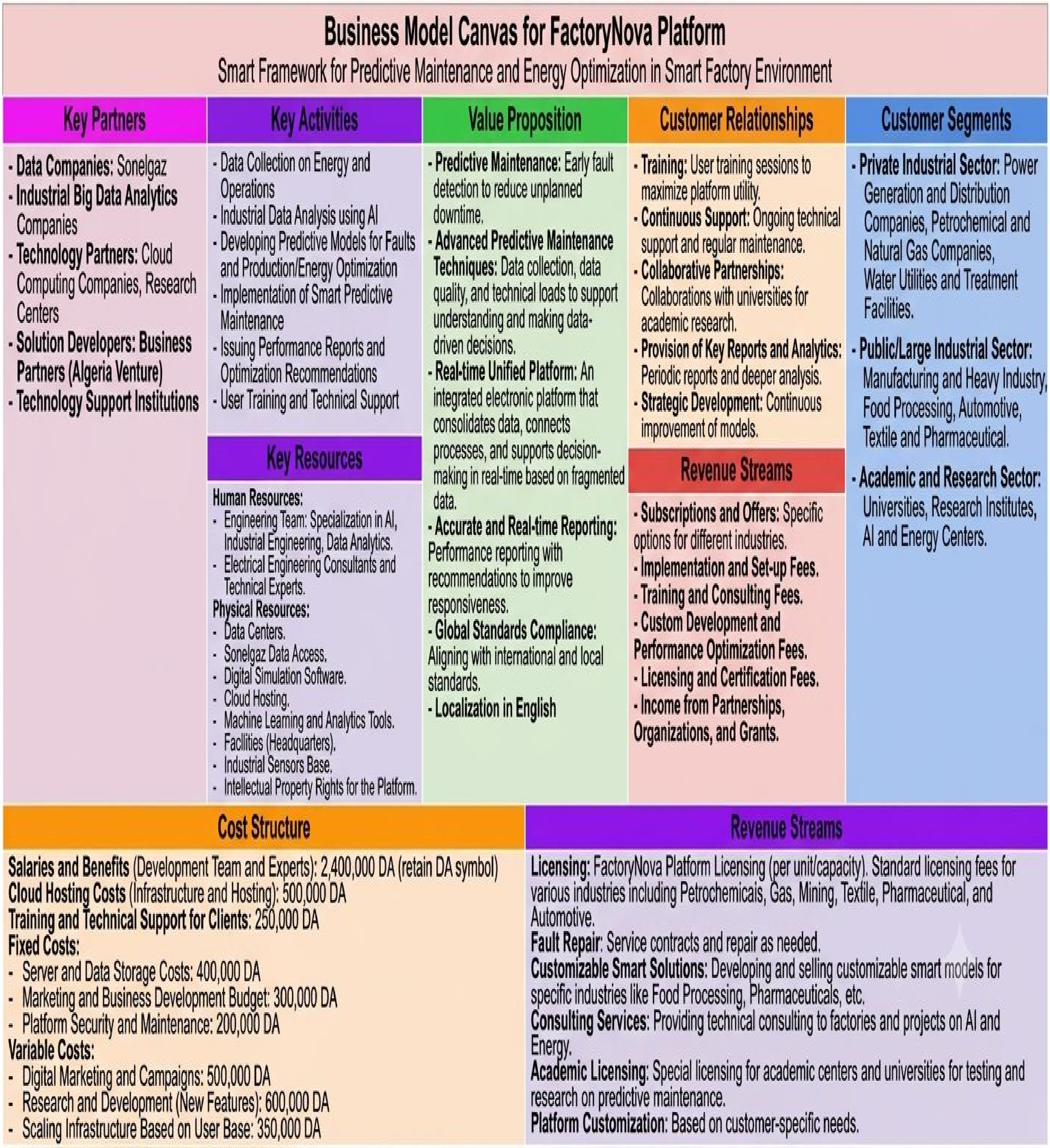
The main contribution of this work lies in the development of an integrated AI-driven Platform that combines anomaly detection, fault diagnosis, demand forecasting, and energy management within a single platform. The study also demonstrates the practical applicability of advanced AI methods in a real industrial context and proposes a physics-informed synthetic data generation strategy to address the scarcity of labeled fault data.

Despite the promising results, some limitations remain. The anomaly detection module relied on synthetic fault data, the fault classification system was validated using a public benchmark dataset, and the FactoryNova platform has not yet been deployed in a fully operational industrial environment. Future work should focus on collecting labeled industrial fault data, integrating real-time sensor streams, extending the framework to additional equipment, and evaluating the platform through a pilot deployment in collaboration with SONALGAZ.

Overall, this work demonstrates that the integration of artificial intelligence into industrial energy systems can significantly enhance predictive maintenance capabilities, improve operational efficiency, and support more intelligent and sustainable energy management practices.

List of Annexes

Annex 1: Business Model Canvas



Annex 2: Certificate of incubation / hosting of an innovative project under Ministerial Decision 1275

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي
جامعة محمد بوضياف المسيلة
حاضنة الأعمال

INCUBATEUR
UNIVERSITE DE M'SILA

جامعة محمد بوضياف - المسيلة
Université Mohamed Boudiaf - M'sila

الرقم : 23/ح أ/2026

شهادة توطين مشروع مبتكر ضمن القرار 008

يشهد السيد (ة): مدير(ة) حاضنة الأعمال لـ جامعة محمد بوضياف بالمسيلة أن المشروع المقترح تحت عنوان:

An intelligent Framework for Predictive Maintenance and Energy consumption Optimization

أن الطالب/الطلبة التالية أسماؤهم:

الاسم واللقب	الطور الدراسي	القسم	الكلية
عريوة أميرة	ماستر سنة ثانية	قسم الأعلام الآلي	كلية الرياضيات و الأعلام الآلي
إيمان شلابي	ماستر سنة ثانية	قسم الأعلام الآلي	كلية الرياضيات و الأعلام الآلي

تحت إشراف الأستاذ/الأساتذة التالية:

الاسم واللقب	الرتبة	القسم	الكلية
قادري سعيد	MCA	قسم الأعلام الآلي	كلية الرياضيات و الأعلام الآلي

تم تسجيله على مستوى حاضنة الأعمال لـ جامعة محمد بوضياف بالمسيلة خلال السنة الجامعية 2026/2025

ضمن القرار 008 (شهادة مؤسسة اقتصادية) المعدل والمتمم للقرار الوزاري 1275.

سلمت هذه الشهادة بطلب من المعني(ة) للإدلاء بها في حدود ما يسمح به القانون.

حرر في: المسيلة بتاريخ: 2026/06/10

مدير الحاضنة:

مديـرة حاضنـة الأعمـال
جـامعـة المسيلـة
د. بن التومي

References

[1] SONELGAZ, *Présentation de l'unité de production M'sila,,* M'sila: Production Branch – M'sila, presentation,2024.

[2] S. Djiabe, «Smart Home Controller Using Mobile Device,» Université de Mohamed Khider – Biskra, Biskra, 2025.

[3] F. Goudarzi et al., «A Survey on IoT-Enabled Smart Grids: Emerging, Applications, Challenges, and Outlook,» *Energies,* vol. 15, n° %119, p. 6984, 2022.

[4] C. Saleem et al. «Internet of Things-Aided Smart Grid: Technologies, Architectures, Applications, Prototypes, and Future Research Directions,» *IEEE Access,* vol. 7, p. 20, 2019.

[5] S. Rana et al. «A Conceptual Model Development for Evaluating AI-Driven Predictive Maintenance in SCADA-Integrated Smart Grid Systems,» *American Journal of Data Science and Analytic,* vol. 6, n° %110, p. 12, 2025.

[6] A. Momoh et al. Energy Processing and Smart Grid, USA: Wiley-IEEE Press: Hoboken, 2018.

[7] M. Balamurugan et al. «Role of artificial intelligence in smart grid – a mini review,» *Frontiers in Artificial Intelligence,* vol. 8, p. 19, 2025.

[8] M. Kiasari et al. «A Comprehensive Review of the Current Status of Smart Grid Technologies for Renewable Energies Integration and Future Trends: The Role of Machine Learning and Energy Storage Systems,» *Energies,* vol. 17, n° %116, 2024.

[9] G. Samatas, «Predictive Maintenance: Bridging Artificial Intelligence and IoT,» *arXiv preprint arXiv,* 2021.

[10] T. Zonta et al.«A Survey of Predictive Maintenance: Systems, Purposes and Approaches,» *IEEE Access,* vol. 8, 2020.

[11] O. Apata et al. «Artificial Intelligence for Predictive Maintenance and Performance Optimization in Renewable Energy Systems: A Comprehensive Review,» *Energies,* vol. 19, n° %12, 2026.

[12] M. Suci et al. «Artificial intelligence in renewable energy: A review of predictive maintenance and energy optimization,» *Journal of Clean Technology,* vol. 2, n° %11, p. 29, 2025.

[13] E. Omol et al.«Anomaly Detection in IoT Sensor Data Using Machine Learning Techniques for Predictive Maintenance in Smart Grids,» *International Journal of Science, Technology & Management,* vol. 5, n° %11, p. 201, 2024.

[14] L. Boylestad et al. Introductory Circuit Analysis, Pearson, 2016.

[15] H. Ritchie et al. «Energy Production and Consumption,» 2020. [En ligne]. Available: https://ourworldindata.org/energy-production-consumption. [Accès le 01 03 2026].

[16] U. Assad et al. «Smart Grid, Demand Response and Optimization: A Critical Review of Computational Methods,» *Energies,* vol. 15, n° %16, 2022.

[17] S. Aslam et al. «Optimization Algorithms in Smart Grids: A Systematic Literature Review,» *arXiv preprint arXiv,* 2023.

[18] B. moey et al. «Deep Learning for Intelligent Demand Response and Smart Grids: A Comprehensive Survey,» *arXiv preprint arXiv,* 2021.

[19] Y. Lei et al.«Machinery Health Prognostics: A Systematic Review from Data Acquisition to RUL Prediction,» *Mechanical Systems and Signal Processing,* vol. 104, 2018.

[20] T. Zonta et al. «Predictive Maintenance in the Industry 4.0: A Systematic Literature Review,» *Computers & Industrial Engineering,* vol. 150, 2020.

[21] T. Hong et al. «Probabilistic Energy Forecasting: Global Energy Forecasting Competition 2014 and Beyond,» *International Journal of Forecasting,* vol. 32, n° %13, 2016.

[22] P. Girdhar et al. Practical Machinery Vibration Analysis and Predictive Maintenance, USA: Elsevier:Burlington, MA, 2004.

[23] B. Randall et al. Vibration-Based Condition Monitoring: Industrial, Aerospace and Automotive Applications, U.K.: John Wiley & Sons: Chichester, 2011.

[24] R. Yan et al. «Wavelets for Fault Diagnosis of Rotary Machines: A Review with Applications,» *Signal Processing,* vol. 96, pp. 1-11, 2014.

[25] K. Jardine et al. «A Review on Machinery Diagnostics and Prognostics Implementing Condition-Based Maintenance,» *Mechanical Systems and Signal Processing,* vol. 20, n° %17, 2006.

[26] P. Flandrin et al. Time-Frequency/Time-Scale Analysis, San Diego: CA, USA: Academic Press, 1999.

[27] S. Mallat et al. A Wavelet Tour of Signal Processing: The Sparse Way, Burlington: MA, USA: Academic Press, 2009.

[28] S. Yang et al. «Support Vector Machine in Machine Condition Monitoring and Fault Diagnosis,» *Mechanical Systems and Signal Processing,* vol. 21, n° %16, p. 12, 2007.

[29] W. Zhang, «Data-Driven Methods for Predictive Maintenance of Industrial Equipment: A Survey,» *IEEE Systems Journal,* vol. 13, n° %13, p. 8, 2019.

[30] A. Susto et al. «Machine Learning for Predictive Maintenance: A Multiple Classifier Approach,» *IEEE Transactions on Industrial Informatics,* vol. 11, n° %13, p. 6, 2015.

[31] T. Wuest et al. «Machine Learning in Manufacturing: Advantages, Challenges, and Applications,» *Production & Manufacturing Research,* vol. 4, n° %11, p. 9, 2016.

[32] O. Janssens et al. «Convolutional Neural Network Based Fault Detection for Rotating Machinery,» *Journal of Sound and Vibration,* vol. 377, p. 12, 2016.

[33] P. Malhotra, «Long Short Term Memory Networks for Anomaly Detection in Time Series,» *European Symposium on Artificial Neural Networks (ESANN),* p. 8, 2015.

[34] L. Marino et al. «Building Energy Load Forecasting Using Deep Neural Networks,» *42nd Annual Conference of the IEEE Industrial Electronics Society,* p. 9, 2016 .

[35] T. Yairi et al. «Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction,» chez *2nd Workshop on Machine Learning for Sensory Data Analysis*, Gold Coast.

[36] S. Cho et al. «Variational Autoencoder Based Anomaly Detection Using Reconstruction Probability,» *Special Lecture on IE,* vol. 2, n° %11, p. 11, 2015.

[37] D. Li et al.«Anomaly Detection with Generative Adversarial Networks for Multivariate Time Series,» *arXiv,* p. 5, 2019.

[38] C. Zhang et al. «Multi-Layer Graph Convolutional Network for Fault Diagnosis of Rotating Machinery,» *IEEE Transactions on Industrial Electronics,* vol. 68, n° %19, p. 13, 2021.

[39] S. Yin et al. «A Review on Basic Data-Driven Approaches for Industrial Process Monitoring,» *IEEE Transactions on Industrial Electronics,* vol. 61, n° %111, p. 7, 2014.

[40] S. Hariri et al. «Extended Isolation Forest,» *IEEE Transactions on Knowledge and Data Engineering,* vol. 33, n° %14, 2021.

[41] W. Duin et al. «Support Vector Data Description,» *Machine Learning,* vol. 54, n° %11, p. 9, 2004.

[42] A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, Sebastopol: CA, USA: O'Reilly Media, 2019.

[43] M. G. a. S. Uchida, «A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data,» *PLoS ONE,* vol. 11, n° %14, 2016.

[44] G. Athanasopoulos et al. Forecasting: Principles and Practice, Melbourne, Australia: OTexts, 2021.

[45] P. Carvalho et al. «A Systematic Literature Review of Machine Learning Methods Applied to Predictive Maintenance,» *Computers & Industrial Engineering,* vol. 137, 2019.

[46] A. Bousdekis et al.«A Review of Data-Driven Predictive Maintenance: Methods, Datasets and Applications,» *Sensors,* vol. 20, n° %124, 2020.

[47] S. Tuli et al. «TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data,» *Proceedings of the VLDB Endowment,* vol. 15, n° %16, 2022.

[48] M. Alhanaf et al. «Deep Learning-Based Fault Detection and Classification in Smart Grid Systems Using ANN and CNN Models,» *IEEE Access,* vol. 11, 2023.

[49] Y. Huang et al. «Energy Management Strategy Based on LSTM Load Forecasting and Dynamic Programming for Smart Grids,» *Energy Reports,* vol. 9, 2023.

[50] P. Jupyter, «JupyterLab: The Next-Generation Web-Based User Interface for Project Jupyter,»

2024. [online]. Available: https://jupyter.org. [Accès le 09 04 2026].

[51] E. Bisong, Google Colaboratory. In: Building Machine Learning and Deep Learning Models on Google Cloud Platform, Berkeley, CA, USA: Apress, 2019.

[52] M. Corporation, «Visual Studio Code Documentation,» 2024. [online]. Available: https://code.visualstudio.com. [Accès le 15 04 2026].

[53] Object Management Group (OMG), «OMG Unified Modeling Language (OMG UML),,» 2017. [online]. Available: https://www.omg.org/spec/UML/2.5.1/. [Accès le 19 02 2026].

Abstract

This dissertation addresses the challenges of reactive maintenance and limited exploitation of industrial data in power generation systems. The objective is to design and implement an intelligent framework for predictive maintenance and energy optimization within the M'sila Gas Turbine Production Unit of SONALGAZ.

The proposed framework, named **FactoryNova**, integrates three artificial intelligence modules: an Autoencoder-based anomaly detection system for gas turbine monitoring, a CatBoost-based fault diagnosis and severity classification system for rotating machinery, and a stacked ensemble model for electricity demand forecasting and energy optimization. All modules were integrated into a unified web-based platform providing monitoring, predictive analytics, and decision-support functionalities.

Experimental results demonstrated the effectiveness of the proposed approach. The anomaly detection model achieved an accuracy of 94.8% and a ROC-AUC of 0.972. The fault diagnosis module achieved accuracies exceeding 99.9%, while the electricity demand forecasting model obtained a MAPE of 0.2046% and an R^2 of 0.9994. These results confirm the potential of artificial intelligence to improve equipment reliability, support predictive maintenance, and optimize energy management in industrial environments.

Keywords: Artificial Intelligence; Predictive Maintenance; Anomaly Detection; Fault Diagnosis; Energy Optimization; Smart Grid.

الملخص

تتناول هذه المذكرة إشكالية الاعتماد على الصيانة التفاعلية وضعف استغلال البيانات الصناعية في أنظمة إنتاج الطاقة الكهربائية. ويهدف هذا العمل إلى تصميم وتطوير إطار ذكي للصيانة التنبؤية وتحسين استهلاك الطاقة على مستوى وحدة إنتاج التوربينات الغازية التابعة لسونلغاز بالمسيلة.

يعتمد الإطار المقترح، المسمىFactoryNova ، على دمج ثلاثة مكونات للذكاء الاصطناعي: نظام للكشف عن الشذوذات باستخدام نموذج Autoencoder لمراقبة التوربينات الغازية، ونظام لتشخيص الأعطال وتحديد شدتها باستخدام خوارزميةCatBoost ، بالإضافة إلى نموذج تجميعي للتنبؤ بالطلب على الطاقة الكهربائية وتحسين إدارتها. وقد تم دمج هذه المكونات ضمن منصة ويب موحدة توفر وظائف المراقبة والتحليل التنبئي ودعم اتخاذ القرار.

أظهرت النتائج فعالية الحل المقترح، حيث حقق نموذج كشف الشذوذات دقة بلغت 94.8% وقيمة ROC-AUCتساوي 0.972، بينما تجاوزت دقة نظام تشخيص الأعطال 99.9%. كما حقق نموذج التنبؤ بالطلب الكهربائي قيمة MAPE بلغت 0.2046% ومعامل تحديد R^2 يساوي 0.9994. وتؤكد هذه النتائج قدرة تقنيات الذكاء الاصطناعي على تحسين موثوقية المعدات ودعم الصيانة التنبؤية وترشيد إدارة الطاقة في البيئات الصناعية.

الكلمات المفتاحية: الذكاء الاصطناعي؛ الصيانة التنبؤية؛ كشف الشذوذات؛ تشخيص الأعطال؛ تحسين استهلاك الطاقة؛ الشبكات الذكية.