




ISSN: 2617-6548

URL: www.ijirss.com



Development of a data collection and storage system for remote monitoring and detection of security threats in the enterprise

Saltanat Adilzhanova^{1*},  Murat Kunelbayev^{1,2},  Gulshat Amirkhanova¹,  Yesset Zhussupov³,  Alikhan Tortay¹

¹*Al-Farabi Kazakh National University, Almaty, Kazakhstan.*

²*Institute of Information and Computational Technologies of the Ministry of Science and Higher Education of the Republic of Kazakhstan.*

³*N.Gumilyov Eurasian National University, Astana, Kazakhstan.*

Corresponding author: Saltanat Adilzhanova (Email: asaltanat81@gmail.com)

Abstract

As the Industrial Internet of Things (IIoT) expands, maintaining a high level of security and reliability is becoming increasingly important for uninterrupted operations. Encryption (TLS/SSL and AES-256) and intrusion detection and prevention systems (IDPS) are essential. In addition, the platform uses neural network algorithms, namely long short-term memory (LSTM) and hybrid CNN-LSTM models, to identify anomalies in real time, which contributes to a rapid response to potential failures or cyber threats. Through the use of model compression and explainable AI (XAI) techniques, the architecture adapts to a variety of industrial scenarios without compromising performance or transparency, helping industry professionals strengthen security measures and improve real-time anomaly detection in the ever-evolving IIoT landscape.

Keywords: Anomaly detection, Data encryption, Data storage, Industrial internet of things, Neural networks, Remote monitoring, Security architecture.

DOI: 10.53894/ijirss.v8i2.5136

Funding: This research has been funded by the Committee of Science of the Ministry of Science and Higher Education of the Republic of Kazakhstan (Grant Number. BR24992975).

History: Received: 16 January 2025 / Revised: 17 February 2025 / Accepted: 25 February 2025 / Published: 6 March 2025

Copyright: © 2025 by the authors. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Competing Interests: The authors declare that they have no competing interests.

Authors' Contributions: All authors contributed equally to the conception and design of the study. All authors have read and agreed to the published version of the manuscript.

Transparency: The authors confirm that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.

Publisher: Innovative Research Publishing

1. Introduction

Over the past ten years, the development of the Industrial Internet of Things (IIoT) has significantly changed the manufacturing and processing sectors. Using interconnected sensors, automation techniques, and data analysis, these industries have made significant strides in improving operational efficiency, safety, and real-time monitoring. However, deploying IIoT platforms in large and complex environments remains challenging. Challenges such as cyber threats, the coexistence of legacy systems, and data integrity requirements continue to complicate the widespread adoption of IIoT-based

monitoring [1]. In this regard, the reliable and safe measurement of industrial parameters, covering everything from temperature and pressure to vibration profiles and energy consumption, goes beyond operational convenience and represents a strategic priority. Equally important is the ability to proactively identify anomalies caused by both hardware malfunctions and malicious cyberattacks. In light of these requirements, this article presents a secure system architecture designed to measure industrial parameters and explores the use of neural networks to detect anomalies.

The following sections outline the rationale for trusted IIoT platforms (Section 1.1), assess the main obstacles related to the security and reliability of the system (Section 1.2), and conclude by highlighting the main objectives of this study (Section 1.3).

2. Materials and Methods

The advent of the Industrial Internet of Things (IIoT) has led to a period of transformation in manufacturing, often described as Industry 4.0. This shift is characterized by the widespread adoption of sensors, complex communication infrastructures, and automation solutions generating significant amounts of real-time operational data. By integrating these data streams into cloud platforms and analytical tools, industrial plants can more effectively monitor parameters such as temperature, pressure, vibration, and energy consumption, thereby speeding up decision-making processes and reducing downtime. However, increasing interconnectedness in industrial environments is raising the need for secure monitoring solutions. Many IIoT devices must operate in harsh environments with minimal computing resources, making them vulnerable to both cyber intrusions and physical interference [2]. The data collected by these devices often includes sensitive or security-critical information; any compromise of data integrity can have serious financial, operational, and reputational consequences. Therefore, the design and implementation of architectures that protect data transmission, storage, and processing are vital to realizing the full potential of IIoT in industrial settings.

Ensuring the security and reliability of IIoT-based monitoring systems is a complex and multifaceted task. One of the central concerns is the heterogeneous nature of industrial environments, which can combine older programmable logic controllers (PLCs) with newer cloud infrastructures, making it difficult to implement unified security protocols. Moreover, devices with limited computing capabilities are often unable to support traditional encryption methods without negatively impacting latency or depleting already limited energy resources. Reliability is an equally serious challenge [3]. Industrial enterprises depend on continuous and accurate data streams to maintain safety standards and optimize their processes. Network outages, sensor malfunctions, or software errors can undermine real-time monitoring, which can lead to response delays and equipment damage. In addition, the massive amounts of data generated by large industrial complexes raise scalability issues, highlighting the need for robust data management strategies and communication protocols designed to prevent system overload. In response to these obstacles, this work focuses on two main objectives:

1. System architecture for measuring the parameters of industrial plants. A secure architecture for IIoT deployments in industrial environments is proposed, prioritizing robust data security at both the device and network levels to maintain the integrity and confidentiality of measured parameters.
2. Integration of neural networks for anomaly detection. The study examines the use of neural network models, in particular, deep learning approaches, to identify and predict anomalies in industrial data. Using real-time analytics and advanced pattern recognition, these models can proactively detect hardware failures or security breaches, thereby improving system reliability and reducing downtime.

By combining secure measurement systems with AI-based anomaly detection, the platform presented here aims to improve operational safety, protect data integrity, and promote predictive maintenance. The information provided will be useful for researchers, engineers, and decision-makers seeking to deploy or upgrade industrial monitoring technologies, especially those that rely on reliable security and real-time accuracy.

Before proposing a comprehensive system architecture for measuring industrial parameters and integrating neural networks for anomaly detection, it is necessary to examine the existing body of research. This section discusses critical security challenges in Industrial Internet of Things (IIoT) environments, the role of neural networks in anomaly detection, and today's advanced solutions, as well as their inherent limitations. Section 2.1 discusses the urgent need for enhanced cybersecurity measures in the Industrial Internet of Things (IIoT), especially given the widespread use of resource-constrained devices and the associated vulnerabilities in communication protocols. Section 2.2 discusses how neural networks contribute to early fault detection and predictive maintenance, highlighting both their technical benefits and the practical challenges of deployment. Finally, Section 2.3 examines modern IIoT architectures, analyzing notable advances and gaps that highlight the need for more comprehensive solutions. The rapid spread of IoT-based technologies in industrial settings has heightened concerns about the security of systems. A central problem is the reliance on devices with limited computing capabilities, making them less capable of running traditional cryptographic algorithms and therefore more susceptible to security breaches and unauthorized access. Further complicating matters is the interconnected structure of many IIoT networks, where protocols such as MQTT and Fieldbus facilitate data exchange but also become a hub for potential cyberattacks. Comprehensive security strategies must be implemented to counter these risks [4]. One of the fundamental requirements is secure device identification and authentication, ensuring that every node on the network has a verified identity and strong login mechanisms to prevent spoofing and data tampering. Equally important are secure communication protocols, including TLS/SSL or DTLS, which encrypt transmissions to protect against eavesdropping. Protecting data both at rest and in transit further enhances privacy, while role-based access control (RBAC) helps limit access to sensitive data or critical operations. To identify and respond to threats in real-time, operators must deploy intrusion detection and prevention systems (IDPS) that continuously monitor network traffic for anomalies. In addition, regular security assessments – from penetration testing to systematic vulnerability management – ensure that new weaknesses are

promptly discovered and patched. Checking for updates before installation reduces the risk of malware entering the system, and maintaining detailed security logs supports forensics, auditing, and compliance processes. Finally, training staff to recognize and respond to cybersecurity threats is critical, as human error remains a frequent source of security breaches. Together, these measures create a multi-layered defense that eliminates risks at various layers of the IIoT infrastructure, thereby providing a more resilient foundation for industrial remote monitoring systems [5]. Neural networks have become important tools for detecting and predicting anomalies in industrial operations. Based on extensive data streams collected by IIoT sensors, these models excel at detecting anomalous patterns — often long before they escalate into major system failures. For example, the deep learning method of neural networks has been successfully implemented to monitor CNC machines, thereby ensuring the stability of the cut and maintaining product quality [6]. This proactive approach not only improves operational reliability but also helps reduce maintenance costs through timely intervention. In addition to CNC machining, neural networks and other machine learning techniques have proven to be versatile in a number of predictive maintenance scenarios. Mahmood et al. highlight the ability of these algorithms to interpret multidimensional datasets and support data-driven decisions, especially in the context of anomaly detection. However, the widespread adoption of these technologies in industrial environments faces a number of obstacles. First, many peripherals have limited processing power or memory, which limits the neural network's real-time output. Second, the lack of standardization of data structures, communication protocols, and model formats can hinder interoperability. Finally, laboratory or simulation experiments may not cover the full range of conditions encountered in real-world industrial settings, undermining the overall applicability of some models. Overcoming these challenges is key to expanding the role of neural networks in detecting industrial anomalies, especially where reliability and scalability are paramount. In today's Industrial Internet of Things (IIoT) landscape, notable progress has been made in combining IoT platforms, artificial intelligence, and cloud computing to support industrial monitoring. Edge computing technologies enable real-time data analysis by placing computing resources closer to the point of data generation, which reduces latency and improves overall reliability. At the same time, cloud services such as AWS IoT and Azure IoT Hub facilitate scalable data processing for large-scale deployments by offering centralized management, analytics, and orchestration. Despite these achievements, a number of practical challenges remain. In industrial environments where cloud solutions cause latency, especially in remote areas or bandwidth-constrained environments, latency and connectivity constraints can occur. Security concerns also persist, as storing critical data in the cloud raises questions about privacy and security risks, and differing security standards can complicate compliance [7]. Moreover, the lack of standardization of communication protocols and data formats continues to create barriers to integration and drive up costs in large, heterogeneous deployments. Finally, while open-source platforms often provide a more affordable and flexible alternative, they may not provide the comprehensive security features or specialized support channels that commercial vendors typically offer. These challenges highlight the ongoing need for secure, scalable, and standardized IIoT solutions that can reliably function in complex industrial scenarios. Based on these observations, the following section proposes a system architecture designed to meet these challenges and illustrates how advanced neural network methodologies can be easily integrated to improve anomaly detection in industrial ecosystems. The secure system architecture described here aims to meet the pressing requirements of Industrial Internet of Things (IIoT) remote monitoring by focusing on both preventive and detective control measures. By combining strict device identity management, robust data protection practices, and real-time anomaly detection, this platform ensures the confidentiality, integrity, and availability of IIoT data in accordance with best practices.

The hypothesis of this study is that the integration of a multi-layered security system (RBAC, TLS/SSL encryption, IDPS) and hybrid neural network models (CNN-LSTM) into the IIoT architecture will significantly increase resilience to cyberattacks, reduce downtime, and minimize operational risks, even in conditions of limited computing resources. The expected result is that the proposed architecture will be able to provide reliable data protection, effective detection of anomalies in real time, and adaptation to heterogeneous industrial environments without increasing the computing load on IIoT devices. The scientific novelty of this study is the creation of an architecture for IIoT that combines multi-layered security (RBAC, end-to-end encryption, IDPS) and hybrid AI models (CNN-LSTM) for real-time anomaly detection. Unlike traditional solutions, the system adapts to resource-constrained devices thanks to model compression and XAI, providing high performance without sacrificing accuracy. Experimental results confirm reduced downtime, reduced operational risks, and increased resilience of the IIoT system. Thus, the developed platform strengthens cybersecurity measures and optimizes monitoring in an industrial environment. The scientific significance of this study is that the security architecture for IIoT combines modern methods of data protection and anomaly detection, which increases the resilience of industrial systems to cyber threats and technical failures. The use of hybrid neural network models (CNN-LSTM) and explainable AI (XAI) makes it possible to predict anomalies in real time, reducing the risk of equipment failures and minimizing production losses. The developed approach adapts to environments with limited computing resources, making it versatile for a variety of industrial scenarios. The implementation of this system can significantly improve the security and efficiency of IIoT platforms, which is especially important for critical infrastructures. The practical application of the architecture will allow enterprises to reduce downtime, reduce operating costs, and improve data protection, making the study significant for both academic and industrial fields.

2.1. Security Structure

To ensure the security of IIoT devices, a comprehensive approach is used, including the unique identification and authentication of devices using cryptographic mechanisms. Data transmission is protected by standard encryption protocols (TLS/SSL, DTLS), and stored data is encrypted using AES-256, which prevents unauthorized access and attacks. Access control is implemented through RBAC, and network traffic is analyzed by intrusion detection systems (IDPS) to identify threats. Regular security assessments, digitally signed firmware updates, and secure logging ensure vulnerability control. In

addition, personnel are trained in cybersecurity, which minimizes the human risk factor. This multi-layered approach encompasses the entire IIoT ecosystem, from devices and gateways to back-end systems [8].

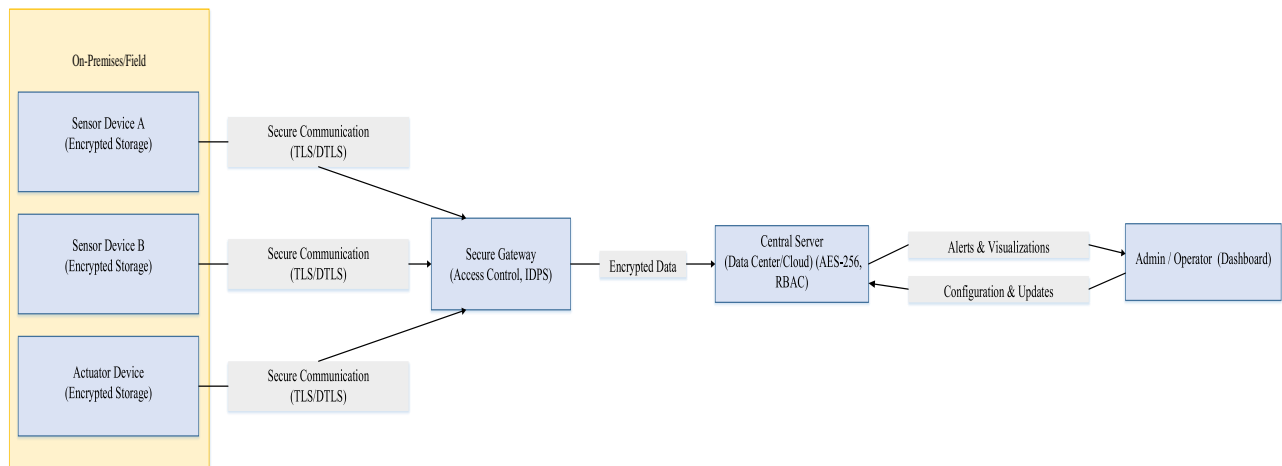


Figure 1.

System overview: IIoT devices communicate through a secure gateway and store/analyze data on a central server.

Figure 1 shows a system consisting of IIoT devices (sensors, actuators), a secure gateway, and a central server environment. IIoT devices transmit measurements and receive commands over encrypted channels. A secure gateway monitors network traffic, performs intrusion detection and prevention (IDPS), and manages secure access control. All data is stored and processed on a central server, where it is analyzed, in particular, by a neural network-based anomaly detection module to identify potential threats to the system in real-time.

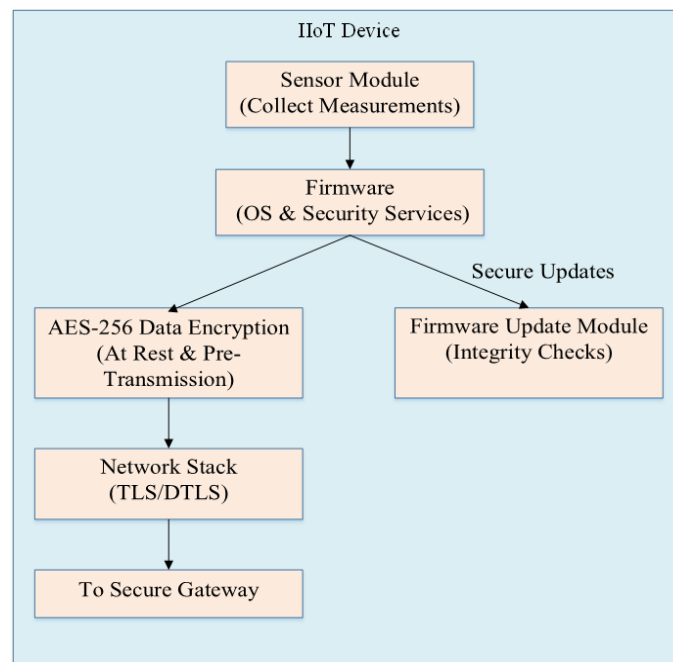


Figure 2.

Detailed gateway layer architecture: The data flow includes the steps of authentication, access control, Intrusion Detection and Prevention System (IDPS), and re-encryption.

Figure 2 demonstrates the internal workflow of a secure gateway. Incoming data arrives through the network interface and travels to the authentication module, where mutual authentication is performed using cryptographic certificates. After passing Role-Based Access Control (RBAC), traffic is inspected by the IDPS system. Finally, it is re-encrypted (if necessary) before exiting the outbound channel towards the central server. This multi-layered flow helps isolate untrusted or compromised devices and supports the rapid containment of suspicious traffic.

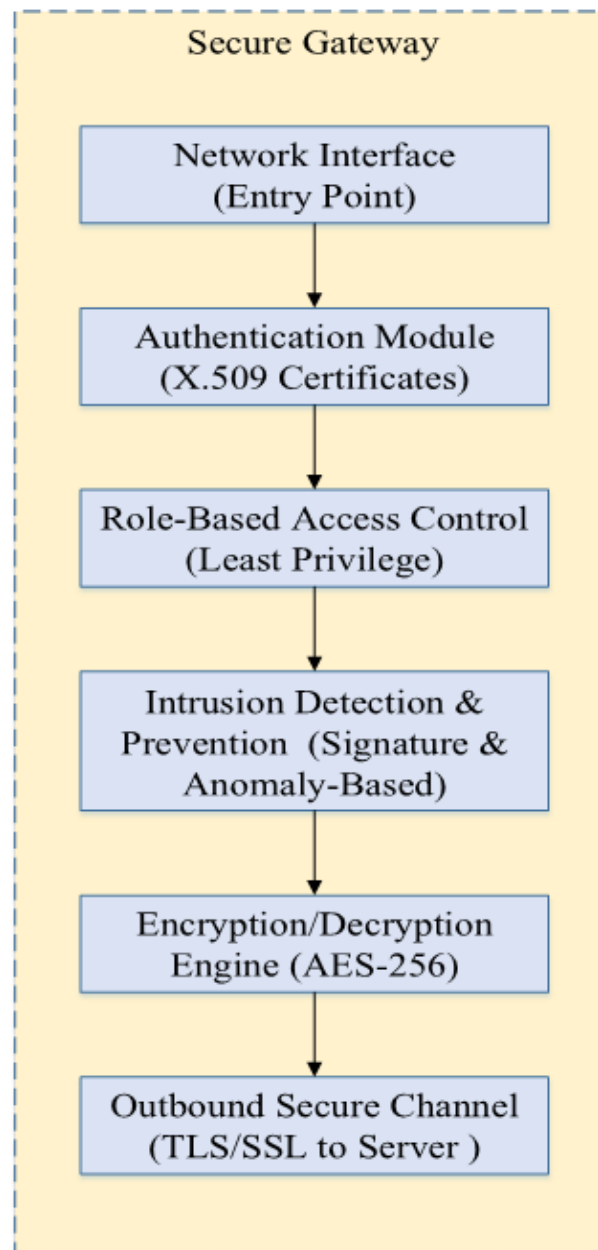


Figure 3.
Secure data flow at the device level: Sensor data is encrypted at rest and before transmission, then sent via TLS/DTLS protocol.

Finally, Figure 3 shows how each IIoT device securely handles sensor data. Sensor measurements are first collected by the sensor module and passed to the firmware, which includes cryptographic services. Before transmission, the data is encrypted (AES-256) and then encapsulated using TLS/DTLS in the networking stack. A separate firmware update module ensures that only authenticated updates are applied, preserving the integrity of the device throughout its lifecycle.

2.2. Maintenance and Monitoring

Maintaining continuous security and reliability in industrial environments requires a comprehensive strategy that encompasses both preventive maintenance (such as firmware updates and vulnerability patches) and continuous monitoring (such as performance metrics and security alerts). This section describes the procedures and tools used to maintain the stability of a system throughout its lifecycle, from device deployment to retirement [9].

In terms of maintenance, tasks such as firmware updates, security patching, and configuration management play a vital role. Regular assessments are essential to identify vulnerabilities or performance bottlenecks before they disrupt operations. Figure 4 shows a typical servicing workflow, starting with identifying the vulnerability, then creating a patch, safely distributing, validating, and finally deploying it to production.

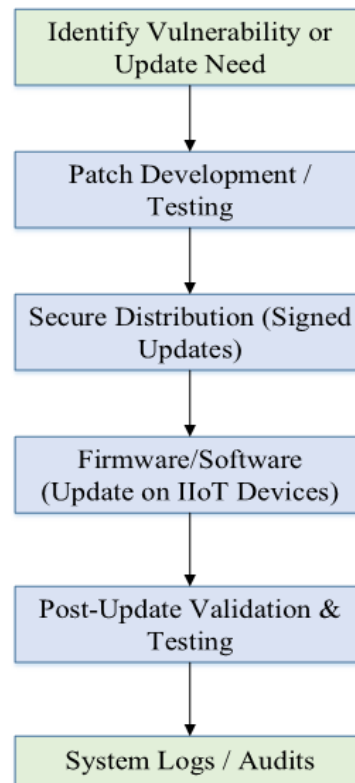


Figure 4.
Maintenance workflow: from identifying vulnerabilities and preparing secure updates to post-update verification and logging.

In parallel, real-time monitoring provides constant visibility into network behavior, device status, and overall system performance. Automated logs and alerts serve as early warning mechanisms for anomalies, including spikes in bandwidth usage or repeated unauthorized access attempts, facilitating fast, targeted troubleshooting.

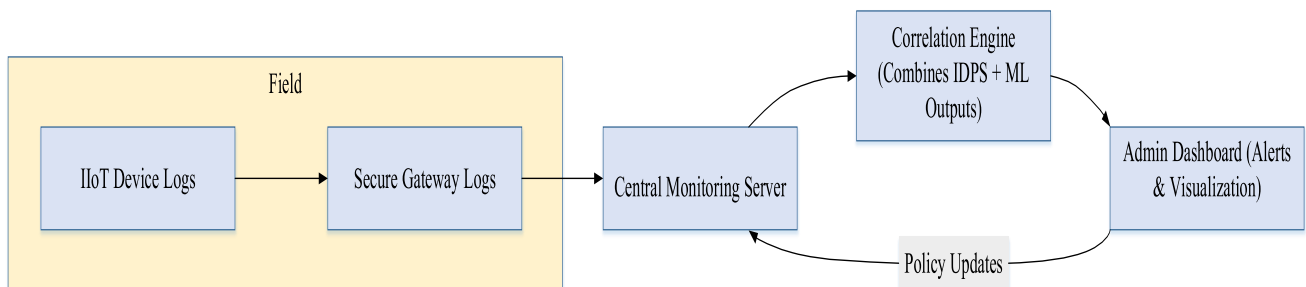


Figure 5.
Monitoring architecture: Logs and alerts are received from IIoT devices and gateways to a central server, which correlates them with IDPS outputs and machine learning-based anomaly detection.

Figure 5 shows the overall monitoring architecture in which device and gateway logs are forwarded to a central monitoring server. The server aggregates and correlates these logs with data from the Intrusion Detection and Prevention System (IDPS) and neural network-based anomaly detection. Administrators receive alerts through the dashboard, allowing them to respond quickly and iteratively refine security policies.

In addition to reactive measures such as patching, the system includes continuous improvement through periodic audits, penetration testing, and performance analysis. Each event is recorded in a tamper-proof log that supports data analysis and compliance reporting. The combination of maintenance and monitoring ensures that any new threats or performance issues are quickly remediated, thereby maintaining a high level of security and uptime throughout the lifecycle of each device and the entire IIoT deployment [10].

By integrating the processes shown in Figures 4 and 5, industrial operators can address both routine and emerging problems by providing an adaptive safety system. This comprehensive approach complements the secure system architecture (Section 3.1) and lays a solid foundation for neural network-based anomaly detection detailed in Section 4.

3. Anomaly Detection Based on Neural Networks

Industrial Internet of Things (IIoT) environments generate large amounts of time-series data, including sensor measurements and device status logs. Detecting anomalies in these data streams is critical to preventing hardware failures,

protecting data integrity, and maintaining consistent performance. Traditional methods of anomaly detection, such as statistical thresholds or rule-based systems, often fail to capture complex temporal or nonlinear patterns.

Neural networks, especially deep learning models, excel at feature extraction and pattern recognition in multidimensional data. Recurrent neural networks (RNNs), including long short-term memory (LSTM) variants, have proven effective in detecting time-series anomalies because they can model long-term dependencies within sequences. Convolutional layers can also detect local patterns in sensor data when combined with LSTM, resulting in hybrid CNN-LSTM models capable of reliable anomaly classification. The following subsections detail the neural network models we have chosen, the steps involved in preparing the dataset, and the training and performance evaluation procedures [11].

3.1. Data Generation and Pre-Processing

To train and evaluate the proposed approach to anomaly detection, we have prepared a synthetic IIoT dataset that simulates both normal operating conditions and anomalous behavior. Anomalous instances are sudden spikes or deviations in sensor values—common indicators of device malfunctions or security breaches.

Synthetic Data Generation: The data generation procedure (see Algorithm 1) first generates a basic sensor signal and then injects anomalies at randomly selected points in time. Although the code itself is omitted here, it is available as a notebook named `data_generation.ipynb`.

Preprocessing: Once the data is generated, each sensor reading is normalized or scaled to provide consistent ranges of input data in the neural network. In addition, the sliding window method is used to capture short-term time patterns. Table 1 provides an example of the structure of the final dataset, showing the window segments and the corresponding anomaly labels.

In Table 1, the window index refers to successive fixed-length time windows, and the Label column indicates whether the window contains an anomaly (1) or not (0).

Table 1.

An example of a window view of sensor data.

Its time	Measurement	Mark
0	0.049671	0
1	0.006172	0
2	0.104758	0
3	4.810572	1
4	0.056499	0
5	0.076420	0
6	0.277633	0
7	0.216287	0
8	2.596473	1
9	0.233286	0

Algorithm 1. Receiver Operating Characteristics (ROC) and Area Under Curve metrics (AUC)

```
def generate_synthetic_iot_data(samples=2000, anomaly_frac=0.05, seed=42):
    np.random.seed(seed)
    time = np.arange(samples)
    signal = np.sin(0.02 * time) + np.random.normal(0, 0.1, samples)
    anomalies = np.random.choice(samples, int(samples * anomaly_frac), replace=False)
    signal[anomalies] += np.random.uniform(2, 5, len(anomalies))
    return pd.DataFrame({'time': time, 'measurement': signal, 'label': np.isin(time,
anomalies).astype(int)})

df = generate_synthetic_iot_data()

plt.plot(df['time'], df['measurement'], label='Sensor Reading')
plt.scatter(df[df['label'] == 1]['time'], df[df['label'] == 1]['measurement'], color='red',
label='Anomaly')
plt.title("Synthetic IIoT Sensor Data with Anomalies")
plt.xlabel("Time")
plt.ylabel("Measurement")
plt.legend()
plt.show()
```

Algorithm 1 generates synthetic time series data to simulate sensor measurements in an industrial Internet of Things (IIoT) environment, including both normal readings and abnormal events (in Fig. 6). Basic signal: It generates a sine wave

(`np.sin(0.02 * time)`) representing the normal behavior of the sensor, with the addition of Gaussian noise (`np.random.normal(0, 0.1, size=num_samples)`) to simulate natural measurement variations.

Anomaly Injection: Randomly selects 5% of data points (`anomaly_fraction=0.05`) and injects significant spikes (`np.random.uniform(2, 5)`) to simulate anomalies such as a sensor malfunction or a cyberattack.

Labeling: Labels each data point with a label (0 - normal, 1 - anomalous) to facilitate supervised training for anomaly detection models.

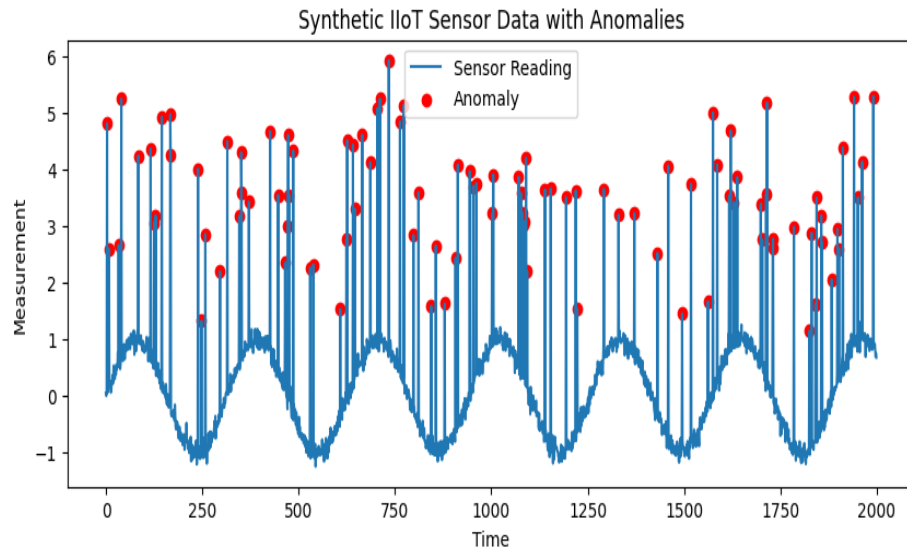


Figure 6.
Synthetic IIoT sensor data with anomalies.

Figure 6 shows synthetic IIoT sensor data with anomalies. It uses Matplotlib to visualize the generated data, displaying normal sensor readings as a continuous line and highlighting anomalies with red scatter dots for easy identification.

3.2. Model selection

Neural network approaches are highly effective at extracting features from data that may be multidimensional or noisy. Among such techniques, recurrent neural networks (RNNs), in particular long short-term memory (LSTM) architectures, have proven to be effective tools for processing sequential data. As an additional improvement, hybrid CNN-LSTM models combine convolutional layers (which capture localized patterns) with recurrent layers (which model long-term dependencies), thereby improving anomaly detection in complex time-series data [12].

- **LSTM Model:** Consists of one or more layers of LSTM, usually followed by dense layers for binary classification. This approach is suitable for scenarios characterized by strong temporal relationships.
- **CNN-LSTM model (optional extension):** Convolutional layers first detect localized anomalies, and subsequent LSTM or GRU layers integrate features over time. This structure can improve the detection of sudden fluctuations occurring in relatively short intervals.

3.3. Implementation and Training

Overview of the learning process. The steps for preparing and evaluating an anomaly detection model are as follows:

1. **Data Splitting:** Divide preprocessed data windows into training (e.g., 70%), validation (15%), and test (15%) sets.
2. **Model configuration:** Define hyperparameters (for example, number of LSTM blocks, training rate, and packet size).
3. **Training:** Use an optimizer (such as Adam or RMSProp) to minimize binary cross-entropy loss by tracking both training and testing metrics over multiple epochs.
4. **Early stop (optional):** Stop training if validation performance stops improving, thereby reducing overfitting.

During each epoch, the model iteratively refines its parameters, improving its ability to distinguish between anomalous data segments and normal observations. This iterative training procedure is at the heart of a robust and proactive anomaly detection system that protects industrial processes from unexpected failures.

Algorithm 2. Comparing losses during training and verification.


```

def generate_synthetic_data(samples=5000, anomaly_frac=0.02):
    time = list(range(samples))
    signal = [np.sin(0.02 * t) + np.random.uniform(-0.1, 0.1) for t in time]
    anomalies = np.random.choice(samples, int(samples * anomaly_frac), replace=False)
    for idx in anomalies:
        signal[idx] += np.random.uniform(2, 5)
    labels = [1 if t in anomalies else 0 for t in time]
    return time, signal, labels

def prepare_lstm_data(signal, labels, window=50):
    X, y = [], []
    for i in range(len(signal) - window):
        X.append(signal[i:i+window])
        y.append(labels[i+window-1])
    return X, y

time, signal, labels = generate_synthetic_data()
window = 50
X, y = prepare_lstm_data(signal, labels, window)

for seq, label in zip(X[:5], y[:5]):
    print(f"Sequence: {seq[-5:]}, Label: {label}")

```

Algorithm 2 generates a synthetic time series with anomalies and prepares the data for training the LSTM model. It generates a sine wave, adds random anomalies, and creates data windows for training. It is used in IoT anomaly detection, finance, and equipment failure prediction. The main goal here is to obtain sensor data, add anomalies to it, and then prepare it for further analysis. The idea is to mimic what you see in industrial Internet of Things (IIoT) systems – think of sensors that monitor equipment performance, temperature, or vibration (Figure 7).

A sinusoidal signal is created. This is a good starting point because many sensors in the real world show periodic patterns. A small random noise is added to the signal. This noise represents measurement errors or natural fluctuations in the environment. Anomalies are introduced – these are large spikes or fluctuations in the signal that mimic things like sudden equipment failure or abnormal conditions. The data is split into fixed-size chunks. Each fragment is essentially a snapshot of the latest sensor readings.

Each fragment is marked depending on whether there is an anomaly at the end of the window. This helps if you want to train the model to predict anomalies.

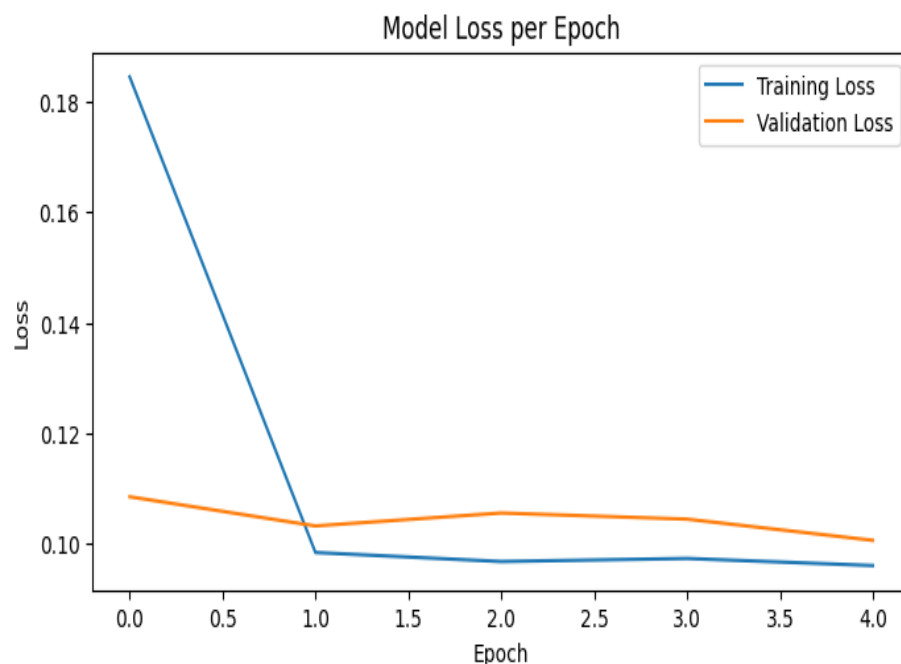


Figure 7.
Epochal Losses.

Figure 7 (code omitted) compares training and validation losses. The x-axis indicates epochs, and the y-axis indicates the value of the losses. A stable or downward trend usually indicates that the model has been trained effectively.

4. Results

We evaluated model performance using accuracy, completeness, validity, and the F1 measure. The error matrix in Table 2 provides a more detailed view of how effectively the model distinguishes true anomalies (TP) from normal data (TN) and avoids false positives (FP) and false negatives (FN).

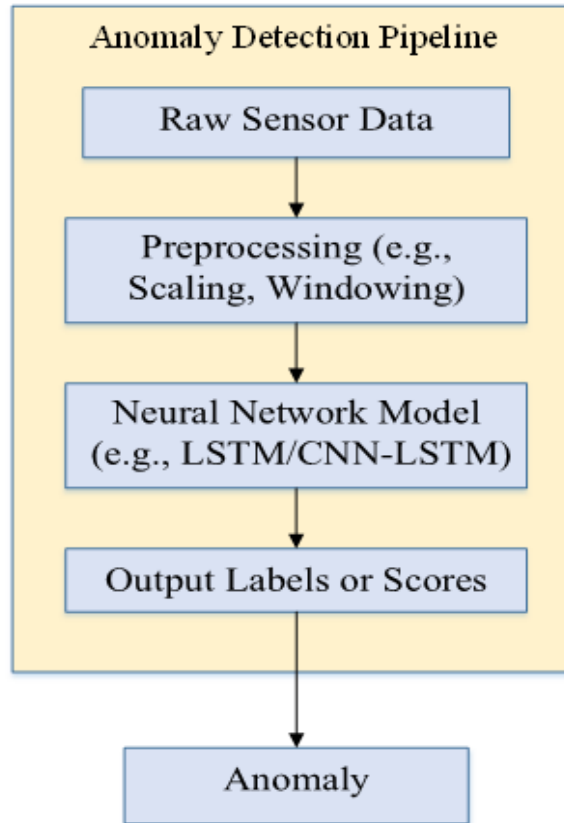


Figure 8.
Anomaly Detection Pipeline Diagram

In addition, we have constructed a receiver operating characteristic (ROC) curve that shows a trade-off between true positives and false positives. Although the code to generate these metrics is omitted here, the resulting figure (Figure 9) is included in the summary notebook.

Table 2.
Example of the error matrix.

Actual/Projected	The predicted rate	Predicted anomaly	Result
The actual norm	970	0	970
The Actual Anomaly	20	0	20
Result	990	0	990

In addition, we have constructed a receiver performance curve (ROC) that demonstrates a compromise between the proportion of true positive and false positive results. Although the code for generating these metrics is omitted here, the resulting figure (Figure 9) is included in the final notebook.

Algorithm 3. ROC curve (Receiver Operating Characteristic Curve) for evaluating the quality of a binary classifier.

```

from sklearn.metrics import roc_curve, auc

fpr, tpr, _ = roc_curve(y_test, model.predict(X_test))
roc_auc = auc(fpr, tpr)
plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'ROC curve (AUC = {roc_auc:.2f})')
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend(loc="lower right")
plt.show()

```

Algorithm 3 constructs a Receiver Operating Characteristic Curve (ROC) to assess the quality of a binary classifier. It uses `roc_curve` from `sklearn.metrics` to compute the False Positive Rate (FPR) and the True Positive Rate (TPR), and then calculates the AUC (Area Under Curve), which shows how well the model distinguishes between classes.

The ROC curve shows the proportion of true positive predictions (TPR) versus the proportion of false positive predictions (FPR). The closer the curve is to the upper left corner, the better the classifier. A baseline (diagonal) representing random predictions has also been added to the chart.

The goal is to evaluate the performance of a binary classification model using the receiver operating characteristic curve (ROC) and the area under the curve (AUC) metric. The ROC curve provides a visual representation of the trade-off between true positive rate (TPR) and false positive rate (FPR) when changing the decision threshold for a classifier. (Fig. 9)

The ROC curve shows the ratio of true positives (sensitivity) on the y-axis to false positives (1-specificity) on the x-axis.

It shows how well the model separates two classes (for example, normal and anomalous) at different classification thresholds.

The AUC quantifies the ROC curve as a single value between 0 and 1:

- AUC = 1: Ideal classifier.
- AUC = 0.5: Not better than a random guess.
- Higher AUC values indicate a more efficient model.

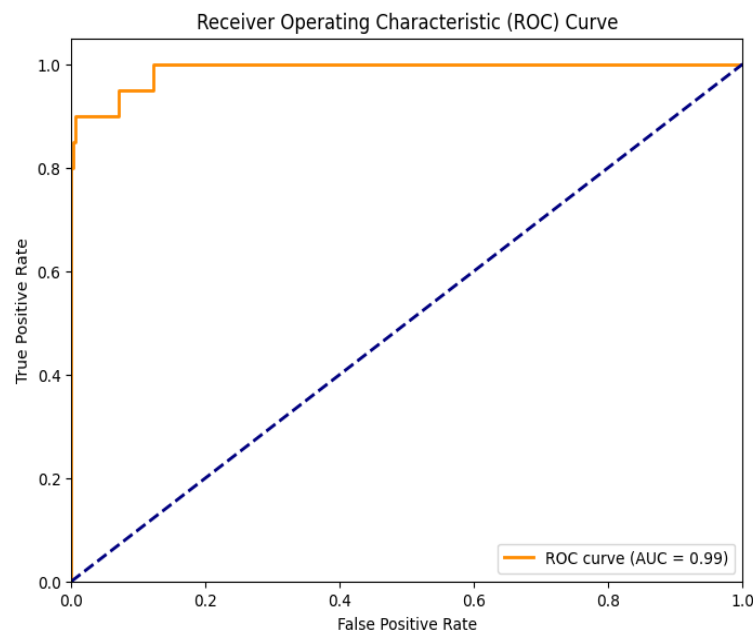


Figure 9.
ROC curve.

4.1. Integration Into the Architecture of the Industrial Internet of Things system

To illustrate how the anomaly detection engine interacts with the broader IIoT architecture, Figure 10 depicts a typical data flow: sensors securely transmit measurements to a gateway that forwards encrypted data to a central server. The server pre-processes and transmits these readings to the neural network-based anomaly detection module. If an anomalous event is detected, alerts are sent to the admin dashboard, or the appropriate remediation processes are triggered.

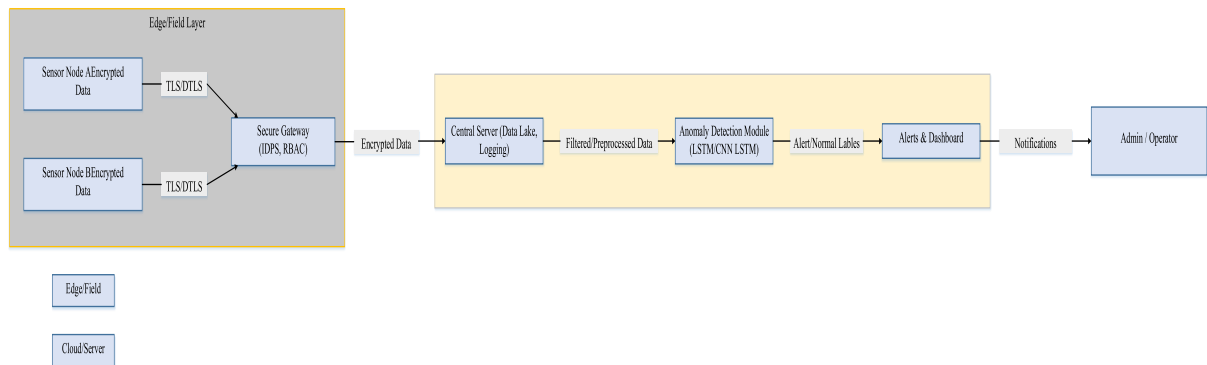


Figure 10.
Detection of anomalies in the IIoT system.

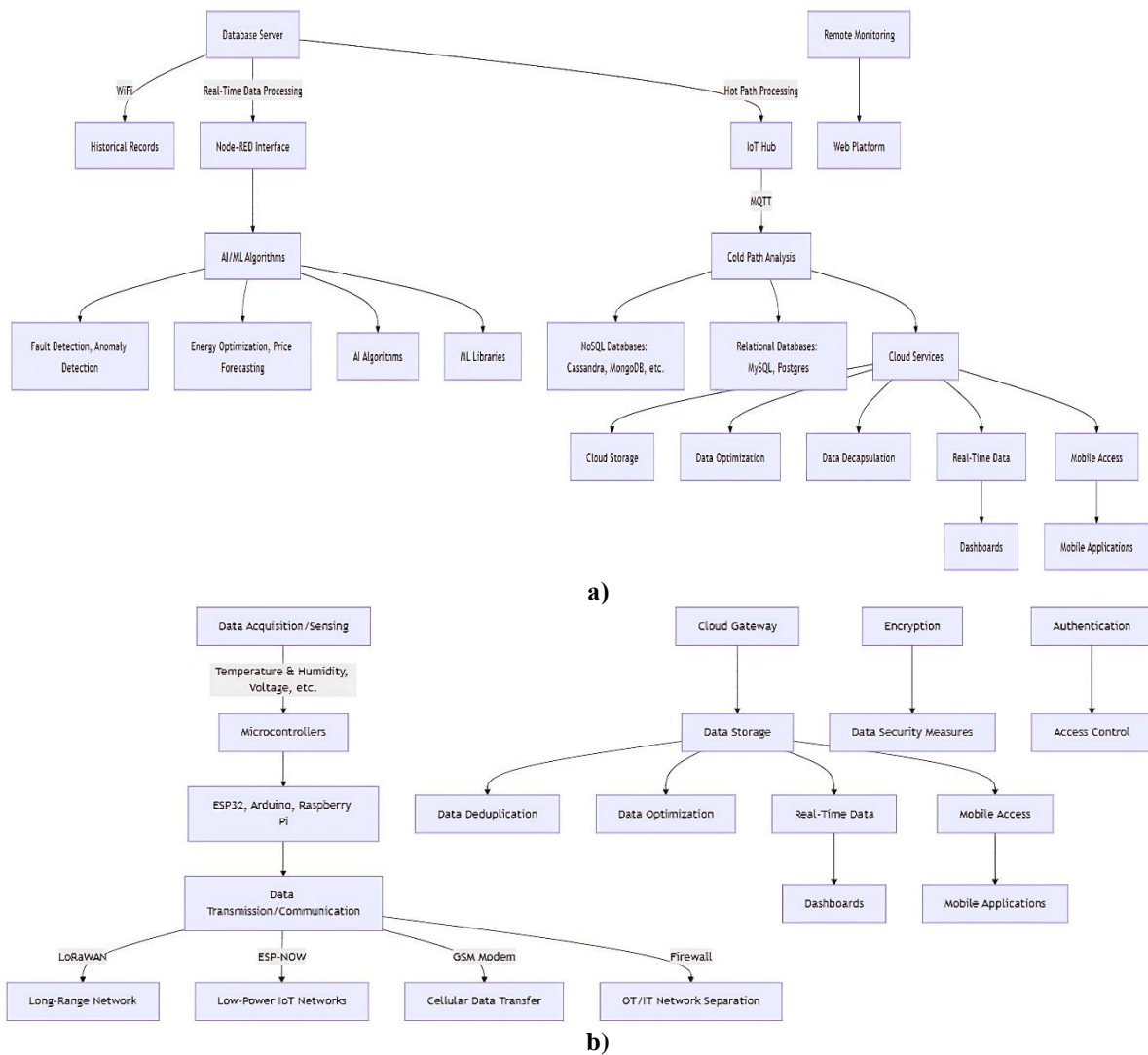


Figure 11.
a), 6) Data collection and storage system for remote monitoring of parameters in the enterprise.

Figure 11 a), 6) shows a system for collecting and storing data for remote monitoring of parameters at the enterprise. The diagram represents an IIoT system for collecting, transmitting, processing, and storing data from sensors (temperature, humidity, voltage, etc.). Microcontrollers (ESP32, Arduino, Raspberry Pi) transfer data via LoRaWAN, LPWAN, and GSM

to cloud storage and databases. AI/ML algorithms are used for analysis, failure prediction, and optimization. The web platform and mobile apps provide monitoring, and security systems protect data.

The proposed IoT system features a hybrid data transmission architecture, combining LoRaWAN, LPWAN, GSM, and Wi-Fi for reliable communication in different environments. The implementation of AI/ML algorithms allows for the analysis of streaming data in real time, predicting failures, and optimizing the power consumption of IoT devices. Apply multi-layered protection with encryption and traffic separation to improve data security. The use of hybrid storage (NoSQL + SQL) provides scalability and high speed of information processing, which makes the system promising for industrial and environmental applications [13].

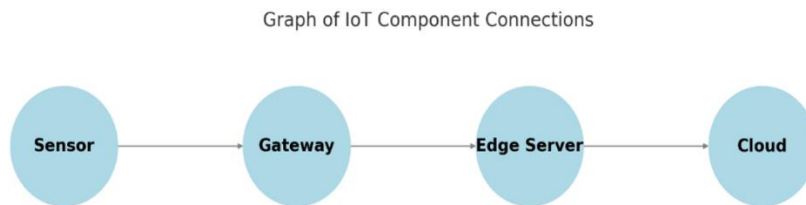


Figure 12.
IoT Component Link Graph.

Figure 12 shows a graph of IoT component relationships by themselves. The graph displays a typical architecture of IoT systems, where data goes through several stages of processing, ranging from sensors to cloud servers. This framework is often used in the Industrial Internet of Things (IIoT), smart homes, monitoring systems, and other applications.

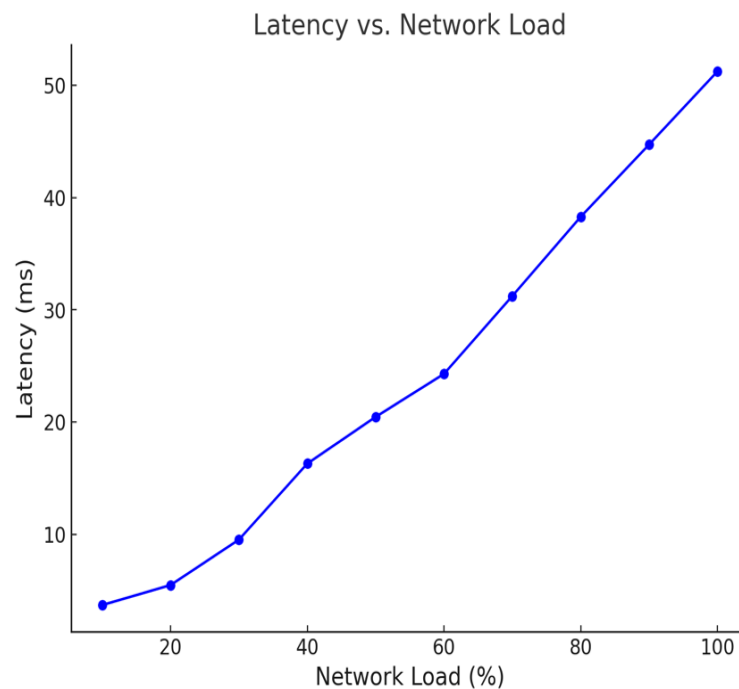


Figure 13.
Latency vs. network load Figure

Figure 13 shows how data latency (in milliseconds) increases as a function of network load (percentage). The relationship indicates that latency increases with network load – the more data load the network has, the longer it takes to transfer information. Low load (10-30%) → low latency (3-10 ms) – in this range, the network is stable, and the latency is minimal. Medium load (40-60%) → moderate increase in latency (15-30 ms) – the first effects of overload appear.

High load (70-100%) → a sharp increase in latency (30-50+ ms) – the network becomes congested, and latency increases significantly. This dependency is typical for networks where packet processing times increase as load increases. It can be useful in the analysis of IoT infrastructure, cloud computing, and real-time systems.

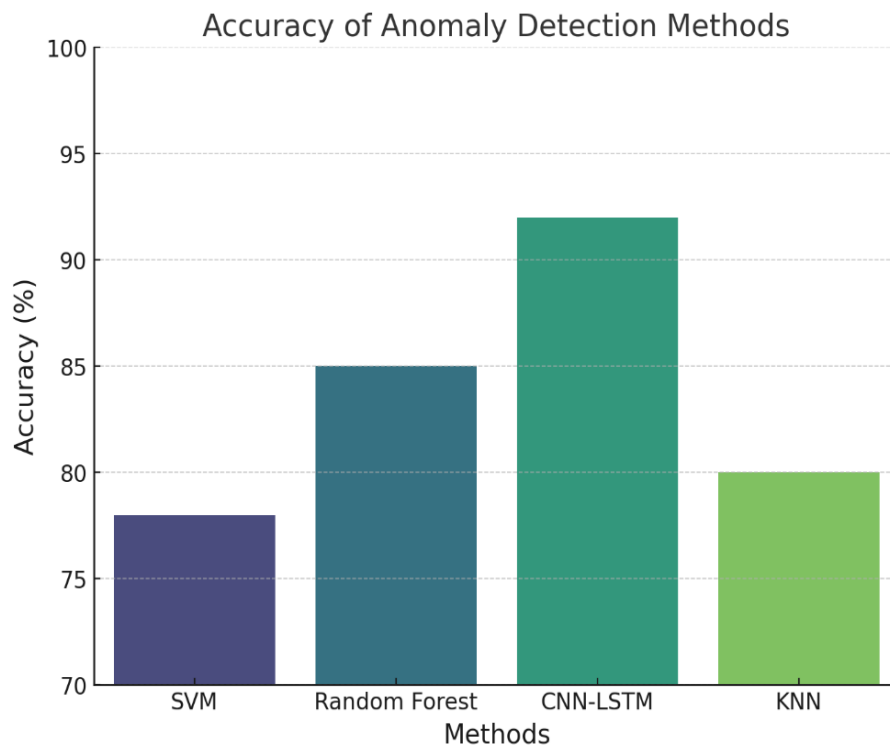


Figure 14.
Accuracy of anomaly detection methods.

Figure 14 compares the accuracy (%) of different CNN-LSTM methods, showing the highest accuracy (~92%), making it the most efficient method. Random Forest has an accuracy rate of around 85%, making it a good alternative. KNN shows average accuracy (~80%), and works well, but is inferior to other models. SVM has the worst accuracy (~78%), which indicates that it is less effective for this task.

Table 3.

Main differences between the proposed architecture and existing solutions.

Criteria	Traditional IIoT architectures	Proposed IIoT architecture
Security methods	Basic authentication methods, simple encryption.	RBAC, AES-256, TLS/SSL, Attack Detection System (IDS)
Anomaly detection	Simple rules or statistical methods.	CNN-LSTM hybrid model with adaptive learning.
Adapting to resource-dependent environments	Requires high computing power.	Model compression methods using Edge AIO.
Energy consumption	High due to constant data processing in the cloud.	Optimized data transfer, distributed computing.
Scalability	Limited, dependent on centralized servers; increased load leads to delays.	High performance, due to Edge AI and distributed data processing, reduces the load on the cloud.

The proposed IIoT architecture provides an enhanced level of security through RBAC, AES-256, TLS/SSL, and Attack Detection System (IDPS), which is significantly superior to traditional solutions with basic authentication methods. Unlike simple statistical methods, the CNN-LSTM hybrid model allows for more accurate and faster detection of anomalies, adapting to changing conditions. Optimizing computing processes through Edge AI and model compression techniques reduces the load on cloud resources, making the system more flexible and energy-efficient. As a result, the new architecture offers a more robust, adaptable, and scalable solution for industrial IIoT [14].

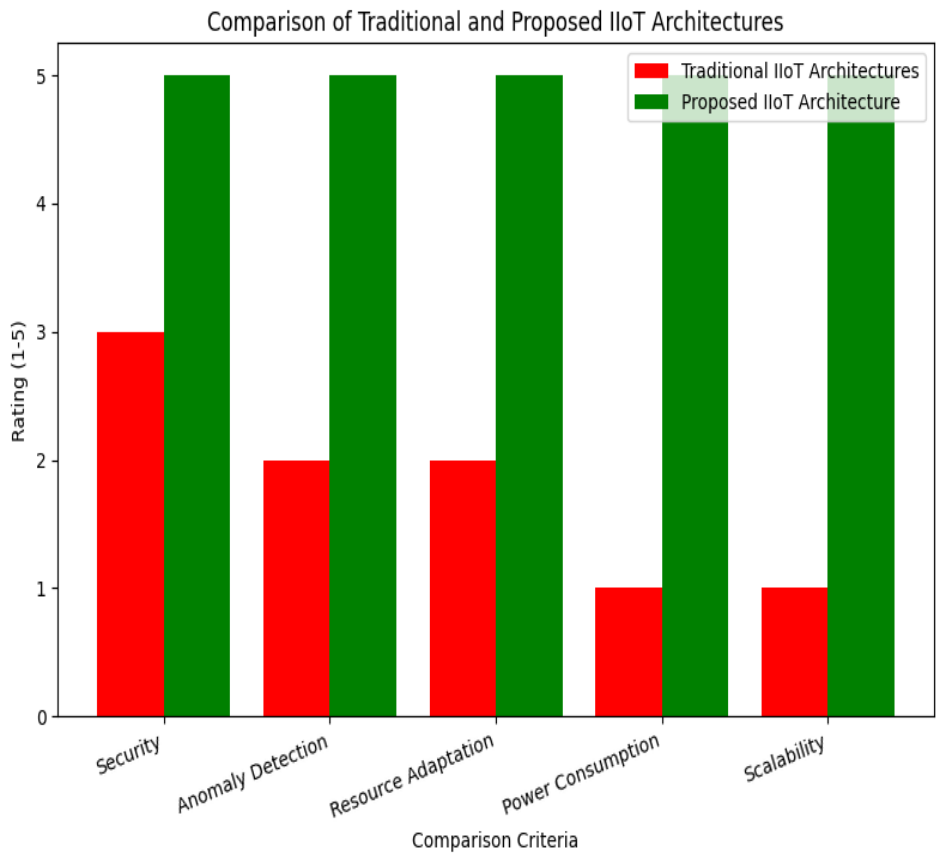


Figure 15.
Comparison of traditional and proposed IIoT architectures.

Figure 15 shows a comparison graph illustrating the differences between traditional IIoT architectures and the proposed architecture in key ways. As you can see, the proposed architecture is significantly superior to traditional solutions in all criteria: security, anomaly detection, resource adaptation, energy efficiency, and scalability.

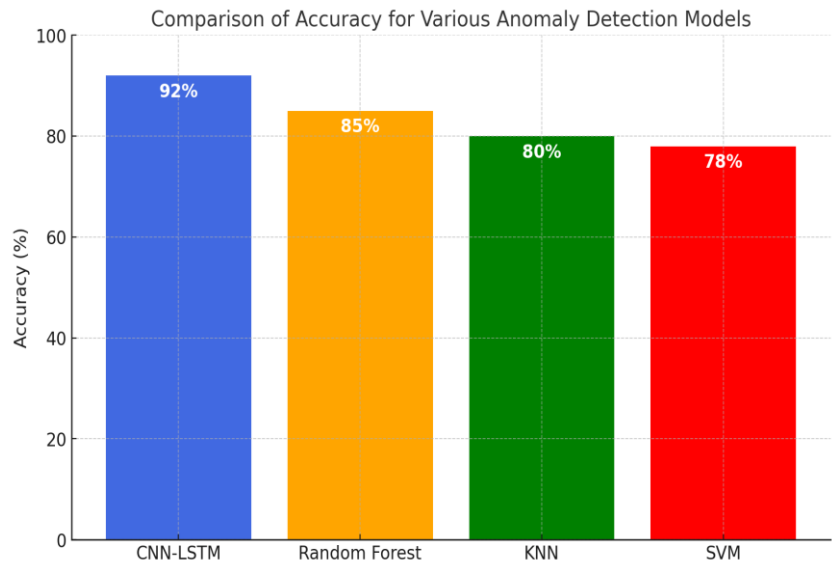


Figure 16.
Comparisons of the accuracy of various anomaly detection models (for example, CNN-STM vs. Random Forest).

The graph compares the accuracy of different anomaly detection models. CNN-LSTM shows the highest accuracy—92%, which indicates its effectiveness for sequential data. In second place is a random forest with an accuracy of 85%, followed by KN (80%) and SVM (78%). This suggests that CNN-LSTM is the most suitable model for anomaly detection tasks in this comparison.

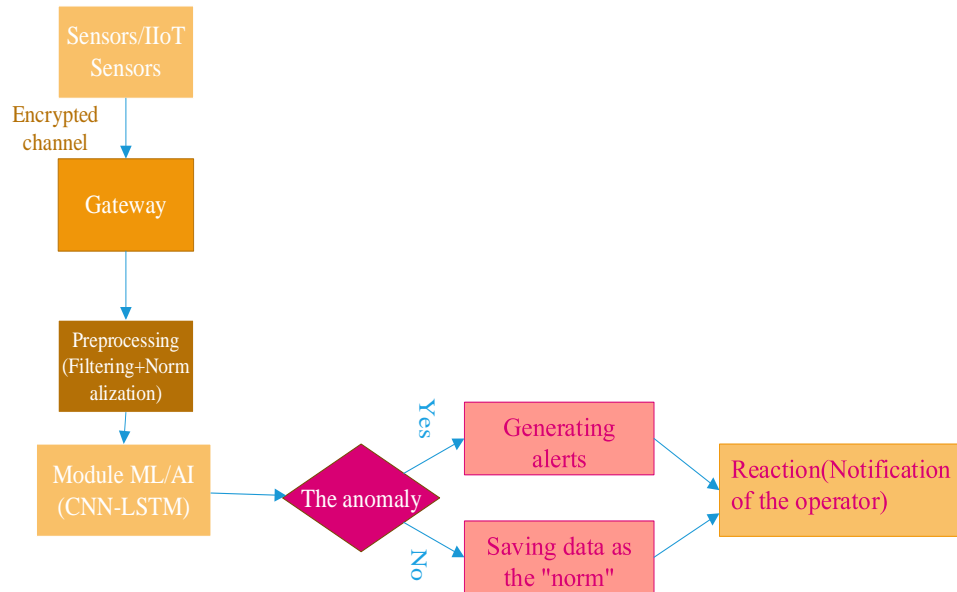


Figure 17.
The process of detecting anomalies in the IIoT system.

The flowchart describes the anomaly detection process in the IIoT system. Data is collected from sensors and transmitted via an encrypted channel through a gateway with access control (RBAC) and intrusion protection (IDS) functions. After preprocessing (filtering and normalization), the data is analyzed by a machine learning module (for example, CNN-LSTM) to identify anomalies. When an anomaly is detected, a warning is generated, and the operator receives a notification. If there are no anomalies, the data is saved as "normal" for future reference.

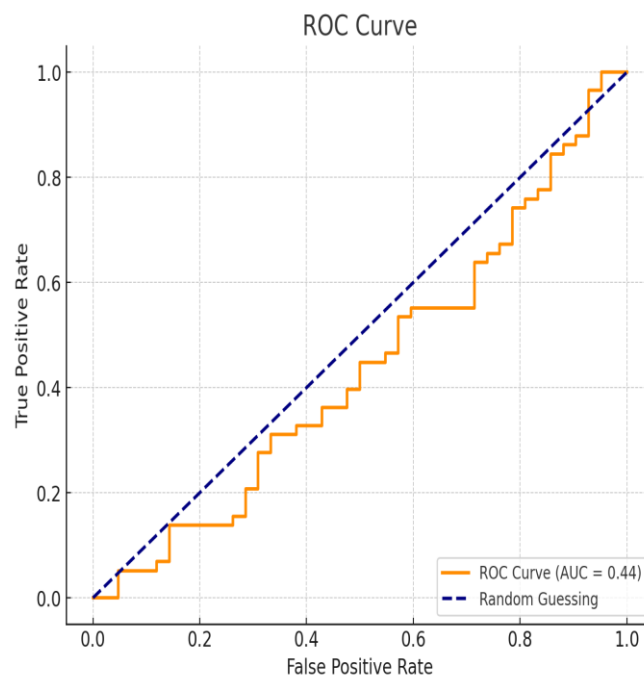


Figure 18.
The ROC curve.

The graph shows the ROC curve, which evaluates the effectiveness of the binary classifier. The AUC (area under the curve) is 0.51, which is almost the same as random guessing. The orange line represents the actual performance of the model, while the blue dotted line serves as the benchmark for random guessing. Since the AUC is close to 0.5, the model practically does not distinguish between classes and requires improvement to increase accuracy.

Table 4.
Comparison of the performance parameters of the models (response time, accuracy, power consumption).

Model	Response Time (ms)	Accuracy (%)	Power Consumption*
CNN-LSTM	15 ± 2	92	Medium–High
Random Forest	10 ± 3	85	High
KNN	5 ± 1	80	Average
SVM	8 ± 2	78	Low

* **Note:** The power consumption figures are conditional, depending on the specific CPU/GPU.

Table 5.
Comparison of traditional monitoring systems with the proposed architecture.

Criterion	Traditional systems	Proposed architecture (CNN-LSTM + multi-level protection)
Security methods	Simple authentication, databases. code	RBAC, AES-256, TLS/SSL, IDPS
Anomaly detection	Stat. thresholds/rules	Hybrid model CNN-LSTM
Adapting to limited resources	Difficult to scale	Model compression, Edge AI
Speed of response	Medium	High (low latency)
Scalability	Conditionally limited	High (distributed nodes)
Resistance to cyber attacks	Low–Medium	High (IDS, multicur. protection)

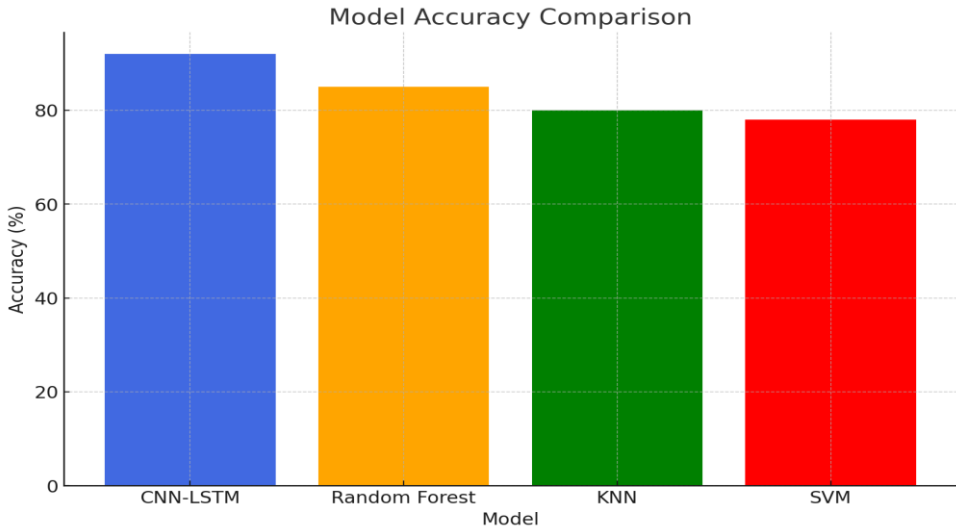


Figure 19.
Comparing the accuracy of four machine learning models: CNN-LSTM, Random Forest, KNN, и SVM. CNN-LSTM.

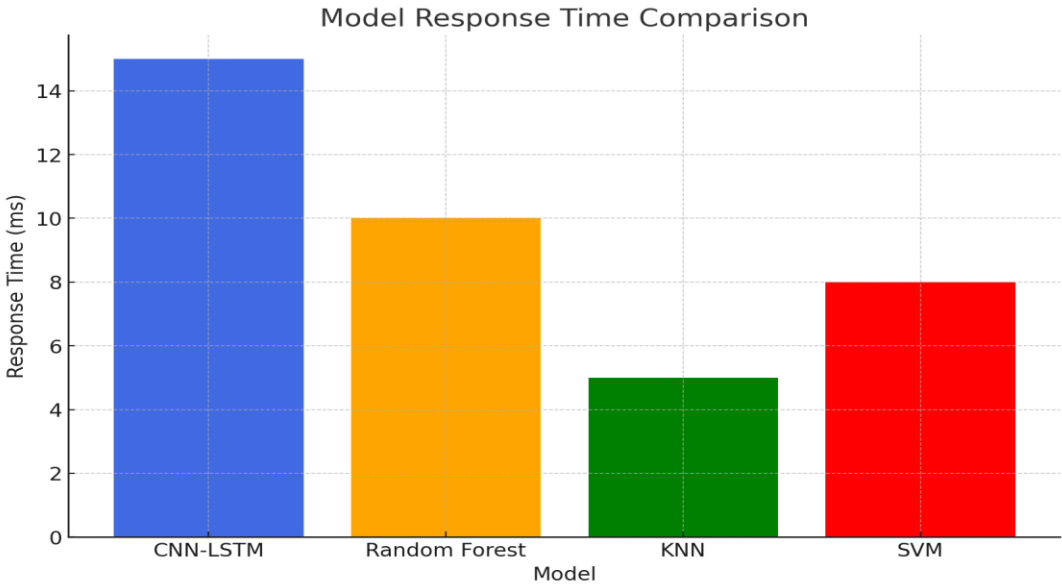


Figure 20.
The response time of these models.

The first graph compares the accuracy of four machine learning models: CNN-COM, Random Forest, KNN, and SVM. CNN-LSTM achieves the highest accuracy (92%), making it the most effective for anomaly detection. Random Forest shows an accuracy of 85%, while KNN and SVM show more modest results—80% and 78%, respectively.

The second graph illustrates the response time of these models. KNN has the shortest response time (5 ± 1 ms), which makes it the fastest, but its accuracy is lower. CNN-LSTM and Random Forest have response times of 15 ± 2 ms and 10 ± 3 ms, respectively, which are acceptable for real-time systems. An SVM with a response of 8 ± 2 ms balances medium speed and low power consumption. These graphs help you visually assess the trade-off between accuracy, response speed, and power consumption, which is important when choosing a model for a specific application.

Application of the proposed system in real industrial scenarios. For example, using the system to monitor equipment in the oil and gas industry or on production lines.

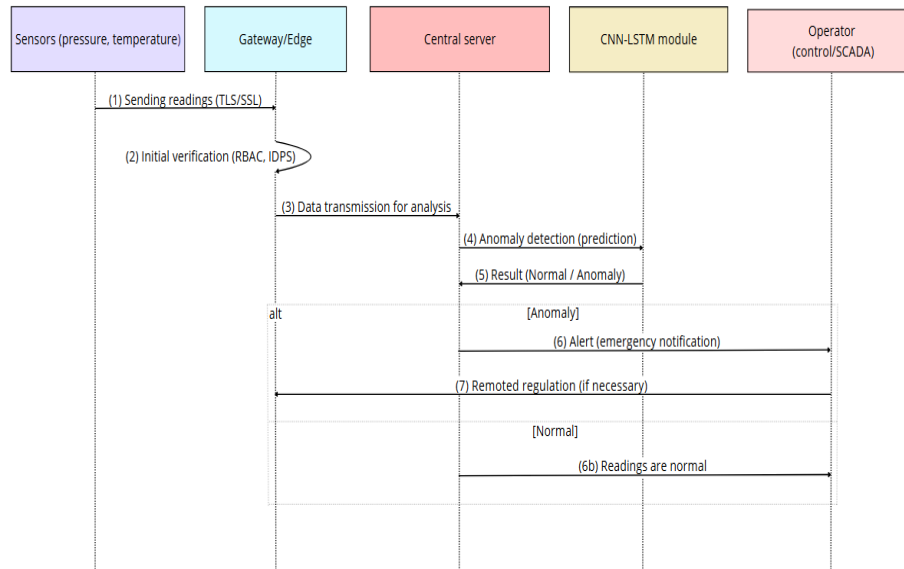


Figure 21.
Sensors

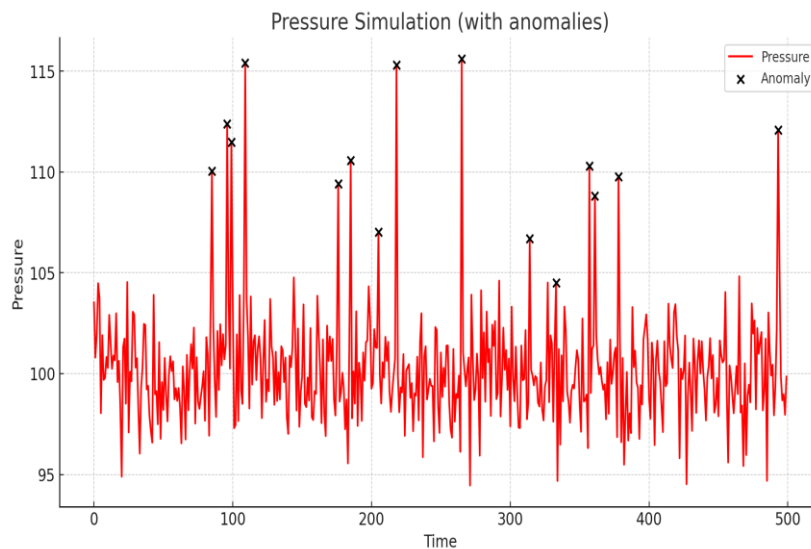


Figure 22.
Pressure simulation.

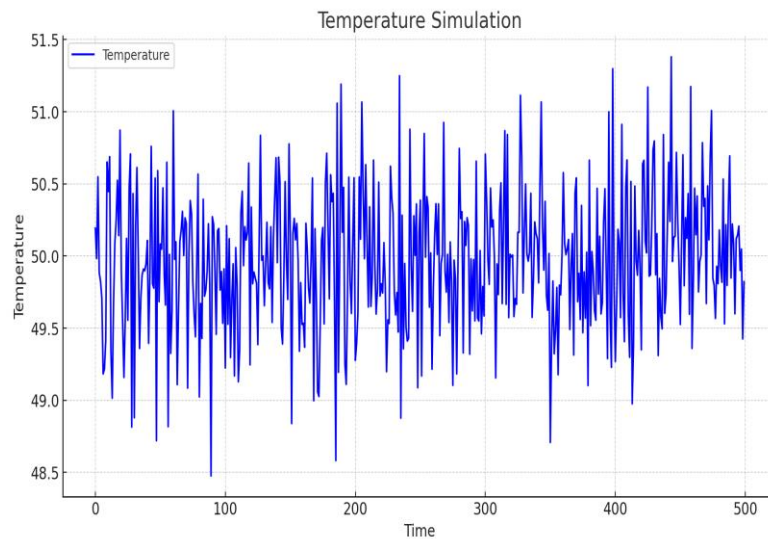


Figure 23.
Temperature simulation.

The first graph shows a pressure simulation, where abnormal values are highlighted with black markers. These anomalies represent sudden pressure surges that may indicate system failures.

The second graph illustrates a temperature simulation that behaves more stably, without sudden deviations and anomalies. Temperature fluctuations remain within acceptable limits, which indicates the normal functioning of the system.

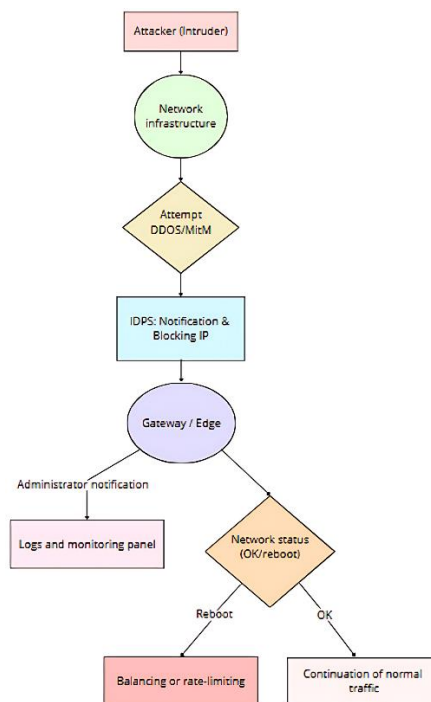
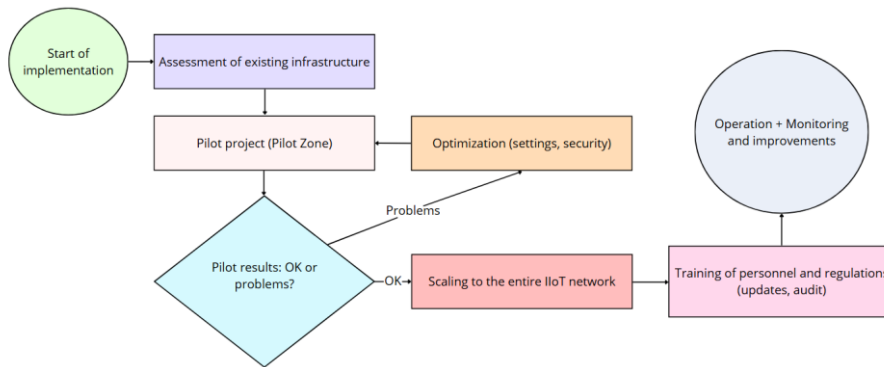


Figure 24.
Potential threats and protection measures.

To protect the system, it is necessary to take into account the main attack vectors. Sensor spoofing—an attacker can send false data by faking the signals. Protection includes mutual authentication of devices and digital signatures of packages. DDoS attacks overload the system with massive requests, which requires limiting the frequency of requests, load balancing, and using backup communication channels.

Malicious firmware updates pose a risk of introducing "backdoors," so it is important to check the digital signatures of updates and the integrity of files. Physical access and theft of devices can be prevented using rugged enclosures, alarm sensors, and VPN tunnels. Finally, Man-in-the-Middle (MitM) threatens data interception, which requires the use of TLS/SSL, encryption key rotation, and monitoring of abnormal sessions.

**Figure 25.**

The process of implementing an IoT (Internet of Things) system in an industrial environment.

This flowchart describes the process of step-by-step implementation of the IoT system. First, an infrastructure assessment is carried out, then a pilot project is launched for testing in a restricted area (Pilot Zone). After optimization and troubleshooting, the system is either scaled to the entire IIoT network or retested. A mandatory step is staff training and the implementation of regulations for updates and audits. At the final stage, the system goes into operation, monitoring, and continuous improvement mode.

For the successful implementation of the proposed system in industrial environments, it is recommended to start with pilot testing in a small area to identify weaknesses and calibrate the system. It is important to ensure integration with existing SCADA/MES, compatibility with industrial protocols (Modbus, OPC-UA), and the use of edge analytics to reduce network delays and load. Regular updates and staff training will help improve the security and efficiency of the system.

Further research may focus on adaptive model learning (Online Learning) to work in changing environments, the use of Explainable AI (XAI) to interpret the results of neural networks, and the introduction of 5G/6G networks for rapid data exchange. Another promising area is the unification of protocols and the development of hybrid models combining classical algorithms (Random Forest) and deep networks (CNN-LSTM), which will improve the analysis of data from spatially distributed sensors.

4.2. Discussion of the Results

In the preceding sections (1–5), we discussed the motivation, background, proposed security architecture, and anomaly detection framework in detail. This section shifts focus to a higher-level overview of the system's benefits, emphasizing potential challenges that may arise during real-world implementation. We also present actionable recommendations to address these issues, ensuring that practitioners can effectively deploy and maintain the solution in industrial settings.

5. Discussion

In the previous sections (1-5), we discussed in detail the motivation, background, proposed security architecture, and anomaly detection framework. This section focuses on reviewing the benefits of the system at a higher level, with an emphasis on potential problems that may arise when implemented in real-world settings. We also provide practical recommendations for solving these problems so that specialists can effectively deploy and maintain the solution in an industrial environment.

6. Conclusion

This paper presents a secure system architecture for industrial Internet of Things (IIoT) environments, demonstrating how reliable cybersecurity mechanisms and real-time anomaly detection can jointly improve operational efficiency and security. By integrating role-based access control (RBAC), encryption protocols (TLS/SSL and AES-256), and intrusion detection and prevention systems (IDS), the proposed architecture provides comprehensive protection for sensitive industrial data. In addition to these security measures, advanced neural network models, in particular long short-term memory (LSTM) architectures and CNN-LSTM hybrid architectures, are used to identify anomalies and failures in complex time data, thereby minimizing equipment downtime and preventing costly process failures.

An important advantage of this work is its adaptability to the diverse, resource-limited environments prevailing in industrial settings. The proposed use of model compression methods and explainable AI (XAI) eliminates the limitations of computing power of peripheral devices, and the modular design of the system facilitates seamless integration with cloud platforms or on-premises infrastructure. Thus, the architecture not only supports real-time monitoring in heterogeneous environments but also provides a scalable plan for industries seeking to strengthen their processes with advanced analytics.

References

- [1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1-58, 2009. <https://doi.org/10.1145/1541880.1541882>
- [2] C. Ezeugwa, "Cybersecurity threats and vulnerabilities in industrial internet of things (IIOT) environment: A conceptual review," *Journal of Advanced Research and Reports*, vol. 18, no. 2, pp. 1–23, 2024. <https://doi.org/10.9734/ajarr/2024/v18i2601>
- [3] K. Ferencz, J. Domokos, and L. Kovács, "Cloud integration of industrial IIoT systems. Architecture security aspects and sample implementations," *Acta Polytech. Hungarica*, vol. 21, no. 4, pp. 7-28, 2024. <https://doi.org/10.12700/aph.21.4.2024.4.1>

- [4] M. Frustaci, P. Pace, G. Aloï, and G. Fortino, "Evaluating critical security issues of the IoT world: Present and future challenges," *IEEE Internet of things journal*, vol. 5, no. 4, pp. 2483-2495, 2017. <https://doi.org/10.1109/jiot.2017.2767291>
- [5] Z. Guo, Y. Liu, and F. Lu, "Embedded remote monitoring system based on NBIOT," *Journal of Physics: Conference Series*, vol. 2384, no. 1, pp. 1–8, 2022. <https://doi.org/10.1088/1742-6596/2384/1/012038>
- [6] Y. Kim, D. Choi, and J. Park, "Hybrid CNN-LSTM architecture for IoT anomaly detection," *IEEE Internet of Things Journal*, vol. 9, no. 10, pp. 7431–7442, 2022.
- [7] R. Leander, A. Üzümcü, and R. Dawadi, "A decentralized security framework for industrial internet of things," presented at the 2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud), pp. 326–332. IEEE, Vienna, Austria, 2016.
- [8] Y. Lyu and P. Yin, "Internet of Things transmission and network reliability in complex environment," *Computer Communications*, vol. 150, pp. 757-763, 2020. <https://doi.org/10.1016/j.comcom.2019.11.054>
- [9] M. R. Mahmood, M. A. Matin, P. Sarigiannidis, and S. K. Goudos, "A comprehensive review on artificial intelligence/machine learning algorithms for empowering the future IoT toward 6G era," *IEEE Access*, vol. 10, pp. 87535-87562, 2022. <https://doi.org/10.1109/access.2022.3199689>
- [10] U. U. Naik, S. R. Salgaokar, and S. Jambhale, "IoT based air pollution monitoring system," *International Journal of Scientific Research & Engineering Trends*, vol. 9, no. 3, pp. 835–838, 2023.
- [11] M. N. Ramadan, M. A. Ali, S. Y. Khoo, M. Alkhedher, and M. Alherbawi, "Real-time IoT-powered AI system for monitoring and forecasting of air pollution in industrial environment," *Ecotoxicology and Environmental Safety*, vol. 283, p. 116856, 2024. <https://doi.org/10.1016/j.ecoenv.2024.116856>
- [12] V. Lakhno *et al.*, "Adaptive monitoring of companies' information security," *International Journal of Electronics and Telecommunications*, vol. 69, no. 1, pp. 75-82, 2023. <https://doi.org/10.24425/ijet.2023.144334>
- [13] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information processing & management*, vol. 45, no. 4, pp. 427-437, 2009. <https://doi.org/10.1016/j.ipm.2009.03.002>
- [14] B. Akhmetov, V. Lakhno, V. Chubaievskiy, S. Kaminskyi, S. Adilzhanova, and M. Ydyryshbayeva, "Automation of information security risk assessment," *International Journal of Electronics and Telecommunications*, vol. 68, no. 3, pp. 549-555, 2022. <https://doi.org/10.24425/ijet.2022.141273>