Multilayer Perceptron-Based Defense Mechanisms for Securing Industrial IoT in Industry 4.0 Environments

Lahcen Idouglid*, Said Tkatek, and Khalid El Fayq
Computer Sciences Research Laboratory, Ibn Tofail University, Kenitra, Morocco
E-mail: lahcen.idouglid@uit.ac.ma, said.tkatek@uit.ac.ma, khalid.elfayq@uit.ac.ma
*Corresponding author

Keywords: industrial internet of things (iiot), intrusion detection systems (ids), multilayer perceptron (mlp), deep learning (dl), industry 4.0

Received: October 24, 2024

In the Industry 4.0 era, the Industrial Internet of Things (IIoT) has transformed manufacturing by facilitating seamless connectivity and real-time data exchange between physical devices and systems. This transformation has bolstered efficiency, productivity, and decision-making in industrial settings. However, the increased connectivity also brings heightened cybersecurity risks. Securing the IIoT environment is critical to safeguard critical infrastructure, data, and operations against cyber threats. As IIoT adoption expands across sectors, ensuring system security and resilience becomes imperative to maintain operational continuity and preserve trust. This paper proposes a deep learning-based approach, leveraging the CIDDS, BOT-IoT, and Edge_IIoTset datasets, to fortify IIoT and manufacturing systems against cyber threats. Multilayer Perceptron (MLP) is identified as the top-performing model, achieving an accuracy of 99.26%, precision of 98.74%, and recall of 98.86% on the CIDDS dataset. Similar superior performance was observed on the BOT-IoT (99.52%, 99.52%, and 99.99%) and Edge_IIoTset (99.93%, 99.93%, and 99.99%) datasets, making MLP a robust solution for anomaly detection in industrial IoT environments.

Povzetek: Članek predstavi MLP kot najučinkovitejši model za zaznavanje vdorov v IIoT, ki na treh realnih podatkovnih zbirkah preseže druge algoritme po točnosti, odzivnosti in robustnosti.

1 Introduction

The Industrial Internet of Things (IIoT) is central to Industry 4.0, driving the transformation of industrial processes through advanced digital technologies. The Industrial Internet of Things (IIoT) has transformed manufacturing by enabling low-latency communication and real-time data exchange between interconnected devices and systems. It encompasses interconnected sensors, devices, and instruments across sectors like manufacturing, energy, and healthcare [1], [2]. IIoT boosts productivity, efficiency, and sustainability while opening new business avenues[1], [3].

However, IIoT's extensive integration also invites significant security risks. Cyberattacks threaten data integrity, confidentiality, and service availability[4]. The dynamic and diverse IIoT landscape complicates traditional security methods reliant on predefined rules[5], [6]. Advanced security solutions are urgently needed to counter evolving cyber threats[7], [8], [9].

The Industrial Internet of Things (IIoT) orchestrates intelligent devices spanning various industrial domains, collecting and processing data. Its architecture, comprising perception, network, and processing layers, faces unique security hurdles, demanding robust Intrusion Detection Systems (IDS) [10], [11].

Embedding Machine Learning (ML) within IDS elevates detection accuracy, flexibility, and scalability. ML enables IDS to learn from data, identify crucial features, and construct models adept at recognizing both known and unknown threats[12], [13]. Furthermore, ML facilitates IDS adaptation to the intricacies of Fog/Edge computing, a burgeoning approach bringing computation closer to IIoT data sources[14], [15], [16].

The figure 1 shows An Exploratory Visualization of IIoT. Here are the key contributions of the paper summarized:

- Proposes a deep learning-based approach to strengthen Industrial IoT (IIoT) and manufacturing systems against cyberattacks.
- Highlights the importance of data-driven solutions for evolving security threats.
- Leverages insights from real-world data by analyzing benchmark datasets like CIDDS, BOT-IoT, and Edge_IIoTset.
- Identifies Multilayer Perceptron (MLP) as the most effective deep learning algorithm for this application.

Figure 1 provides an exploratory visualization of the Industrial Internet of Things (IIoT), capturing key elements of the interconnected infrastructure and their relationship with emerging security challenges.

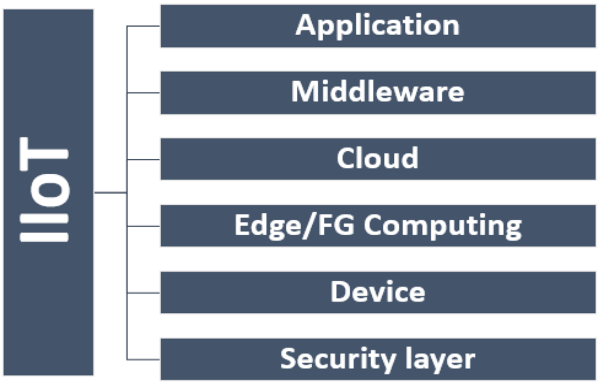

Figure 1: An exploratory visualization of the industrial internet of things (IIoT)

The remainder of this paper is arranged as follows. In Section II, we discuss an in-depth examination of intrusion detection studies in the Industrial Internet of Things (IIoT), focusing on Deep/Machine Learning. Next section explores the innovative architecture proposed in detail. Then, it meticulously presents and compares results, culminating in insightful conclusions. Lastly, the concluding section offers valuable suggestions for future research in the ever-evolving realm of IIoT security.

2 Related works

In this section, we thoroughly review and analyze previous studies that are relevant to our research focus. These studies propose novel methods aimed at improving security in IoT settings, with a particular emphasis on using machine learning techniques to develop intrusion detection systems (IDSs) suited for IoT environments.

A pioneering study [17] introduces the IDS-SIoEL framework, employing Ensemble Learning with AdaBoost and feature selection techniques. Achieving exceptional performance on BoT-IoT, Edge-IIoT, and IoT-23 datasets, it boasts a remarkable 99.9% accuracy, recall, and precision. With swift learning and detection times, this model offers robust security solutions for diverse smart city applications.

The study published in 2022 [18] analyzes the feasibility of implementing a Host-Intrusion Detection System (HIDS) based on Deep Learning (DL-HIDS) across diverse commercial IoT devices. Findings demonstrate significant promise with up to 99.74% accuracy and minimal inference time. It emphasizes the necessity of customizing IDS for individual device classes due to their diverse architectures, providing valuable insights for securing IoT environments.

Another significant contribution comes from a study [1] tackling modern network and Industrial Internet of Things (IIoT) security challenges by designing an advanced Intrusion Detection System (IDS). Leveraging deep learning technologies, the proposed methodology optimizes network configurations, resulting in a robust IIoT anti-intrusion detection system. Demonstrating superior performance, the IDS showcases heightened detection rates, minimal false positives, and robust data correctness, aligning with privacy laws.

The paper [19] propose a deep learning ensemble model using Deep Neural Networks (DNN) and Long Short-Term Memory (LSTM) for cybersecurity in IoT environments. The ensemble approach achieves a 99.34% accuracy in detecting complex cyber-attacks, with low latency, making it suitable for real-time anomaly detection in smart environments.

In response to security challenges in the Industrial Internet of Things (IIoT), a groundbreaking solution is introduced by a paper [20], presenting PK-IDS, a cutting-edge hybrid IDS for Edge-Based IIoT. Seamlessly integrating K-NN and PCA, it achieves remarkable results: 99.10% accuracy, 98.4% detection rate, and 2.7% false alarm rate (NSL-KDD); 98.2% accuracy, 97.6% detection rate, and 2.9% false alarm rate (Bot-IoT). This addresses intricate security challenges in edge-based IIoT environments.

The authors of [21] present a hybrid SDN-based deep learning framework combining Convolutional Neural Networks (CNN) and Bidirectional Long Short-Term Memory (Bi-LSTM) to secure Internet of Medical Things (IoMT) devices. With a detection accuracy of 99.97%, the system offers scalability and efficiency, effectively addressing cyber threats in healthcare IoT environments.

Additionally, addressing security concerns in the Internet of Things (IoT), a paper [22] presents an Access Key Agreement (AKA) scheme, RDAF-IIoT, designed to boost security within IoT, particularly in industrial environments. Validated for resilience against security attacks, the scheme exhibits superiority, showcasing lower computational and communication costs compared to similar security frameworks, along with enhanced security features.

In another study [23], the focus shifts to addressing the growing cybersecurity threats to the Internet of Things (IoT), particularly in Industrial IoT (IIoT) applications. Proposing an effective Intrusion Detection System (IDS), the study employs machine learning algorithms such as K-Nearest Neighbors, Random Forest, and Logistic Regression for comprehensive evaluation on the TON_IoT dataset. The Classification and Regression Tree (CART) algorithm outperforms others, demonstrating the proposed framework's efficiency in mitigating IoT/IIoT intrusion risks.

The article [10] highlights the pivotal role of an Intrusion Detection System (IDS) in safeguarding the Industrial Internet of Things (IIoT), introducing a hybrid IDS architecture incorporating machine learning. The proposed innovations showcase heightened detection accuracy, decreased training time, and enhanced network security, well-suited for the edge scenario of IIoT.

The authors of [24] apply Hyperparameter Optimization for the XGBoost algorithm (HO-XGB) to improve network intrusion detection using the CSE-CIC-IDS 2018 dataset. The optimized model, fine-tuned with parameters like learning rate and max depth, significantly enhances intrusion detection accuracy and efficiency, outperforming traditional methods in real-time network security.

Furthermore, a study [25] addresses escalating security concerns in the IoT environment by proposing a Machine Learning (ML)-enabled Intrusion Detection System (IDS). Focusing on the modified Random Forest (RF) algorithm, the proposed IDS demonstrates effectiveness in safeguarding diverse IoT networks and applications, emphasizing its significance in addressing current security challenges.

This paper [26] conducts a comprehensive survey on the existing state and security challenges within the Internet of Things (IoT). Scrutinizing security principles across the Perception, Network, and Application layers, researchers discuss countermeasures to layer-specific security challenges and introduce future directions, emphasizing the integration of advanced networking protocols to surmount prevailing research challenges in IoT security.

Moreover, this research [27] focuses on anomaly detection in the complex and massive data generated by Industrial IoT (IIoT). Algorithms for machine learning, such as logistic regression, and decision trees are compared for their anomaly detection capabilities using three IIoT benchmark datasets[27].

Additionally, this survey delves into the synergy of AI and ML in Industry 4.0, particularly focusing on fault detection, cyber-security, and human-machine interaction [28]. It emphasizes cloud/fog/edge architectures for efficient data utilization and training and identifies open research issues in these domains[28].

The study [29] on IoT Networks presents an advanced Intrusion Detection System (IDS) using an ensemble of machine learning algorithms, including Logistic Regression (LR), Decision Trees (DT), and Random Forest (RF). Validated on real-world datasets, the IDS achieves high accuracy in detecting attacks, providing a robust solution for securing IoT networks.

Lastly, the paper [30] highlights the effectiveness of machine learning (ML) in crafting Intrusion Detection Systems (IDSs) for IoT security. With empirical analysis illustrating substantial improvements in ML-based IDS accuracy with high-quality data and models, it underscores the pivotal role of data preprocessing and model quality in achieving accurate intrusion detection within IoT environments.

The next table summarizes some related works focusing on different methodologies for enhancing Intrusion Detection Systems (IDS) in IoT and IIoT environments. Each work employs various machine learning and deep learning techniques across different datasets. The results highlight the accuracy achieved by these methods, demonstrating the effectiveness of models such as ensemble learning, hybrid frameworks, and optimized algorithms for improving security in IoT networks.

Table 1: Comparative analysis of related works on intrusion detection systems (IDS) in IoT and IIoT environments

Paper	Year	Method	Dataset	Results (Accuracy)
[17]	2023	Ensemble Learning (AdaBoost, Random Forest)	BoT-IoT, Edge-IIoT, IoT-23	99.9% Accuracy
[23]	2023	Hybrid IDS Framework (Machine Learning)	NSL-KDD, BoT-IoT	99.10% (NSL-KDD), 98.2% (BoT-IoT)
[22]	2023	Access Key Agreement (AKA) Scheme	Simulated IIoT Environment	High computational efficiency and lower communication costs
[25]	2024	Modified Random Forest Algorithm	TON-IoT, UNSW-NB15	High performance
[18]	2022	Optimized Deep Learning (HIDS)	Edge-IIoTset	Accuracy exceeding 99%
[19]	2024	Deep Learning Ensemble (DNN, LSTM)	IoT Environment	99.34% (binary classifier), 98.26% (multiclass classifier)
[24]	2024	Hyperparameter-Optimized XGBoost (HO-XGB)	CSE-CIC-IDS2018	Best performance
[21]	2024	CNN + BiLSTM Hybrid Framework	IoT-Healthcare Dataset	99.97% Accuracy
[25]	2023	ML-Enabled Intrusion Detection System	TON-IoT, UNSW-NB15	99.5% (TON-IoT), 98.7% (UNSW-NB15)

3 Methodology

The methodology section details the structure of our Intrusion Detection System (IDS), comprising three key components. The components of the Intrusion Detection System (IDS) methodology outlined here involve a systematic process:

Our model employs a systematic, three-step approach for detecting anomalies. First, we meticulously preprocess the data, removing noise, outliers, and inconsistencies to ensure data quality. We then carefully select and engineer features that hold the key to distinguishing normal behavior from anomalies. This data preparation stage sets the foundation for effective learning.

Next, we select an appropriate anomaly detection algorithm, considering factors like the specific problem and data characteristics. KNN, Decision Tree, MLP, and SVM are among the algorithms we evaluate. Once chosen, the algorithm trains on a labeled dataset, learning the patterns of normal behavior. Its performance is then rigorously validated on a separate dataset to ensure accurate anomaly detection.

Finally, we deploy the trained model into the real world to identify anomalies in real-time. We closely monitor its performance, analyzing its outputs for potential issues or false positives. This continuous feedback loop empowers us to refine the model iteratively, adjusting parameters or exploring new algorithms as needed. This three-step process, from meticulous data preparation to ongoing refinement, ensures our anomaly detection model delivers reliable results.

Figure 2 illustrates the core components of the Intrusion Detection System (IDS) proposed in this study.

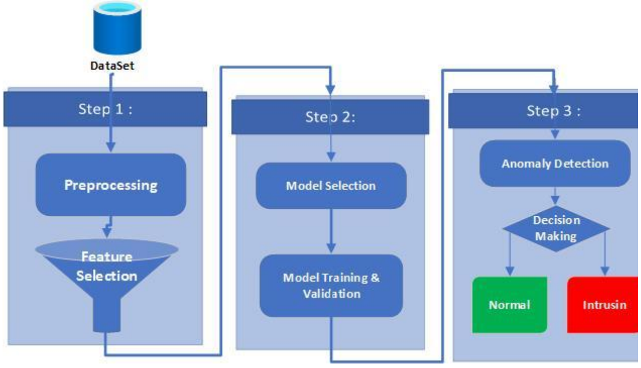

Figure 2: Components of the intrusion detection system (IDS)

The methodology section details the structure of our Intrusion Detection System (IDS), comprising three key components: data preprocessing, algorithm selection, and real-time deployment.

3.1 Data preprocessing

The data preprocessing phase is critical to preparing the datasets for effective model learning and ensuring high-quality input. This phase involved several key steps:

Noise and Outlier Removal: We initially removed noise and irrelevant data points from the datasets to improve overall data quality. Outliers were detected and eliminated using the Isolation Forest algorithm, which is well-suited for identifying anomalies in high-dimensional data.

Handling Missing Data: Missing values were handled using mean imputation for continuous variables and mode imputation for categorical variables. This ensured that no significant information was lost during preprocessing.

Feature Engineering and Selection: To reduce the dimensionality of the datasets and improve model performance, we employed Recursive Feature Elimination (RFE). RFE recursively eliminated less important features by evaluating model performance based on feature subsets, helping us identify the most relevant features. A Random Forest classifier was used to rank the feature importance, and the top 20 features were selected for further model training.

Data Normalization: We applied RobustScaler for normalization, which scaled the features based on interquartile ranges, ensuring that the impact of outliers was minimized and the data was transformed into a format suitable for machine learning algorithms.

Handling Class Imbalance: Given the imbalanced nature of the datasets, especially the BOT-IoT dataset, we applied the Synthetic Minority Over-sampling Technique (SMOTE). SMOTE generated synthetic examples of the minority class, balancing the class distribution and improving the model’s ability to detect underrepresented attack types. This technique enhanced recall rates for anomaly detection and improved the generalization of the model.

3.2 Algorithm selection and training

In the second phase, we selected suitable machine learning algorithms to detect anomalies. The algorithms considered were K-Nearest Neighbors (KNN), Decision Tree (DT), Multilayer Perceptron (MLP), and Support Vector Machine (SVM). Each algorithm was evaluated based on the characteristics of the datasets, with a particular focus on computational efficiency and detection accuracy.

Training Process: The models were trained on labeled datasets using cross-validation (Stratified k-fold cross-validation) to ensure robust performance across different data splits. This technique helped minimize overfitting and provided a more accurate assessment of each model's generalization performance.

Hyperparameter Tuning: For each algorithm, we fine-tuned the hyperparameters using grid search. In the case of MLP, parameters such as the number of hidden layers, activation function, and learning rate were optimized to achieve the best performance.

3.3 Real-time deployment and monitoring

The third phase involved deploying the trained model into real-world industrial environments to identify anomalies in real-time. The model was monitored closely to assess performance in detecting new attacks or unusual behaviors.

Anomaly Detection in Real-Time: The trained model was integrated into the IIoT system to provide real-time anomaly detection. It was designed to alert administrators to potential security threats based on deviations from learned normal behaviors.

Performance Monitoring and Feedback: A continuous feedback loop was established, where the model's outputs were analyzed to identify any issues, such as false positives or missed anomalies. This feedback loop allowed us to iteratively refine the model, adjusting hyperparameters or exploring alternative algorithms as needed.

This comprehensive, three-step methodology—covering meticulous data preprocessing, thorough algorithm selection and training, and ongoing real-time monitoring—ensures a robust and reliable Intrusion Detection System for IIoT environments.

4 Experiments

4.1 Datasets

In evaluating Intrusion Detection Systems (IDSs) with Machine Learning (ML) models, researchers frequently rely on diverse datasets to gauge performance. This study opts for two prominent IoT-based IDS datasets—Bot-IoT and Edge-IIoTset—tailored for IoT applications. These datasets, along with CIDDS, furnish extensive traffic data, enhancing the assessment of IDS models amidst evolving attack landscapes. The table below outlines the key characteristics of these datasets.

Table 2 : Key characteristics of BOT-IOT, EDGE-IIOTSET, AND CIDDS datasets

Feature	Bot-IoT	Edge-IIoTset	CIDDS-001
Purpose	Detect botnet activities and anomalous IoT interactions	Detect cyberattacks in industrial IoT (IIoT) environments	Evaluate anomaly-based intrusion detection systems
Content	Network traffic captures from diverse IoT devices, labeled as normal or botnet activity	Sensor data, Modbus flow information, labeled as normal or attack type	Flow records capturing network traffic features, labeled as normal or attack
Attack Types	Botnet attacks (credential stuffing, data exfiltration, etc.)	DoS/DDoS, information gathering, man-in-the-middle, injection, malware	Various intrusion attempts (e.g., port scans, denial-of-service)
Features	Varies depending on dataset version, typically includes 29 features.	61 extracted features from various sources (logs, traffic, resources, alerts)	53 flow-based features (source/destination IP, ports, packet size, etc.)
Size	73 million inputs	Not explicitly mentioned, described as "large-scale"	1.3M normal flows, 120k attack flows (compressed: 30 GB, uncompressed: 130 GB)

4.1.1 BoT-IoT Datasets:

The Bot-IoT[31] The Bot-IoT dataset, curated by UNSW Canberra Cyber experts, comprises 73 million instances of botnet attacks in IoT networks, demonstrating various attack methods such as data exfiltration, keylogging, and DDoS. Featuring both normal and simulated IoT traffic, it aids researchers and professionals in evaluating and improving IoT network security by validating intrusion detection techniques.

4.1.2 Edge_IIoTset dataset

The Edge_IIoTset Dataset[32] provides meticulously curated sensor data and communication patterns from various industrial environments, specifically tailored for Network-based Intrusion Detection Systems (NIDS) research. With a focus on anomalies and potential intrusions, it offers valuable context for evaluating and improving NIDS effectiveness in securing industrial IoT networks. Researchers and practitioners can utilize its rich contextual information to develop and validate robust security solutions for the ever-evolving landscape of industrial IoT.

4.1.3 CIDDS datasets

The CIDDS [33] (Coburg Intrusion Detection Data Sets) Dataset, created by researchers at Hochschule Coburg, Germany, is tailored for assessing anomaly-based intrusion detection systems. Simulating a small business network environment, it contains both normal and malicious activities, including various attacks like port scans and denial of service. This dataset aids in evaluating intrusion detection methods' effectiveness across diverse network scenarios.

4.2 Algorithm hyperparameters

We evaluated the following machine learning models, and the key hyperparameters for each were as follows:

Multilayer Perceptron (MLP): A 3-layer network with 64, 32, and 16 neurons in the hidden layers, respectively. The activation function used was ReLU, the optimizer was Adam, and the learning rate was set to 0.001. The MLP model was trained for 100 epochs with a batch size of 32.

K-Nearest Neighbors (KNN): The number of neighbors (k) was set to 5, and the distance metric used was Euclidean distance.

Support Vector Machine (SVM): An RBF kernel was employed, with a C parameter of 1.0 and gamma set to 'scale' to adjust automatically based on the dataset.

Decision Tree (DT): The maximum depth of the tree was set to 10, and the criterion used for splitting was Gini impurity.

4.3 Cross-validation technique

To ensure that the models generalized well and to avoid overfitting, we used Stratified k-Fold Cross-Validation with k=10. Stratified cross-validation was employed to ensure that the class distribution (i.e., the proportion of normal and attack samples) remained consistent across each fold. This technique provided a robust evaluation of model performance across different data splits and ensured that the results were reliable.

4.4 Dataset bias and challenges

4.4.1 Class imbalance in BOT-IoT dataset

One of the major challenges associated with the BOT-IoT dataset is class imbalance. The dataset contains a significant number of normal samples compared to attack instances, leading to a situation where the model could become biased toward predicting the majority class (normal traffic) while underperforming in detecting the minority class (attack traffic). In cybersecurity contexts, such biases can result in a higher number of false negatives, where actual attacks go undetected, which can have serious consequences.

4.4.2 SMOTE for addressing class imbalance

To address the issue of class imbalance in the BOT-IoT dataset, we employed the Synthetic Minority Over-sampling Technique (SMOTE). SMOTE works by generating synthetic samples for the minority class (i.e., attack instances), which helps balance the distribution of the dataset. Specifically, SMOTE synthesizes new attack instances by interpolating between existing samples, ensuring that the model is exposed to more representative attack data during training.

This technique helped mitigate the bias toward the majority class (normal traffic) and improved the model's ability to detect attack traffic, as reflected in the improved recall and F1-scores during evaluation.

4.4.3 Other dataset challenges

Edge-IIoTset and CIDDS also presented challenges in terms of diverse feature types, such as network traffic features, sensor data, and Modbus flow information. These datasets required careful preprocessing and feature selection (detailed in the Methodology section) to ensure that the most relevant features were used for training.

While class imbalance was less of an issue in Edge-IIoTset and CIDDS, challenges like varying feature distributions and data sparsity still required careful handling during model training and evaluation.

4.4.4 Performance improvements

The application of SMOTE in balancing the BOT-IoT dataset had a noticeable impact on performance metrics, particularly for models like MLP and SVM. By balancing the class distribution, the recall for attack instances increased, improving the model's overall ability to detect cyberattacks. This improvement is especially critical for intrusion detection systems (IDS), where the cost of false negatives (missed attacks) is high.

4.5 Performance measures

In assessing models, research literature utilizes diverse performance metrics, including precision, recall, accuracy, F1-measure, and Matthew’s correlation coefficient, derived from confusion matrices.

Accuracy: The correctness of the model's predictions is often evaluated through accuracy measures[34].

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \qquad (1)$$

Precision (Positive Predictive Value): Measures the accuracy of positive predictions by evaluating how many of the predicted positive instances are truly positive[35].

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}, \qquad (2)$$

Recall (Sensitivity, True Positive Rate): Indicates the percentage of true positives that the model successfully detected by gauging the model's capacity to collect all pertinent cases[35].

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \qquad (3)$$

F1_Score: Provides a balanced metric between recall and accuracy by representing the harmonic mean of the two[34].

$$\text{F1}_{\text{Score}} = \frac{2\cdot\text{Precision}\cdot\text{Recall}}{\text{Precision+Recall}} \qquad (4)$$

MCC evaluates binary model performance by considering true/false positives/negatives, offering a balanced assessment of classification accuracy [36].

$$\text{MCC} = \frac{\text{TP}\times\text{TN}-\text{FP}\times\text{FN}}{\sqrt{(\text{TP+FP})(\text{TP+FN})(\text{TN+FP})(\text{TN+FN})}} \qquad (5)$$

5 Results

In this study, we analyze the results and findings derived from the two primary benchmark datasets previously mentioned. We assess the performance of the following machine learning algorithms: SVM, MLP, KNN, DT, and XGBoost. Various performance metrics including precision, recall, accuracy, MCC, F1-measure, and training time are utilized in the research literature to gauge the effectiveness of these models.

To evaluate the performance of the proposed model, we use the following metrics:

Accuracy (ACC): The proportion of true results (both true positives and true negatives) among the total number of cases. Accuracy measures the overall correctness of the model's predictions.

Precision: The proportion of true positive predictions among all positive predictions made by the model, representing the model’s ability to avoid false positives.

Recall (Sensitivity): The proportion of true positive cases that are correctly identified by the model, emphasizing the model's capacity to capture actual positive instances.

F1-Score: The harmonic mean of precision and recall, providing a balanced metric when precision and recall are of equal importance.

Matthews Correlation Coefficient (MCC): A more comprehensive metric that takes into account true and false positives and negatives, providing a balanced measure even in cases of class imbalance.

5.1.1 Revisiting results: insights from the CIDDS dataset

The following table (3) presents the performance metrics of various machine learning algorithms on the CIDDS dataset. Metrics include accuracy (ACC%), precision, F1-score, recall, Matthews correlation coefficient (MCC), and training time in seconds. SVM achieves an accuracy of 93.77%, followed by K-NN with 96.66%, MLP with

99.26%, and DT with 98.00%. Precision, recall, and F1-score indicate high model performance across all algorithms, with values close to 1. Matthew's correlation coefficient values range from 0.9122 to 0.9931, suggesting strong correlations between predicted and actual values. Training time varies among the algorithms, with MLP being the fastest at 10.23 seconds.

The Fig. 3 is a graphical metrics representation on the CIDDS dataset showcases the performance of machine learning algorithms. Accuracy varies from 93.77% for SVM to 99.26% for MLP, with MLP leading in precision at 98.74%. F1-score demonstrates consistent high performance, particularly with MLP achieving 98.80%. Recall rates are generally high, ranging from 93.1% for SVM to 98.86% for MLP. Matthew's correlation coefficient (MCC) values, ranging from 91.22% to 99.31%, indicate strong correlations between predicted and actual values.

Table 3: Performance metrics of machine learning algorithms on CIDDS dataset

ML algorithm	**ACC %**	**Precision%**	**F1-score %**	**Recall %**	**MCC %**	**Training Time (s)**
SVM	,9377	0,901	0,941	0,931	0,9622	75,94
K-NN	,9666	0,9263	0,9342	0,943	0,9626	12,56
DT	,98	0,979	0,979	0,979	0,9122	27,5
MLP	,9926	0,9874	0,988	0,9886	0,9931	10,23

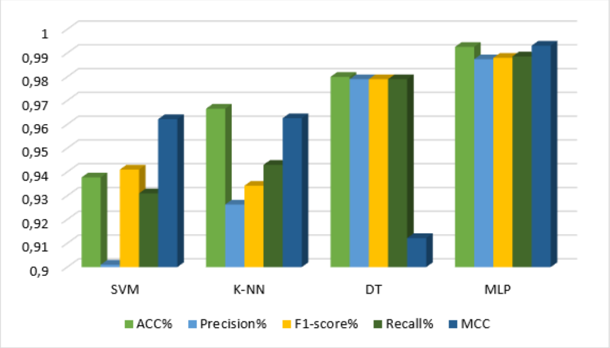

Figure 3: Performance metrics comparison of machine learning algorithms on CIDDS dataset

5.1.2 Revisiting Results: Insights from the BOT-IoT Dataset

The table (1) displays the performance metrics results of various machine learning algorithms on the BOT-IoT dataset. Metrics include accuracy (ACC %), precision, F1-score, recall, Matthew's correlation coefficient (MCC %), and training time in seconds. SVM achieves an accuracy of 98.15%, followed by K-NN with 99.13%, MLP with 99.52%, and DT with 93.4%. Precision, recall, and F1-score indicate high model performance across all algorithms, with values close to 1. Matthew's correlation coefficient values range from 0.9122 to 0.97, suggesting strong correlations between predicted and actual values. Training time varies among the algorithms, with SVM being the fastest at 34.78 seconds.

Table 2: The BOT-IoT dataset's performance metrics results

ML algorithm	**ACC %**	**Precision %**	**F1-score%**	**Recall %**	**MCC %**	**Training Time (s)**
SVM	0,9815	0,9915	0,9915	0,995	0,92	34,78
K-NN	0,9913	0,9933	0,9933	0,999	0,92	56,43
DT	0,934	0,953	0,934	0,963	0,9122	27,3
MLP	0,9952	0,9952	0,9952	0,9999	0,97	48,34

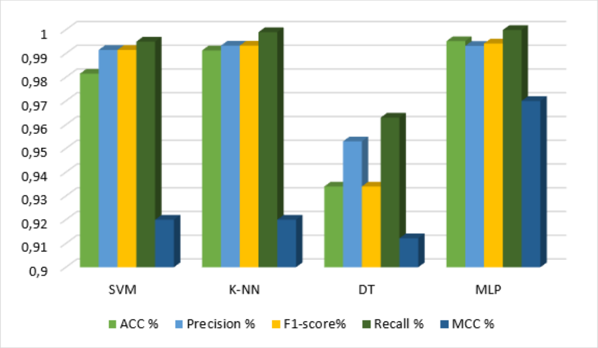

Figure 4: Performance metrics comparison of machine learning algorithms on BOT-IoT dataset

The graphical representation of performance metrics on the BOT-IoT dataset depicts key evaluation measures for machine learning algorithms. Accuracy ranges from 93.4% for DT to 99.52% for MLP, with MLP achieving the highest precision at 99.52%. F1-score shows consistent high performance across all algorithms, particularly with MLP achieving 99.52%. Recall rates are generally high, ranging from 96.3% for DT to 99.9% for MLP. Matthews correlation coefficient (MCC) values, ranging from 91.22% to 97%, highlight strong correlations between predicted and actual values.

5.1.3 Revisiting results: insights from the Edge_IIoTset dataset

This next table presents the performance metrics results of various machine learning algorithms on the Edge_IIoTset dataset. Metrics include accuracy (ACC %), precision, F1-score, recall, Matthew's correlation coefficient (MCC %), and training time in seconds. SVM achieves an accuracy of 99.4%, followed by K-NN with 99.1%, MLP with 99.93%, and DT with 94.9%. Precision, recall, and F1-score indicate high model performance across all algorithms, with values close to 1. Matthew's correlation coefficient values range from 0.931 to 0.992, suggesting strong correlations between predicted and actual values. Training time varies among the algorithms, with MLP being the fastest at 11 seconds.

Table 3: The Edge_IIoTset dataset's performance metrics results

ML algorithm	**ACC %**	**Precision %**	**F1-score %**	**Recall %**	**MCC %**	**Training Time (s)**
SVM	0,994	0,984	0,999	0,9996	0,96	58,42
K-NN	0,991	0,991	0,997	0,9999	0,961	13,61
DT	0,949	0,941	0,989	0,966	0,931	137,2
MLP	0,9993	0,9993	0,999	0,9999	0,992	11

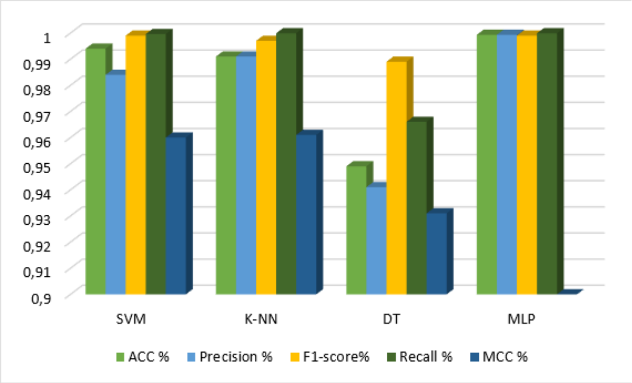

Figure 5: Performance metrics comparison of machine learning algorithms on Edge_IIoTset dataset

The graphical representation illustrates the performance metrics of machine learning algorithms on the Edge_IIoTset dataset. Accuracy ranges from 94.9% for DT to 99.93% for MLP, with MLP achieving the highest precision at 99.93%. F1-score demonstrates consistent high performance across all algorithms, particularly with MLP achieving 99.9%. Recall rates are generally high, ranging from 96.6% for DT to 99.99% for MLP. Matthew’s correlation coefficient (MCC) values, ranging from 93.1% to 99.2%, indicate strong correlations between predicted and actual values.

Figure 6 illustrates the performance metrics of several machine learning algorithms, evaluated across three datasets: CIDDS, BOT-IoT, and Edge_IIoTset. The subfigures (a) to (d) compare metrics such as accuracy, precision, recall, and the Matthews Correlation Coefficient (MCC) among algorithms like SVM and MLP. Notably, MLP consistently demonstrates superior precision, recall, and MCC across all datasets, showcasing its effectiveness in intrusion detection tasks. SVM, on the other hand, achieves commendable accuracy, particularly excelling in classifying instances correctly. This figure highlights the comparative strengths of each algorithm in detecting and managing cyber threats in IIoT environments.

Fig. 6. (a) Accuracy (ACC %): Across all datasets, The SVM model demonstrates strong accuracy across datasets, achieving 93.77% on CIDDS, 98.15% on BOT-IoT, and 99.4% on Edge_IIoTset. It demonstrates the ability of SVM to effectively classify instances into correct categories.

Fig. 6. (b) Precision (%) measured for each dataset: MLP exhibits superior precision across datasets, with values of 98.74% for CIDDS, 99.52% for BOT-IoT, and 99.93% for Edge_IIoTset. This indicates MLP's capability to provide precise positive predictions.

Fig. 6. (c) Recall (%) across datasets: MLP demonstrates outstanding recall rates across all datasets, with values of 98.86% for CIDDS, 99.99% for BOT-IoT, and 99.96% for Edge_IIoTset. It highlights MLP's effectiveness in capturing true positive instances.

Fig. 6. (d) MCC (%) demonstrating balanced classification performance: MLP achieves the highest MCC values for all datasets, with values of 99.31% for CIDDS, 97% for BOT-IoT, and 99.2% for Edge_IIoTset. MCC considers true and false positives and negatives, providing a balanced measure of classification performance.

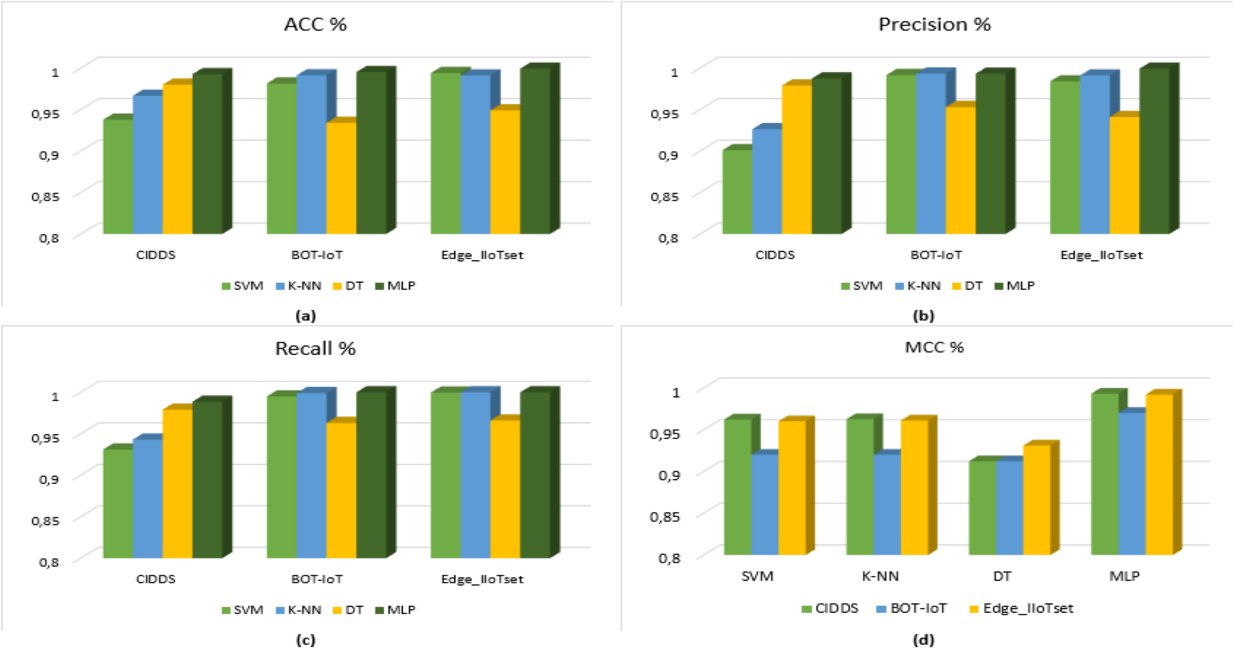

Figure 6: The performance metrics of machine learning algorithms on three distinct datasets: CIDDS, BOT-oT, and Edge_IIoTset

Comparing the performance metrics across the CIDDS, BOT-IoT, and Edge_IIoTset datasets, it becomes evident that MLP consistently outperforms other machine learning algorithms. MLP exhibits superior accuracy, precision, and recall rates across all datasets, indicating its effectiveness in correctly classifying instances and minimizing false positives. Additionally, MLP achieves high F1-scores and Matthew's correlation coefficient values, highlighting its ability to strike a balance between precision and recall while maintaining strong correlations between predicted and actual values. Moreover, MLP demonstrates computational efficiency by requiring relatively shorter training times compared to other algorithms. Thus, MLP emerges as the best model choice for handling diverse and complex datasets, making it suitable for tasks requiring high precision and recall rates in industrial IoT environments.

6 Discussion

The results show that the MLP model consistently outperforms other models, including SVM, KNN, and DT, in terms of accuracy, precision, and recall across all datasets. Specifically, MLP achieved a 99.26% accuracy on the CIDDS dataset, while KNN and DT only reached 96.66% and 98%, respectively.

In comparison to ensemble methods like Random Forest or AdaBoost, MLP continues to show superiority in terms of precision and recall. However, ensemble methods may offer benefits in terms of reducing variance and improving robustness, which could be valuable in certain IIoT scenarios where variability in data is high and frequent model retraining may be required.

The superior performance of MLP can be attributed to its ability to capture complex non-linear patterns in the data through its multilayer architecture. This ability is particularly crucial in detecting multi-stage and sophisticated cyberattacks in IIoT environments, where other models like SVM and KNN might struggle with high-dimensional, noisy, or imbalanced data. The use of deep learning helps in identifying subtle anomalies that could otherwise be missed by traditional models.

MLP also excels in terms of computational efficiency due to its parallel processing capabilities, making it well-suited for real-time anomaly detection. However, it is important to note that the training time for MLP was higher compared to KNN and SVM, which might present challenges in resource-constrained IIoT environments where real-time responsiveness is critical. A hybrid approach that combines MLP for offline analysis with lighter models for real-time intrusion detection could offer a balanced solution to address this challenge.

Despite its advantages, further evaluation of the MLP model in real-world IIoT environments is needed to assess its scalability and performance in dynamic industrial settings. Potential issues like false negatives, which are particularly costly in cybersecurity, should be addressed by optimizing decision thresholds or exploring ensemble learning approaches to reduce the chances of undetected intrusions. Additionally, real-time deployment may face challenges due to latency and computational overhead, which future work could mitigate by optimizing MLP's architecture or incorporating federated learning approaches to distribute the computational load.

While the presented performance metrics, such as accuracy and precision, demonstrate strong overall performance, it is crucial to consider the impact of false negatives in IIoT environments. False negatives occur when an attack is not detected, which can be particularly costly in industrial settings, leading to potential disruptions, operational failures, or even severe security breaches.

To mitigate the risk of false negatives, we propose several strategies:

Adjusting Decision Thresholds: Fine-tuning the decision thresholds of the classifier can lower the rate of false negatives by prioritizing sensitivity (recall) over precision in critical security scenarios.

Ensemble Methods: Leveraging ensemble techniques like Random Forest or boosting algorithms (e.g., AdaBoost, XGBoost) can help improve the detection capabilities by combining the strengths of multiple weak classifiers, leading to a more sensitive system.

Cost-Sensitive Learning: Implementing cost-sensitive learning models can assign a higher penalty to false negatives, ensuring that the model focuses on minimizing missed detections, particularly in cybersecurity contexts where the cost of undetected attacks is high.

6.1 Real-world applicability and deployment

While the MLP model demonstrates strong performance metrics in experimental settings, deploying it in real-world IIoT environments poses several practical challenges:

6.1.1 Scalability

IIoT environments typically involve vast networks of connected devices that generate large amounts of real-time data. Scaling an MLP-based intrusion detection system to handle this volume of data is non-trivial. In practice, distributed computing or edge computing architectures can be used to process data closer to its source, reducing the load on centralized servers. This approach allows for more scalable and efficient detection of anomalies across large-scale IIoT networks.

6.1.2 Latency

In real-time industrial environments, low latency is critical to detecting and responding to cyberattacks before they impact operations. Although MLP provides high accuracy, its inference time can be slower compared to simpler models like Decision Trees or KNN. To ensure real-time detection, optimization strategies such as reducing the number of layers or neurons in the MLP, or employing lightweight models for initial anomaly detection, can help lower the latency without sacrificing detection accuracy.

6.1.3 Computational overhead

MLP models are computationally intensive, especially when applied to large datasets or in environments with limited resources, such as edge or fog computing. To address this, techniques like model pruning, quantization, or offloading computations to specialized hardware (e.g., GPUs or FPGAs) can be used to reduce computational overhead while maintaining model performance. Additionally, deploying the MLP model in a distributed manner across multiple edge nodes can balance the computational demands.

6.2 Hypothetical case study: MLP deployment in a smart factory

To illustrate the practical integration of the MLP model, consider a smart manufacturing facility where various machines, sensors, and devices are connected through an IIoT network to monitor production processes in real-time.

6.2.1 Edge-based deployment

The MLP model could be deployed at the edge of the network, near the data sources (e.g., sensors and industrial machines), to detect anomalies in network traffic in real time. Each edge node would process a subset of the network traffic, allowing for faster detection and reduced network congestion. The system would flag suspicious behavior, such as unauthorized access or unusual data flows, and alert administrators before any significant damage occurs.

6.2.2 Scalability considerations

In large industrial environments with thousands of connected devices, a distributed MLP deployment would be necessary. Each edge node would independently run an instance of the MLP model, analyzing traffic locally and sending detection results to a central server for aggregation and final decision-making. This distributed approach ensures that the IDS can scale efficiently across the entire facility while maintaining high detection accuracy.

6.2.3 Real-time monitoring

The MLP model would continuously monitor network traffic, allowing it to identify deviations from expected patterns of behavior. In the event of a detected anomaly, the system could automatically trigger predefined responses, such as isolating compromised devices or blocking network traffic from suspicious sources. This proactive approach minimizes downtime and ensures uninterrupted production.

6.3 Conclusion on real-world integration

To deploy the MLP model in real-world IIoT environments, challenges related to scalability, latency, and computational overhead must be addressed. Solutions like edge computing, distributed architectures, and hardware acceleration can enable the effective deployment of MLP-based intrusion detection systems in industrial settings. Future work should focus on optimizing MLP models for real-time environments, ensuring they can scale and respond to evolving cyber threats in large, dynamic IIoT networks.

7 Conclusion

In conclusion, this study emphasizes the critical need to secure Industrial Internet of Things (IIoT) environments in the context of Industry 4.0. While IIoT facilitates unprecedented connectivity and efficiency in manufacturing, it also introduces significant cybersecurity challenges. Leveraging deep learning techniques, particularly the Multilayer Perceptron (MLP), presents a promising avenue for fortifying IIoT systems against cyber threats. Through extensive analyses on benchmark datasets, MLP consistently demonstrates superior performance metrics, including accuracy, precision, recall, and efficiency. Its ability to effectively classify instances and maintain strong correlations between predicted and actual values makes it an ideal choice for anomaly detection in industrial IoT environments. Moving forward, future research can explore the integration of MLP-based models into IIoT infrastructures to enhance security and optimize manufacturing processes further. Additionally, ongoing efforts in developing advanced deep learning algorithms tailored for specific IIoT applications can contribute to bolstering cybersecurity and resilience in the industry 4.0 landscape.

References

[1] A. Arun Kumar and R. Krishna Karne, “IIoT-IDS Network using Inception CNN Model,” *JTCSST*, vol. 4, no. 3, pp. 126–138, Aug. 2022, doi: 10.36548/jtcsst.2022.3.002.

[2] J. Du, K. Yang, Y. Hu, and L. Jiang, “NIDS-CNNLSTM: Network Intrusion Detection Classification Model Based on Deep Learning,” *IEEE Access*, vol. 11, pp. 24808–24821, 2023, doi: 10.1109/ACCESS.2023.3254915.

[3] G. Czeczot, I. Rojek, D. Mikołajewski, and B. Sangho, “AI in IIoT Management of Cybersecurity for Industry 4.0 and Industry 5.0 Purposes,” *Electronics*, vol. 12, no. 18, p. 3800, Sep. 2023, doi: 10.3390/electronics12183800.

[4] M. T. Shakir and B. Shannaq, “Enhancing Security through Multi-Factor User Behavior Identification: Moving Beyond the Use of the Longest Common Subsequence (LCS),” *IJCAI*, vol. 48, no. 19, Nov. 2024, doi: 10.31449/inf.v48i19.6270.

[5] L. Idouglid, S. Tkatek, K. Elfayq, and A. Guezzaz, “A NOVEL ANOMALY DETECTION MODEL FOR THE INDUSTRIAL INTERNET OF THINGS USING MACHINE LEARNING TECHNIQUES,” no. 1, 2024, doi: doi: 10.32620/reks.2024.1.12.

[6] Z. Zhang, “SD-WSN Network Security Detection Methods for Online Network Education,” *IJCAI*, vol. 48, no. 21, Nov. 2024, doi: 10.31449/inf.v48i21.6257.

[7] M. Al-Ambusaidi, Z. Yinjun, Y. Muhammad, and A. Yahya, “ML-IDS: an efficient ML-enabled intrusion detection system for securing IoT networks and applications,” *Soft Comput*, Dec. 2023, doi: 10.1007/s00500-023-09452-7.

[8] I. Farhadian Dehkordi, K. Manochehri, and V. Aghazarian, “Internet of Things (IoT) Intrusion Detection by Machine Learning (ML): A Review,” *APJITM*, vol. 12, no. 01, pp. 13–38, Jun. 2023, doi: 10.17576/apjitm-2023-1201-02.

[9] L. Idouglid, S. Tkatek, K. Elfayq, and A. Guezzaz, “Next-gen security in IIoT: integrating intrusion detection systems with machine learning for industry 4.0 resilience,” *IJECE*, vol. 14, no. 3, p. 3512, Jun. 2024, doi: 10.11591/ijece.v14i3.pp3512-3521.

[10] H. Yao, P. Gao, P. Zhang, J. Wang, C. Jiang, and L. Lu, “Hybrid Intrusion Detection System for Edge-Based IIoT Relying on Machine-Learning-Aided Detection,” *IEEE Network*, vol. 33, no. 5, pp. 75–81, Sep. 2019, doi: 10.1109/MNET.001.1800479.

[11] S. Haque, F. El-Moussa, N. Komninos, and R. Muttukrishnan, “A Systematic Review of Data-Driven Attack Detection Trends in IoT,” *Sensors*, vol. 23, no. 16, p. 7191, Aug. 2023, doi: 10.3390/s23167191.

[12] Q. Huang, H. Xian, L. Mei, X. Cheng, N. Li, and N. Li, “Intelligent Distribution Network Operation and Anomaly Detection Based on Information Technology,” *IJCAI*, vol. 49, no. 9, Feb. 2025, doi: 10.31449/inf.v49i9.5584.

[13] M. Alshar’e, K. Abuhmaidan, F. Y. H. Ahmed, A. Abualkishik, M. Al-Bahri, and J. H. Yousif, “Assessing Blockchain’s Role in Healthcare Security: A Comprehensive Review,” *IJCAI*, vol. 48, no. 22, Dec. 2024, doi: 10.31449/inf.v48i22.6155.

[14] R. Alghamdi and M. Bellaiche, “An ensemble deep learning based IDS for IoT using Lambda architecture,” *Cybersecurity*, vol. 6, no. 1, p. 5, Mar. 2023, doi: 10.1186/s42400-022-00133-w.

[15] S. Latif *et al.*, “Deep Learning for the Industrial Internet of Things (IIoT): A Comprehensive Survey of Techniques, Implementation Frameworks, Potential Applications, and Future Directions,” *Sensors*, vol. 21, no. 22, p. 7518, Nov. 2021, doi: 10.3390/s21227518.

[16] H. He, “Automatic Network Traffic Scheduling Algorithm Based on Deep Reinforcement Learning,” *IJCAI*, vol. 48, no. 22, Dec. 2024, doi: 10.31449/inf.v48i22.6943.

[17] C. Hazman, A. Guezzaz, S. Benkirane, and M. Azrour, “lIDS-SIoEL: intrusion detection framework for IoT-based smart environments security using ensemble learning,” *Cluster Comput*, vol. 26, no. 6, pp. 4069–4083, Dec. 2023, doi: 10.1007/s10586-022-03810-0.

[18] I. Idrissi, M. Azizi, and O. Moussaoui, “A Lightweight Optimized Deep Learning-based Host-Intrusion Detection System Deployed on the Edge for IoT,” *IJCDS*, vol. 11, no. 1, pp. 209–216, Jan. 2022, doi: 10.12785/ijcds/110117.

[19] D. Venkataraya Premalatha and S. Ramanujam, “Securing the internet of things frontier: a deep learning ensemble for cyber-attack detection in smart environments,” *IJ-AI*, vol. 13, no. 4, p. 4736, Dec. 2024, doi: 10.11591/ijai.v13.i4.pp4736-4746.

[20] A. Guezzaz, M. Azrour, S. Benkirane, M. Mohy-Eddine, H. Attou, and M. Douiba, “A Lightweight Hybrid Intrusion Detection Framework using Machine Learning for Edge-Based IIoT Security,” *IAJIT*, vol. 19, no. 5, 2022, doi: 10.34028/iajit/19/5/14.

[21] Y. Rbah *et al.*, “Hybrid software defined network-based deep learning framework for enhancing internet of medical things cybersecurity,” *IJ-AI*, vol. 13, no. 3, p. 3599, Sep. 2024, doi: 10.11591/ijai.v13.i3.pp3599-3610.

[22] H. Alasmary, “RDAF-IIoT: Reliable Device-Access Framework for the Industrial Internet of Things,” *Mathematics*, vol. 11, no. 12, p. 2710, Jun. 2023, doi: 10.3390/math11122710.

[23] S. Alshathri, A. El-Sayed, W. El-Shafai, and E. El-Din Hemdan, “An Efficient Intrusion Detection Framework for Industrial Internet of Things Security,” *Computer Systems Science and Engineering*, vol. 46, no. 1, pp. 819–834, 2023, doi: 10.32604/csse.2023.034095.

[24] W. Chimphlee and S. Chimphlee, “Hyperparameters optimization XGBoost for network intrusion detection using CSE-CIC-IDS 2018 dataset,” *IJ-AI*, vol. 13, no. 1, p. 817, Mar. 2024, doi: 10.11591/ijai.v13.i1.pp817-826.

[25] M. Al-Ambusaidi, Z. Yinjun, Y. Muhammad, and A. Yahya, “ML-IDS: an efficient ML-enabled intrusion detection system for securing IoT networks and applications,” *Soft Comput*, Dec. 2023, doi: 10.1007/s00500-023-09452-7.

[26] R. Mahmoud, T. Yousuf, F. Aloul, and I. Zualkernan, “Internet of things (IoT) security: Current status, challenges and prospective measures,” in *2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*, London, United Kingdom: IEEE, Dec. 2015, pp. 336–341. doi: 10.1109/ICITST.2015.7412116.

[27] Bhupal Naik D. S., V. Dondeti, and S. Balakrishna, “Comparative Analysis of Machine Learning-Based Algorithms for Detection of Anomalies in IIoT:,” *International Journal of Information Retrieval Research*, vol. 12, no. 1, pp. 1–55, May 2022, doi: 10.4018/IJIRR.298647.

[28] A. Angelopoulos *et al.*, “Tackling Faults in the Industry 4.0 Era—A Survey of Machine-Learning Solutions and Key Aspects,” *Sensors*, vol. 20, no. 1, p. 109, Dec. 2019, doi: 10.3390/s20010109.

[29] I. T. AL-Halboosi, B. Mohamed Elbagoury, S. Amin El-Regaily, and E.-S. M. El-Horbaty, “Federated inception-multi-head attention models for cyber-attacks detection,” *IJ-AI*, vol. 13, no. 4, p. 4778, Dec. 2024, doi: 10.11591/ijai.v13.i4.pp4778-4794.

[30] S. Alkadi, S. Al-Ahmadi, and M. M. Ben Ismail, “Toward Improved Machine Learning-Based Intrusion Detection for Internet of Things Traffic,” *Computers*, vol. 12, no. 8, p. 148, Jul. 2023, doi: 10.3390/computers12080148.

[31] X. Zhang, O. Upton, N. L. Beebe, and K.-K. R. Choo, “IoT Botnet Forensics: A Comprehensive Digital Forensic Case Study on Mirai Botnet Servers,” *Forensic Science International: Digital Investigation*, vol. 32, p. 300926, Apr. 2020, doi: 10.1016/j.fsidi.2020.300926.

[32] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, “Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning,” *IEEE Access*, vol. 10, pp. 40281–40306, 2022, doi: 10.1109/ACCESS.2022.3165809.

[33] J. Carneiro, N. Oliveira, N. Sousa, E. Maia, and I. Praça, “Machine Learning for Network-based Intrusion Detection Systems: an Analysis of the CIDDS-001 Dataset,” Jul. 02, 2021, *arXiv*: arXiv:2107.02753. Accessed: Sep. 30, 2023. doi: 10.1007/978-3-030-86261-9_15

[34] D. Powers, “Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation,” p. pp.37-63, 2011, doi: 10.48550/arXiv.2010.16061.

[35] T. Wu *et al.*, “DiscrimLoss: A Universal Loss for Hard Samples and Incorrect Samples Discrimination,” Aug. 21, 2022, *arXiv*: arXiv:2208.09884. doi: 10.1109/TMM.2023.3290477.

[36] U. Itai and N. Katz, “Goodness of Fit Metrics for Multi-class Predictor,” Aug. 11, 2022, *arXiv*: arXiv:2208.05651. doi: 10.48550/arXiv.2208.05651.