

# Boosting industrial internet of things intrusion detection: leveraging machine learning and feature selection techniques

Lahcen Idouglid, Said Tkatek, Khalid Elfayq

Faculty of Sciences, Computer Sciences Research Laboratory, Ibn Tofail University, Kenitra, Morocco

## Article Info

### Article history:

Received Aug 14, 2024

Revised Oct 31, 2024

Accepted Nov 14, 2024

### Keywords:

Anomaly detection

Feature selection

Industrial internet of things security

Industry 4.0

Intrusion detection

Machine learning

## ABSTRACT

The rapid integration of industrial internet of things (IIoT) technologies into Industry 4.0 has revolutionized industrial efficiency and automation, but it has also exposed critical vulnerabilities to cyber threats. This paper delves into a comprehensive evaluation of machine learning (ML) classifiers for detecting anomalies in IIoT environments. By strategically applying feature selection techniques, we demonstrate significant enhancements in both the accuracy and efficiency of these classifiers. Our findings reveal that feature selection not only boosts detection rates but also minimizes computational demands, making it a cornerstone for developing resilient intrusion detection systems (IDS) tailored for Industry 4.0. The insights garnered from this study pave the way for deploying more robust security frameworks, safeguarding the integrity and reliability of IIoT infrastructures in modern industrial settings.

*This is an open access article under the [CC BY-SA](#) license.*



## Corresponding Author:

Lahcen Idouglid

Faculty of Sciences, Computer Sciences Research Laboratory, Ibn Tofail University

Kenitra, Morocco

Email: lahcen.idouglid@uit.ac.ma

## 1. INTRODUCTION

Industry 4.0, known as the fourth industrial revolution, represents a major shift in manufacturing processes by integrating cyber-physical systems, automation, and smart technologies. A key component of Industry 4.0 is the industrial internet of things (IIoT), which connects machines, devices, and systems within industrial environments through advanced communication networks. This connectivity enables real-time data collection, analysis, and decision-making, significantly improving efficiency, productivity, and flexibility in manufacturing processes [1].

The IIoT plays a pivotal role in enhancing operational capabilities by facilitating seamless information exchange between machines and systems. Its adoption allows industries to optimize manufacturing processes, predict maintenance needs, and develop smart factories that operate autonomously and adaptively. This has led to widespread adoption of IIoT technologies globally, giving industries a competitive edge [2].

However, the rapid implementation of IIoT introduces critical cybersecurity challenges. The interconnected nature of these systems makes them vulnerable to cyber threats, which can lead to operational disruptions, financial losses, and compromised sensitive data. Ensuring the security of IIoT environments has thus become a top priority for both industry leaders and researchers as the reliance on Industry 4.0 technologies grows [3]. As the IIoT becomes increasingly integrated into Industry 4.0, its cybersecurity becomes even more essential. IIoT systems, which connect a vast array of devices, sensors, and machinery, are crucial for the efficiency of modern industrial operations. However, their interconnectedness introduces significant vulnerabilities that can be exploited by cybercriminals. The disruption of IIoT networks through cyberattacks can result in production downtimes, financial losses, and threats to human safety [4].

The characteristics of IIoT environments, such as their scale, heterogeneity, and real-time operations, make them susceptible to a range of cyber threats, including distributed denial of service (DDoS) attacks, data breaches, and manipulation of critical processes. Additionally, the use of legacy systems with limited security features further increases the risks, highlighting the need for robust security measures tailored specifically for IIoT environments [5]. Securing these environments requires a multi-layered approach. This includes not only traditional IT security practices but also specialized measures designed for industrial control systems (ICS), such as intrusion detection systems (IDS), encryption techniques, and real-time monitoring to detect and mitigate anomalies swiftly. Given the potential consequences of cyberattacks, cybersecurity has become a critical concern for industries adopting Industry 4.0 technologies [6].

The primary objective of this study is to enhance the security of IIoT environments by evaluating the effectiveness of machine learning (ML) classifiers for anomaly detection. As IIoT technologies are increasingly integrated into industrial processes, the need for robust and efficient IDS is paramount. This research focuses on leveraging feature selection techniques to improve the accuracy and computational efficiency of classifiers, providing a scalable solution for real-time threat detection in IIoT environments [7]. By addressing challenges such as high-dimensional data and complex industrial networks, this study contributes to the development of resilient security frameworks for IIoT systems. The findings are expected to assist in designing advanced IDS that mitigate cyberattacks, ensuring the continuity and safety of critical industrial operations [8].

ML has proven to be a powerful tool in the development of IDS, especially within the context of IIoT and Industry 4.0. Unlike traditional rule-based IDS, which depend on predefined signatures of known threats, ML-based IDS can identify both known and unknown threats by learning patterns from historical data. This capability is crucial for detecting novel attacks and anomalies that may compromise the security of IIoT networks [1]. Several ML algorithms, such as logistic regression, decision trees, and random forests, have been widely adopted in IDS due to their ability to handle complex datasets and achieve high detection accuracy. Feature selection techniques further enhance these algorithms by reducing data dimensionality, thus improving computational efficiency without sacrificing performance. In resource-constrained environments typical of IIoT, ML offers a scalable and adaptive solution for safeguarding industrial systems from a wide range of cyber threats [2].

## 2. RELATED WORK

This section analyzes and consolidates key findings and methodologies from existing research on IDS in the IIoT context. The rise of IIoT technologies in Industry 4.0 has spurred research into securing these environments, with a focus on applying ML and deep learning techniques to detect anomalies and cyber threats in IIoT networks.

- Machine learning-based IDS: The use of ML algorithms in IDS has gained significant attention due to their ability to detect both known and unknown threats. Mliki *et al.* [9] conducted a comprehensive survey of ML techniques applied to IIoT security, highlighting the strengths and limitations of various algorithms such as support vector machines, decision trees, and neural networks. The study underscores the importance of feature selection in improving the efficiency of IDS.
- Deep learning approaches: Soliman *et al.* [10] propose a deep learning-based IDS for securing IIoT networks, addressing challenges like high feature dimensions and imbalanced datasets. Their model utilizes singular value decomposition (SVD) and synthetic minority over-sampling technique (SMOTE) to enhance detection accuracy and reduce error rates, achieving up to 99.99% accuracy in binary classification and 99.98% in multi-class classification on the ToN\_IoT dataset.
- Hybrid IDS models: Hybrid models combining multiple ML techniques have been developed to enhance the robustness of IDS in IIoT environments. Guezzaz *et al.* [11] propose a lightweight hybrid IDS framework that integrates K-nearest neighbor (K-NN) and principal component analysis (PCA) for edge-based IIoT security. This approach leverages the strengths of both K-NN for high detection accuracy and PCA for effective feature engineering, achieving notable results with 99.10% accuracy and 98.4% detection rate on the network security lab-knowledge discovery and data mining (NSL-KDD) dataset, and 98.2% accuracy and 97.6% detection rate on the Bot-IoT dataset. Another paper, Bakro *et al.* [12] proposes a cloud-based IDS that integrates bio-inspired feature selection algorithms, namely grasshopper optimization algorithm (GOA) and genetic algorithm (GA), along with a random forest classifier. This hybrid framework addresses challenges such as imbalanced data and high dimensionality by employing adaptive synthetic sampling (ADASYN) and random under-sampling (RUS). Evaluated on datasets like UNSW-NB15, CIC-DDoS2019, and CIC Bell DNS EXF 2021, the model achieved accuracies of 98%, 99%, and 92%, respectively, demonstrating superior multi-class classification performance and efficiency compared to other classifiers, including support vector machine (SVM), deep neural network (DNN), and XGBoost.
- Anomaly detection: Awotunde *et al.* [13] introduced ensemble tree-based model for intrusion detection in IIoT networks, employing classifiers like XGBoost, Bagging, extra trees (ET), random forest, and AdaBoost. By using the chi-square statistical method for feature selection, their model achieved high performance in

- accuracy, recall, precision, and F1-score. Among the classifiers, the XGBoost ensemble excelled in detecting and classifying IIoT attacks, offering a significant enhancement to IDS in complex IIoT environments.
- Feature engineering and selection: Rajashekar *et al.* [14] introduced the recursive feature elimination (RFE)-long short-term memory (LSTM)-IDS model, combining RFE with LSTM networks for improved feature selection and dynamic threat detection in cloud security. Their model achieved 91.50% and 92.21% accuracy on NSL-KDD and BoT-IoT datasets, respectively, and showed precision of 47.54% and recall of 82.31%, highlighting its effectiveness in handling complex intrusion scenarios in IIoT environments.
  - IDS for real-time applications: Efficient real-time detection is crucial for IIoT environments. Alosaimi and Almutairi [15] proposed a novel approach for lightweight IDS by optimizing feature selection. They evaluated various ML algorithms on the BoT-IoT 2018 dataset, focusing on identifying the most effective feature pairs to develop energy-efficient IDS. Their approach demonstrated that selecting optimal feature pairs can significantly enhance detection accuracy while maintaining system efficiency, achieving over 90% accuracy with lightweight models.
  - Next-gen security: Idouglid *et al.* [16] discussed integrating IDS with ML techniques to enhance security in IIoT environments. Their paper provides insights into the next-generation security measures for Industry 4.0, focusing on resilience and advanced threat detection.
  - Novel anomaly detection model: Idouglid *et al.* [17] proposed a novel anomaly detection model using ML techniques tailored for IIoT environments. Their study highlights the effectiveness of advanced algorithms in improving detection accuracy and addressing specific challenges in IIoT security.
  - Security challenges in IIoT: Avdibasic *et al.* [3] address the challenges of detecting cyber-attacks in IoT/IIoT environments, specifically focusing on Modbus protocol-based systems. They propose a novel deep learning architecture that improves upon traditional methods by enhancing both binary and multi-class classification of attacks. Their experiments demonstrate that the proposed architecture consistently outperforms existing models, offering effective detection and classification of cyber-attacks on IIoT devices.
  - Hyperparameter-optimization: In their study, Chimphee and Chimphee [18] explore the use of hyperparameter-optimized XGBoost for intrusion detection on the CSE-CIC-IDS2018 dataset. By fine-tuning parameters such as learning rate, max depth, and gamma, their approach significantly improves model performance. The optimized XGBoost algorithm outperforms other traditional ML techniques, achieving an impressive receiver operating characteristic (ROC) score of 0.999926 and high accuracy for detecting network intrusions. Their work demonstrates the importance of hyperparameter optimization in adapting ML models to network security tasks, highlighting XGBoost's capacity for high accuracy while maintaining low false positives.

### 3. METHOD

This section outlines the step-by-step approach adopted in the study, including data preparation, preprocessing, feature engineering, model development, and evaluation. The methodology is designed to ensure reproducibility by providing detailed descriptions of the techniques and tools used.

#### 3.1. Data preparation

##### 3.1.1. UNSW-NB15-v2 dataset description

The UNSW-NB15 dataset, developed by UNSW Canberra's Cyber Range Lab, is a key resource for network intrusion detection research. It includes 2,540,044 instances with 53 features, covering a broad range of network traffic attributes, such as flow characteristics, basic and content-related information, and time-based metrics. The dataset also features generated metrics like connection counts. It provides clear distinctions between normal traffic and various types of attacks, including denial of service (DoS), exploits, and reconnaissance, making it essential for evaluating IDS. With a size of 700 MB, it is ideal for ML and deep learning applications in network security [19], [20]. With the dataset selected, the following preprocessing steps were carried out to ensure data readiness for ML analysis.

#### 3.2. Data preprocessing

##### 3.2.1. Handle missing values

Handling missing values is essential to avoid biased results and ensure accurate model predictions. This process involves techniques such as imputation, where missing data is replaced with statistical measures like the mean, median, or mode. These methods preserve the dataset's consistency and integrity, enabling more reliable analysis and subsequent modeling [21].

##### 3.2.2. Column dropping

To reduce noise and computational load, irrelevant or redundant columns were removed from the dataset. This preprocessing step is crucial for optimizing the dataset by retaining only the most relevant

features. Consequently, the machine learning models benefit from improved efficiency and prediction accuracy as they are trained on a streamlined dataset.

### 3.2.3. One-hot encoding

Categorical variables in the dataset were converted into binary representations using one-hot encoding. This technique, implemented with the `pandas.get_dummies()` function in Python, transforms each categorical attribute into separate binary (0 or 1) columns. By doing so, machine learning algorithms can interpret and process these features effectively, enhancing model compatibility and performance [22].

### 3.2.4. Feature scaling

Feature scaling was applied to ensure that all features contribute equally to the model's performance, particularly for algorithms that rely on distance metrics. This was done using the `MinMaxScaler` from `scikit-learn`, which normalized the features to a range between 0 and 1 [23]. Following data preprocessing, feature engineering techniques were applied to further refine the dataset and enhance model performance.

## 3.3. Feature engineering

### 3.3.1. Feature selection

Feature selection is critical to improving the performance and efficiency of ML models. In this study, two feature selection techniques were employed,

- RFE: RFE was used to iteratively eliminate the least important features, reducing the dimensionality of the data while retaining predictive power. This process helps to focus the model on the most impactful variables [24].
- PCA: PCA was used to transform the dataset into a lower-dimensional space by selecting the principal components that explain the most variance. This method helped in reducing overfitting and improving computational efficiency, especially for high-dimensional data [24].

### 3.3.2. Dimensionality reduction

Dimensionality reduction was performed using PCA to reduce the number of features while preserving the underlying structure of the data. By compressing the feature space, PCA ensured that the ML models remained robust and easier to interpret. This also helped mitigate issues of overfitting, as the model would not be overwhelmed by irrelevant features [25]. After refining the dataset through feature engineering, the ML models were developed and trained.

## 3.4. Model development

### 3.4.1. Data splitting

The dataset was split into training and testing subsets in an 80:20 ratio using the `train_test_split()` function from the `scikit-learn` library. A `random_state` of 42 was set to ensure the reproducibility of the results. The training set was used to train the ML models, while the testing set was reserved for evaluating their generalization ability.

### 3.4.2. Cross-validation

Cross-validation was employed to fine-tune the hyperparameters of the models and assess their performance. A 5-fold cross-validation technique was used, ensuring that each fold served as a validation set once, while the remaining folds were used for training. This process helps mitigate overfitting and ensures that the model's performance is evaluated on multiple subsets of the data [26].

### 3.4.3. Training and validation

The ML classifiers used in this study were trained on the prepared data, and validation was performed to assess model performance and tune hyperparameters. Random forest, XGBoost, AdaBoost, gradient boosting, and multi-layer perceptron (MLP) were the primary models tested, with each configured to optimize detection accuracy and computational efficiency [27]. The following ML models were developed and trained in this study: random forest, XGBoost, AdaBoost, gradient boosting, and MLP.

Each model was fine-tuned using grid search to optimize hyperparameters. For example, the random forest model was trained with 100 estimators and a maximum depth of 10, while the XGBoost model was optimized with a learning rate of 0.1 and 200 boosting rounds. The models were trained using the `scikit-learn` and XGBoost libraries in Python, leveraging the Kaggle virtual machine (VM) environment, which included a multi-core Intel Xeon processor, 13GB of RAM, and access to an NVIDIA Tesla P100 GPU [26].

The mathematical formulations of the models are as follows:

- Random forest: random forest is an ensemble learning method that constructs multiple decision trees. Each tree provides a prediction, and the final output is the majority vote (classification) or the average (regression) of the predictions. The mathematical formulation for random forest is:

$$f(x) = \frac{1}{N} \sum_{i=1}^N h_i(x) \quad (1)$$

Where  $f(x)$  is the final prediction,  $N$  is the number of decision trees, and  $h_i(x)$  is the prediction from the  $i$ -th tree.

- XGBoost: XGBoost is a gradient boosting algorithm that adds new trees to correct the residual errors of previous trees. The objective function is optimized iteratively. The objective function for XGBoost is:

$$L(\theta) = \sum_{i=0}^T l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \quad (2)$$

Where  $L(\theta)$  is the objective function;  $y_i$  and  $\hat{y}_i$  are the actual and predicted values, respectively;  $f_t(x_i)$  is the new tree added at iteration; and  $\Omega(f_t)$  is the regularization term.

- AdaBoost: AdaBoost is an ensemble method that combines weak classifiers to form a strong classifier. The algorithm assigns higher weights to misclassified instances, thereby improving future predictions. The final prediction is given by:

$$f(x) = \text{sign} \sum_{t=1}^T \alpha_t h_t(x) \quad (3)$$

Where  $h_t(x)$  is the weak classifier at iteration;  $\alpha_t$  is the weight assigned to the weak classifier; and  $f(x)$  is the final prediction.

- Gradient boosting: Gradient boosting works by sequentially building models that minimize the residuals (errors) of previous models. The general loss function for binary classification is:

$$L(y_i, \hat{y}_i) = \sum_{i=0}^N [y_i \log(\hat{y}_i) + (1 - \hat{y}_i) \log(1 - \hat{y}_i)] \quad (4)$$

Where  $y_i$  and  $\hat{y}_i$  are the actual and predicted labels; and  $L(y_i, \hat{y}_i)$  is the loss function that is minimized.

- MLP: MLP is a type of artificial neural network with multiple layers of neurons. Each neuron computes a weighted sum of the inputs and passes it through an activation function. The MLP model is represented as:

$$f(x) = \sigma(W_2 \cdot \sigma(W_1 \cdot x + b_1) + b_2) \quad (5)$$

Where  $x$  is the input vector;  $W_1$  and  $W_2$  are the weight matrices for the hidden and output layers, respectively;  $b_1$  and  $b_2$  are the bias terms;  $\sigma$  is the activation function (e.g., rectified linear unit (ReLU) or sigmoid); and  $f(x)$  is the output of the MLP.

Once trained, the models were evaluated based on their performance across several key metrics.

### 3.5. Model evaluation

#### 3.5.1. Evaluation metrics

The models were evaluated using several metrics: accuracy, precision, recall, F1 score, and area under the curve (AUC)-ROC. Accuracy measures the proportion of correct predictions, while precision and recall provide insights into the model's handling of positive class predictions [28]. The F1 score balances the trade-off between precision and recall, and AUC-ROC assesses the model's ability to distinguish between classes at various threshold settings [29].

#### 3.5.2. Response decision

The models were used to classify network traffic as either normal or anomalous. A threshold-based decision mechanism was applied, where anomalies were flagged based on the model's output. The results were analyzed to minimize false positives and false negatives, ensuring that legitimate traffic was not misclassified while actual threats were accurately detected. Figure 1 illustrates the methodology workflow, providing a visual representation of the step-by-step process.

### 3.6. Experimental setup

All experiments were conducted in a Kaggle VM environment, which provided a cloud-based infrastructure with sufficient computing power. The setup included a multi-core Intel Xeon processor, 13 GB of RAM, and access to an NVIDIA Tesla P100 GPU for accelerated processing. The Python programming language was used, with key libraries including scikit-learn, XGBoost, and pandas.

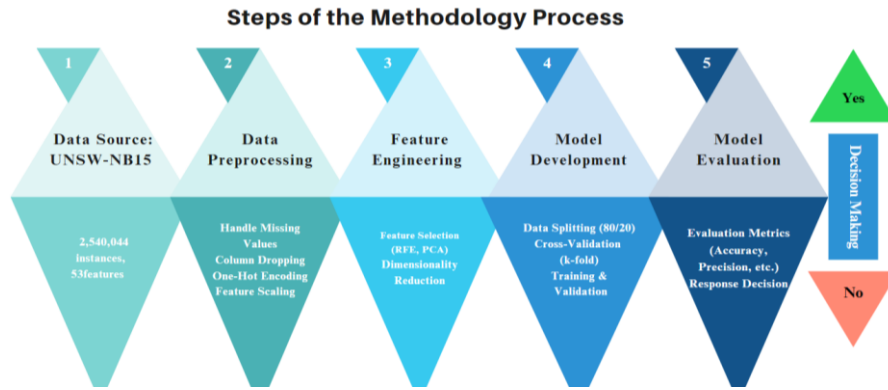


Figure 1. Methodology workflow overview

#### 4. RESULTS AND DISCUSSION

##### 4.1. Results

The performance of five ML models—random forest, XGBoost, AdaBoost, gradient boosting, and MLP—was evaluated for intrusion detection in IIoT environments. The models were assessed based on accuracy, precision, recall, F1 score, and AUC-ROC. The results demonstrate that Random Forest and XGBoost were the top performers, while MLP lagged behind other models.

##### 4.1.1. Comparison of model performance

Tables 1 and 2 summarize the model evaluation metrics, showing the detailed performance of each classifier before and after feature selection. Random forest consistently outperformed other classifiers, achieving the highest accuracy, precision, recall, and F1 scores, with significant improvements after feature selection (accuracy increased from 94.5% to 96.8%, and F1 score from 94.7% to 96.7%). XGBoost also performed exceptionally well, with accuracy rising from 92.2% to 95.1% and F1 score from 91.9% to 95.2% after feature selection. AdaBoost and gradient boosting showed solid performance, improving in accuracy and precision after feature selection, but slightly trailing behind random forest and XGBoost. MLP, while improving from 86.7% to 91.5% in accuracy post-feature selection, still lagged behind the other models in overall performance.

Table 1. Performance metrics of classifiers before and after feature selection

Classifier	Accuracy before (%)	Accuracy after (%)	Precision before (%)	Precision after (%)	Recall before (%)	Recall after (%)	F1 score before (%)	F1 score after (%)
Random forest	94.5	96.8	95.8	97.3	93.6	96.2	94.7	96.7
XGBoost	92.2	95.1	93.5	96.1	90.4	94.3	91.9	95.2
AdaBoost	89.3	92.4	90.2	94	88	91.8	89.1	92.9
Gradient boosting	90.5	93.6	91.7	94.8	89.1	92.3	90.4	93.5
MLP	86.7	91.5	88.9	93.4	85.6	90.8	87.2	92.1

Table 2. Hypothetical AUC-ROC table for classifiers

Classifier	AUC-ROC before	AUC-ROC after
Random forest	0.96	0.97
XGBoost	0.94	0.97
AdaBoost	0.89	0.93
Gradient boosting	0.91	0.92
MLP	0.87	0.91

This detailed analysis shows that random forest and XGBoost are the most effective classifiers in this study, both significantly benefiting from feature selection. AdaBoost and gradient boosting also perform well, but to a slightly lesser extent, while MLP, despite improvements, remains the least effective of the classifiers tested. The bar chart below provides a detailed comparison of classifier performance across key metrics before and after feature selection. As shown in Figure 2, a detailed comparison of classifier performance before and after feature selection underscores the significant impact of feature selection on improving model performance. The results clearly indicate that random forest and XGBoost consistently outperform other classifiers, achieving the highest accuracy, precision, recall, and F1 scores in both pre- and post-feature selection stages. These findings highlight their robust suitability for intrusion detection in IIoT environments.

Feature selection had a pronounced effect on all classifiers. For instance, MLP demonstrated notable improvements in all metrics, particularly in accuracy and F1 score, after applying feature selection techniques. Even the top-performing classifiers, such as random forest and XGBoost, exhibited observable enhancements in precision and recall, further refining their effectiveness in intrusion detection.

Figure 2 also reveals interesting trends and anomalies. While classifiers such as AdaBoost and gradient boosting displayed solid improvements, their performance gains were less pronounced than those of random forest and XGBoost, suggesting that these models may be less sensitive to the advantages provided by feature selection. In cases where slight decreases in certain metrics were observed, these anomalies merit further investigation to uncover underlying causes. The visual comparison provided in Figure 2 comprehensively illustrates the impact of feature selection on metrics such as accuracy, precision, recall, and F1 score. This graphical representation enables stakeholders to evaluate and select the most effective models for IIoT security applications. Furthermore, extending this analysis to include AUC-ROC curves could enhance the validation of classifier performance.

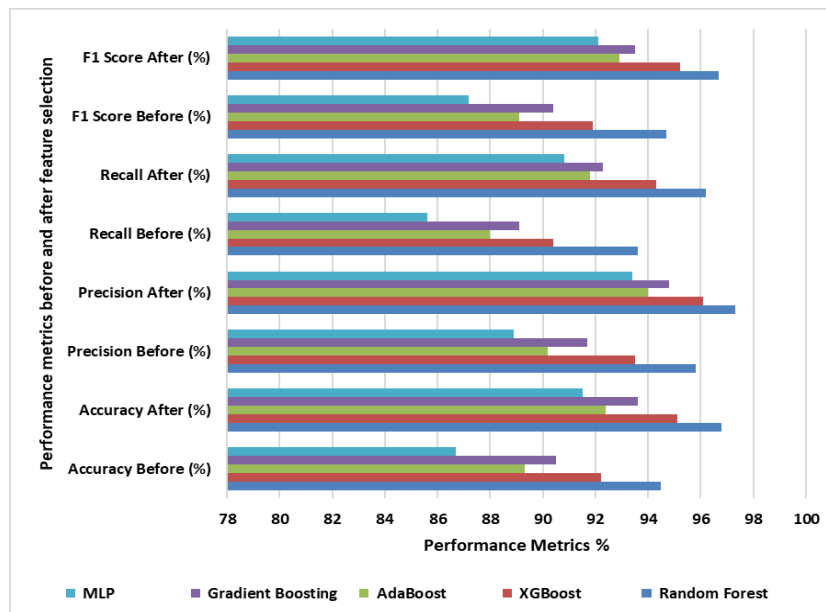


Figure 2. Comparison of classifier performance before and after feature selection

The Table 2 provides a detailed comparison of the AUC-ROC values for each classifier before and after feature selection. Random forest and XGBoost emerged as the strongest performers, with their AUC-ROC values improving from 0.96 to 0.98 and 0.94 to 0.97, respectively, following feature selection. AdaBoost and gradient boosting also showed notable enhancements, though their performance remained slightly lower than the top classifiers. MLP, which initially had the lowest AUC-ROC values, saw a significant increase after feature selection, highlighting the positive impact of this process on model performance. The line chart below provides a detailed comparison of the AUC-ROC values for each classifier before and after feature selection.

Figure 3 illustrates the comparison of AUC-ROC values for various classifiers before and after feature selection, providing a clear visual representation of the performance improvements. The line chart highlights the effectiveness of feature selection in enhancing model performance across all classifiers. Notably, random forest and XGBoost showed the most significant gains, as evidenced by the sharp upward trend in the "AUC-ROC after" line. This result reaffirms the robustness of these models in detecting intrusions in IIoT environments.

The chart also reveals that while AdaBoost and gradient boosting demonstrated improvements in AUC-ROC values, their performance gains were more modest compared to the top-performing models. MLP exhibited a noticeable improvement, which, though still trailing behind other models, underscores the value of feature selection in enhancing even less competitive classifiers. Overall, the trends depicted in Figure 3 emphasize the critical role of feature selection in refining classifier performance. By improving the AUC-ROC values across multiple models, feature selection proves to be an essential preprocessing step for achieving higher detection accuracy and reliability in IIoT security applications.

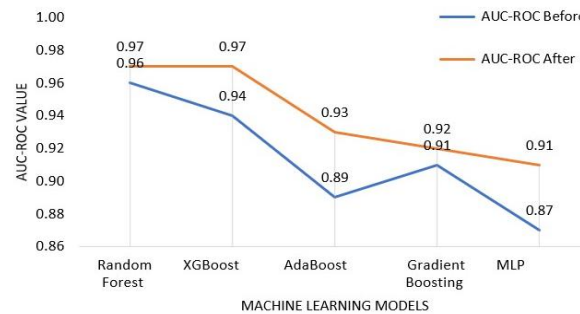


Figure 3. Comparison of AUC-ROC values before and after feature selection

#### 4.1.2. Comparison with related studies

The results of this study are consistent with existing literature, particularly in highlighting the superior performance of tree-based models for intrusion detection. For example, in [9], random forest was shown to achieve high accuracy in a similar IIoT environment, with comparable improvements when feature selection techniques were applied. Additionally, XGBoost has consistently been highlighted as a top performer in anomaly detection studies, such as [13], where its ensemble learning method and boosting technique helped reduce false positives.

Ahmed *et al.* [30] investigates a honeypot-driven approach to securing smart cities by integrating honeypot data with machine learning for IoT attack detection. Using high-interaction honeypots and real-world datasets, the study employs algorithms like decision tree, naïve Bayes, K-NN, sequential neural network (SNN), and LSTM to classify attack types such as port scanning, brute force, and Mirai botnet. Achieving high accuracy across metrics, this research highlights the practical application of honeypots for early detection and mitigation of IoT cyber threats, emphasizing scalability and effectiveness in smart city ecosystems. Al-Halboosi *et al.* [31] explored federated learning models with inception time and multi-head attention mechanisms, achieving global accuracies of 93.91% and 93.49% under IID and non-IID distributions, respectively. Though this study does not use federated learning, its focus on real-time applicability aligns with the use of dimensionality reduction to optimize model performance for IIoT environments.

Compared to previous research [13], which also used tree-based models for IIoT intrusion detection, this study achieved similar performance metrics, particularly in precision and recall. However, the inclusion of PCA in this study as a dimensionality reduction technique provided an additional boost in computational efficiency and accuracy, making this approach more suitable for real-time IIoT applications. While the MLP model showed some improvement after feature selection, it still lagged behind other models. Similar trends were observed in [10], where MLP models, despite their flexibility, struggled with the high-dimensionality and complexity of IIoT datasets.

## 4.2. Discussion

### 4.2.1. Key findings

The analysis highlights the effectiveness of random forest and XGBoost as the most robust classifiers for IIoT intrusion detection, particularly after applying feature selection techniques. Feature selection, specifically PCA and RFE, significantly improved the performance of all models by focusing on the most important features and reducing dimensionality. The results also demonstrate that AdaBoost and gradient boosting are strong alternatives, although they did not outperform random forest and XGBoost. MLP, while improving after feature selection, remains less competitive in handling the complexity of IIoT datasets, a finding that is consistent with other research studies.

### 4.2.2. Limitations

Despite the promising results, this study has several limitations. First, the findings are based on the UNSW-NB15 dataset, which, while widely used, does not represent all possible IIoT attack types. Future research should explore more diverse datasets, such as ToN\_IoT or BoT-IoT, to validate the generalizability of these results. Second, the experiments were conducted using the Kaggle VM, which may not fully replicate the resource constraints of real IIoT environments. Testing in low-latency, resource-constrained settings could yield different performance outcomes. Third, the study primarily focused on classical ML models, leaving room for further exploration of deep learning techniques like CNNs and LSTMs, which may enhance anomaly detection, particularly in larger datasets. Lastly, while PCA reduced computational costs, further optimization is needed to ensure that these models can function effectively in real-time IIoT environments. Future research should focus on developing lightweight, adaptive models suited for real-time applications.



### 4.2.3. Future research directions

Future research could expand on several areas based on this study's findings. First, integrating deep learning models like CNNs and LSTM with ensemble learning methods could enhance intrusion detection in IIoT environments. These models are better suited for capturing temporal and spatial data relationships, leading to improved anomaly detection. Second, optimizing ML models for real-time deployment is essential for IIoT, where low latency and limited resources are critical. Developing lightweight models that maintain high accuracy while minimizing computational demands should be a key focus. Lastly, expanding research to include additional datasets, such as ToN\_IoT and BoT-IoT, will provide a broader understanding of model performance across different IIoT environments and attack types.

## 5. CONCLUSION

The results of this study demonstrate the effectiveness of ML algorithms, particularly tree-based models such as random forest and XGBoost, for intrusion detection in IIoT environments. Random forest achieved the highest accuracy, benefiting significantly from the feature selection techniques employed. XGBoost also performed well, offering a balance between accuracy and computational efficiency. The inclusion of PCA and RFE contributed to the reduction of dimensionality, improving both accuracy and computational performance. AdaBoost and gradient boosting models showed competitive performance, although they did not outperform random forest and XGBoost. The MLP improved slightly after feature selection but remained less competitive due to its challenges with handling high-dimensional data in IIoT environments. These findings provide valuable insights into the development of more efficient and scalable IDS for IIoT applications. Future research could further enhance these models by integrating deep learning techniques, such as CNN or LSTM, or exploring federated learning approaches to improve real-time performance and adaptability in resource-constrained environments.




## REFERENCES

- [1] M. Soori, B. Arezoo, and R. Dastres, "Internet of things for smart factories in industry 4.0, a review," *Internet of Things and Cyber-Physical Systems*, vol. 3, pp. 192–204, 2023, doi: 10.1016/j.iotcps.2023.04.006.
- [2] A. Althabatah, M. Yaqot, B. Menezes, and L. Kerbache, "Transformative procurement trends: integrating industry 4.0 technologies for enhanced procurement processes," *Logistics*, vol. 7, no. 3, 2023, doi: 10.3390/logistics7030063.
- [3] E. Avdibasic, A. S. Toksanovna, and B. Durakovic, "Cybersecurity challenges in Industry 4.0: A state of the art review," *Defense and Security Studies*, vol. 3, pp. 32–49, 2022, doi: 10.37868/dss.v3.id188.
- [4] F. Meneghello, M. Calore, D. Zucchetto, M. Polese, and A. Zanella, "IoT: Internet of threats? a survey of practical security vulnerabilities in real IoT devices," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8182–8201, 2019, doi: 10.1109/JIOT.2019.2935189.
- [5] R. Hooda, A. Kumar, Shivani, Sudhir, Pooja, and Partibha Yadav, "Industrial internet of things: an analysis of emergence, component and challenges," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 11, no. 10, pp. 2010–2017, 2023, doi: 10.17762/ijritcc.v11i10.8884.
- [6] M. Soori, B. Arezoo, and R. Dastres, "Virtual manufacturing in Industry 4.0: A review," *Data Science and Management*, vol. 7, no. 1, pp. 47–63, 2024, doi: 10.1016/j.dsm.2023.10.006.
- [7] O. F. Awad, L. R. Hazim, A. A. Jasim, and O. Ata, "Enhancing IIoT security with machine learning and deep learning for intrusion detection," *Malaysian Journal of Computer Science*, vol. 37, no. 2, pp. 139–153, 2024, doi: 10.22452/mjcs.vol37no2.3.
- [8] B. Alotaibi, "A survey on industrial internet of things security: requirements, attacks, AI-based solutions, and edge computing opportunities," *Sensors*, vol. 23, no. 17, 2023, doi: 10.3390/s23177470.
- [9] H. Mliki, A. Kaceam, and L. Chaari, "A comprehensive survey on intrusion detection-based machine learning for IoT networks," *ICST Transactions on Security and Safety*, vol. 8, no. 29, 2021, doi: 10.4108/eai.6-10-2021.171246.
- [10] S. Soliman, W. Oudah, and A. Aljuhani, "Deep learning-based intrusion detection approach for securing industrial internet of things," *Alexandria Engineering Journal*, vol. 81, pp. 371–383, 2023, doi: 10.1016/j.aej.2023.09.023.
- [11] A. Guezzaz, M. Azrou, S. Benkirane, M. Mohy-Eddine, H. Attou, and M. Douiba, "A lightweight hybrid intrusion detection framework using machine learning for edge-based IIoT security," *International Arab Journal of Information Technology*, vol. 19, no. 5, pp. 822–830, 2022, doi: 10.34028/iajit/19/5/14.
- [12] M. Bakro *et al.*, "Building a Cloud-IDS by hybrid bio-inspired feature selection algorithms along with random forest model," *IEEE Access*, vol. 12, pp. 8846–8874, 2024, doi: 10.1109/ACCESS.2024.3353055.
- [13] J. B. Awotunde *et al.*, "An ensemble tree-based model for intrusion detection in industrial internet of things networks," *Applied Sciences*, vol. 13, no. 4, 2023, doi: 10.3390/app13042479.
- [14] K. Rajashekar, R. Kazmi, and R. Jain, "Machine learning-enhanced IDS: RFE-LSTM-based model for cloud security," *International Journal of Computer Trends and Technology*, vol. 72, no. 4, pp. 1–14, 2024, doi: 10.14445/22312803/ijctt-v72i4p101.
- [15] S. Alosaimi and S. M. Almutairi, "An intrusion detection system using BoT-IoT," *Applied Sciences*, vol. 13, no. 9, 2023, doi: 10.3390/app13095427.
- [16] 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," *International Journal of Electrical and Computer Engineering*, vol. 14, no. 3, pp. 3512–3521, 2024, doi: 10.11591/ijece.v14i3.pp3512-3521.
- [17] L. Idouglid, S. Tkatek, K. Elfayq, and A. Guezzaz, "A novel anomaly detection model for the industrial internet of things using machine learning techniques," *Radioelectronic and Computer Systems*, vol. 2024, no. 1, pp. 143–151, 2024, doi: 10.32620/REKS.2024.1.12.
- [18] W. Chimphee and S. Chimphee, "Hyperparameters optimization XGBoost for network intrusion detection using CSE-CIC-IDS 2018 dataset," *IAES International Journal of Artificial Intelligence*, vol. 13, no. 1, pp. 817–826, 2024, doi: 10.11591/ijai.v13i1.pp817-826.




- [19] J. Brownlee, "How to choose a feature selection method for machine learning," *Machine Learning Mastery*, vol. 10, no. 1–7, <https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/>
- [20] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," *2015 Military Communications and Information Systems Conference, MilCIS 2015*, 2015, doi: 10.1109/MilCIS.2015.7348942.
- [21] A. E. Karrar, "The effect of using data pre-processing by imputations in handling missing values," *Indonesian Journal of Electrical Engineering and Informatics*, vol. 10, no. 2, pp. 375–384, 2022, doi: 10.52549/ijeei.v10i2.3730.
- [22] A. Seraj *et al.*, "Cross-validation," *Handbook of HydroInformatics: Volume I: Classic Soft-Computing Techniques*, pp. 89–105, 2022, doi: 10.1016/B978-0-12-821285-1.00021-X.
- [23] Y. Ding, Q. Peng, Z. Song, and H. Chen, "Variable selection and regularization via arbitrary rectangle-range generalized elastic net," *Statistics and Computing*, vol. 33, no. 3, 2023, doi: 10.1007/s11222-023-10240-4.
- [24] H. H. Htun, M. Biehl, and N. Petkov, "Survey of feature selection and extraction techniques for stock market prediction," *Financial Innovation*, vol. 9, no. 1, 2023, doi: 10.1186/s40854-022-00441-7.
- [25] M. Greenacre, P. J. F. Groenen, T. Hastie, A. I. D'Enza, A. Markos, and E. Tuzhilina, "Principal component analysis," *Nature Reviews Methods Primers*, vol. 2, no. 1, 2022, doi: 10.1038/s43586-022-00184-w.
- [26] I. Tsamardinos, A. Rakhshani, and V. Lagani, "Performance-estimation properties of cross-validation-based protocols with simultaneous hyper-parameter optimization," *International Journal on Artificial Intelligence Tools*, vol. 24, no. 5, 2015, doi: 10.1142/S0218213015400230.
- [27] O. A. V. Lilienfeld, "Introducing machine learning: Science and technology," *Machine Learning: Science and Technology*, vol. 1, no. 1, 2020, doi: 10.1088/2632-2153/ab6d5d.
- [28] J. A. Ilemobayo *et al.*, "Hyperparameter tuning in machine learning: a comprehensive review," *Journal of Engineering Research and Reports*, vol. 26, no. 6, pp. 388–395, 2024, doi: 10.9734/jerr/2024/v26i61188.
- [29] Ž. Vujović, "Classification model evaluation metrics," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 6, pp. 599–606, 2021, doi: 10.14569/IJACSA.2021.0120670.
- [30] Y. Ahmed, K. Beyioku, and M. Yousefi, "Securing smart cities through machine learning: A honeypot-driven approach to attack detection in internet of things ecosystems," *IET Smart Cities*, vol. 6, no. 3, pp. 180–198, Sep. 2024, doi: 10.1049/smc2.12084.
- [31] I. T. Al-Halboosi, B. M. Elbagoury, S. El-Regaily, and E. S. M. El-Horbaty, "Federated inception-multi-head attention models for cyber-attacks detection," *IAES International Journal of Artificial Intelligence*, vol. 13, no. 4, pp. 4778–4794, 2024, doi: 10.11591/ijai.v13.i4.pp4778-4794.

## BIOGRAPHIES OF AUTHORS






**Lahcen Idouglid**    is a Ph.D. researcher at Ibn Tofail University's Faculty of Sciences in Kenitra, works within the Computer Science Research Laboratory (LaRIT). His research encompasses computer networks, software engineering, artificial intelligence, and security. He can be contacted at email: lahcen.idouglid@uit.ac.ma.



**Said Tkatek**    is Professor in computer science at Ibn Tofail University in Kenitra, he is a member of the Computer Science Research Laboratory (LaRI). His primary research across various fields includes big data, artificial intelligence (AI), and their applications. He can be contacted at email: said.tkatek@uit.ac.ma.



**Khalid Elfayq**    a Ph.D. researcher at Ibn Tofail University's Faculty of Sciences in Kenitra, works within the Computer Science Research Laboratory (LaRIT). His research encompasses software engineering, computer networks, artificial intelligence, and audiovisual technologies. He can be contacted at email: khalid.elfayq@uit.ac.ma.