Computers, Materials & Continua
DOI: 10.32604/cmc.2022.029283
Article

Tech Science Press

Anomaly Detection Framework in Fog-to-Things Communication for Industrial Internet of Things

Tahani Alatawi* **and Ahamed Aljuhani**

College of Computing and Information Technology, University of Tabuk, Tabuk, 71491, Saudi Arabia
*Corresponding Author: Tahani Alatawi. Email: 421009987@stu.ut.edu.sa
Received: 28 February 2022; Accepted: 06 April 2022

Abstract: The rapid development of the Internet of Things (IoT) in the industrial domain has led to the new term the Industrial Internet of Things (IIoT). The IIoT includes several devices, applications, and services that connect the physical and virtual space in order to provide smart, cost-effective, and scalable systems. Although the IIoT has been deployed and integrated into a wide range of industrial control systems, preserving security and privacy of such a technology remains a big challenge. An anomaly-based Intrusion Detection System (IDS) can be an effective security solution for maintaining the confidentiality, integrity, and availability of data transmitted in IIoT environments. In this paper, we propose an intelligent anomaly-based IDS framework in the context of fog-to-things communications to decentralize the cloud-based security solution into a distributed architecture (fog nodes) near the edge of the data source. The anomaly detection system utilizes minimum redundancy maximum relevance and principal component analysis as the featured engineering methods to select the most important features, reduce the data dimensionality, and improve detection performance. In the classification stage, anomaly-based ensemble learning techniques such as bagging, LPBoost, RUSBoost, and Adaboost models are implemented to determine whether a given flow of traffic is normal or malicious. To validate the effectiveness and robustness of our proposed model, we evaluate our anomaly detection approach on a new driven IIoT dataset called X-IIoTID, which includes new IIoT protocols, various cyberattack scenarios, and different attack protocols. The experimental results demonstrated that our proposed anomaly detection method achieved a higher accuracy rate of 99.91% and a reduced false alarm rate of 0.1% compared to other recently proposed techniques.

Keywords: Anomaly detection; anomaly-based IDS; fog computing; Internet of Things (IoT); Industrial Internet of Things (IIoT); IDS; Industrial Control Systems (ICSs)

1 Introduction

The Internet of Things (IoT) has been deployed in many critical sectors, including healthcare, energy, transportation, and industrial. This emerging technology allows multiple connected devices to send, process, and receive data with little to no human intervention. The main concept behind the IoT paradigm is to transform regular "things" into "smart objects," such as smart homes, smart vehicles, and smart factories, with the goal of facilitating human lives and improving quality of service [1,2]. The technology has been significantly involved in the manufacturing and industrial domains to increase productivity and portability, leading to the Industrial Internet of Things (IIoT) [3]. Consequently, many critical industrial systems have integrated the IoT technology as an integral part of their system's operation. Specifically, an industrial control system (ICS) is an essential component that many technologies adopt to support control systems of critical infrastructure, such as supervisory control and data acquisition (SCADA), remote terminal units, and human-machine interfaces [4].

Although the deployment of IoT in industrials provides great features for both service providers and consumers, many security and privacy issues remain a major concern, some of which were inherited from the IoT [5–8]. Cyberattacks threaten critical system infrastructure in IIoT environments, posing a significant challenge in detecting, responding to, and mitigating such attacks in a timely manner. Attacks such as distributed denial of service (DDoS) jeopardize system availability when an intruder sends a large number of malicious packets to the target, rendering the system unresponsive to its intended services and interrupting the system's operation, resulting in slow response or system failure [9,10]. Another cyber risk that endangers data confidentiality in IIoT networks is reconnaissance. An attack of this type collects vital information about connected devices [11], which an intruder can use to disrupt system functionality. In another cyberattack known as "man-in-the-middle," an attacker acts as a legitimate node between two authorized nodes. In such a situation, an intruder eavesdrops and intercepts the communication link, allowing an attacker to manipulate and discard messages [12,13]. Such an attack is extremely dangerous and difficult to detect because a legitimate device continues to communicate with the wrong node believing it is the authentic node. Fig. 1 depicts a typical attack scenario on industrial sensors in IIoT systems.

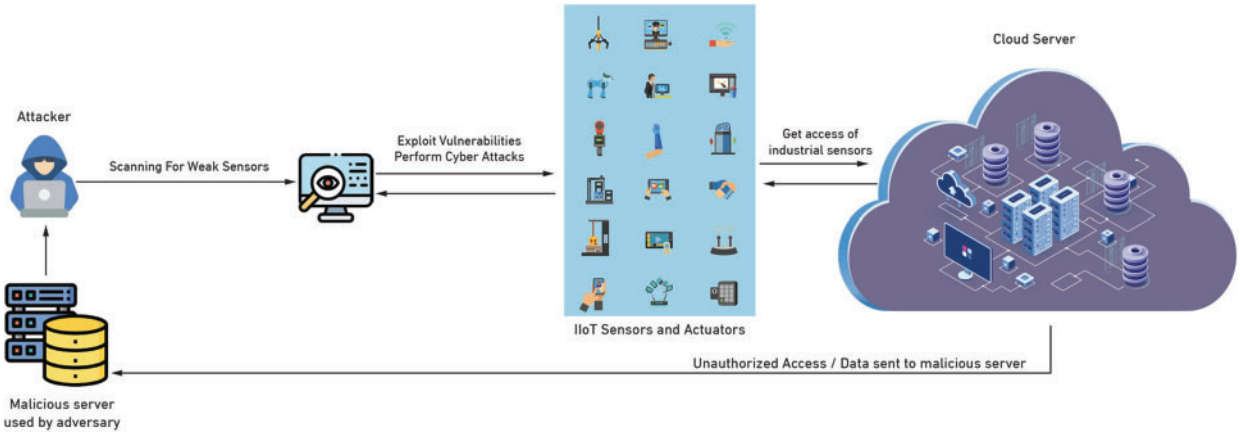


Figure 1: Cyberattack scenario performed by intruder on industrial sensors in IIoT environment

To preserve data confidentiality, integrity, and availability in IIoT environments, an intrusion detection system (IDS) can be deployed as an effective and efficient security solution to overcome many

cybersecurity risks in the IIoT networks. The IDS is categorized in two main forms [14,15]: Signature-based IDS and anomaly-based IDS. The signature-based IDS stores a list of attack signatures in the database, so whenever the incoming traffic matches the signature in the database, it is considered an attack. The main disadvantage of signature-based IDS is that it cannot detect unknown attacks (zero-day attacks). In addition, such a technique requires regular updating of the signature list to include new cyberattacks. As a result, experts must create, analyze, and update such new signature rules. The anomaly-based IDS technique overcomes the limitations of the signature-based IDS. The anomaly-based IDS technique can detect both known and unknown attacks by establishing a baseline of normal network traffic and activities. The anomaly-based IDS technique can compare the current state of network traffic to this baseline in order to detect patterns that are not normally present in traffic. Many traditional anomaly-based IDS techniques are developed for dedicated hardware or traditional networking architectures; less work has been put into designing anomaly-based IDS for IIoT environments [16]. Although different techniques have been used in the development of anomaly-based IDSs, such as statistical methods, rule-based models, and classical ML, these techniques have increased false positive rates [17].

The IIoT system is a collection of heterogeneous networks that include sensors, communication devices, and actuators collaborating to achieve common goals [18]. Anomaly detection in IIoT requires special services such as storage space, network bandwidth, and low latency to provide an efficient and effective anomaly detection mechanism. However, the traditional solutions which adapt a centralized approach of stand-alone cloud technology cannot fulfilled the security requirements of anomaly detection systems owing to resource constrains and computational overheads [19]. To overcome these limitations, a new network paradigm named fog computing can improve security services by providing cost-effective and efficient security solutions. Fog computing offers a unique opportunity for offloading security features to multiple fog nodes and supporting the provision of cyberattack detection close to the data source [20]. Because such a technology operates in a distributed architecture, the massive data generated from different sensors and actuators are processed locally to detect suspicious activities and anomalies.

In this research, we propose an intelligent anomaly-based IDS framework in the context of fog-to-things communication to ensure data confidentiality, integrity, and availability in IIoT networks. The proposed anomaly detection employs two feature engineering techniques to select the most important features, reduce data dimensionality, and improve detection performance: Minimum redundancy maximum relevance (MRMR) and principal component analysis (PCA). In the classification design, we implement different ensemble learning models such as bagging, LPBoost, RUSBoost, and Adaboost to determine whether a given flow of traffic is normal or malicious. In addition, we suggest a deployment architecture for deploying the proposed anomaly detection system as Detection as a Service (DaaS) at the fog layer and Mitigation as a Service (MaaS) at the cloud layer. The proposed model is trained and validated by using the most recent intrusion detection dataset that is specified for IIoT environments called X-IIoTID, which includes new IIoT protocols, various cyberattack scenarios, and different attack protocols. We evaluate, analyze, and compare our proposed anomaly detection method with recent previous studies. The main contributions of our research can be summarized as follows:

- We propose an anomaly detection framework in fog-to-things communications to decentralize the cloud-based security solution into a distributed architecture (fog nodes) near the edge of the data source.
- We employ two feature engineering techniques to improve the anomaly detection performance by selecting the most important features: MRMR and PCA.

- We implement the anomaly detection-based ensemble learning method to determine whether a given flow of traffic is normal or malicious.
- To address the challenges of dynamic, heterogeneous, and homogenous networks such as IIoT, fog, and cloud, we present a framework for deploying security functions as Detection as a Service (DaaS) in the fog side and Mitigation as a Service (MaaS) in the cloud side.
- Most existing anomaly detection systems are evaluated using out-of-date datasets and do not support modern IIoT-based cyberattacks. However, the effectiveness of our proposed anomaly detection model is evaluated using X-IIoTID, the most recent intrusion detection dataset that is specified for IIoT environments.
- We train, validate, and evaluate our anomaly detection model in terms of accuracy, sensitivity, specificity, f1-score, false positive rate (FPR), positive predicative value (PPV), and negative predicative value (NPV).
- We analyze and compare our proposed anomaly detection model with other existing works and find that our proposed method outperforms other techniques, with an accuracy rate of 99.91% and a reduced false alarm rate of 0.1%.

2 Related Works

A number of related works have been proposed for anomaly-based IDS in IoT/IIoT networks. Muna in [21] proposed intrusion detection technique to compact several cyberattacks in industrial control systems. The proposed method used aut-encoder and deep learning techniques to classify different malicious traffic in IIoT networks. The proposed method achieved a higher detection rate and reduced false alarm comparing to other techniques. However, due to a lack of IIoT datasets, the proposed method was tested on two datasets: NSL-KDD and UNSW-NB15.

A study by [22] proposed a deep learning technology-based detection model that can be trained and tested using data collected from remote telemetry unit streams in gas pipeline systems. Use sparse and denoising auto encoder techniques for unsupervised learning and deep neural networks for supervised learning to generate high-level data representations from unlabeled noisy data. Their results showed that the proposed model has excellent performance in identifying malicious activity. Intrusion Detection System monitors network traffic and detect network activity. However, such application is considered one of the most important security solutions to protect IIoT applications from attacks. Recently, it has been shown that applying machine and deep learning techniques can mitigate some security threats and improve intrusion detection performance.

Another study by [23] proposed an integrated protection mechanism for IoT device networks that enables threat identification, activates secure mechanisms for information transfer, and adapts to the computing capabilities of industrial IoT. The proposed solution is presented as a viable mechanism for achieving the proposed goals and detecting and containing intruders within the IoT network. In some cases, it overcomes traditional detection mechanisms such as IDS. In addition, [24] briefly described the requirements and challenges of IIoT network security and provided an overview of existing methods for detecting network anomalies. It also described other ways to detect anomalies that are particularly applicable to IIoT networks. These methods take advantage of the deterministic properties of the physical world in detecting observed behavioral anomalies.

Al-Hawawreh et al. [25] used different machine learning techniques to identify malicious traffic in IIoT. Authors created a new dataset for IIoT named X-IIoTID, and utilized different learning techniques such as decision tree, support vector machine, K-nearest Neighbor, logistic regression, naïve bayes, and deep neural network. The best performance result was obtained by the decision tree

model with accuracy rate of 99.54%. Authors in [26] proposed an ensemble technique for detecting anomalies utilizing deep learning methods. The suggested technique obtained a good accuracy result but with a higher false alarm rate. Another study by [27] proposed a feedforward neural network model for intrusion detection in IIoT environments. The proposed approach performed well in terms of accuracy metric; however, the dataset used was not designed for the IIoT environments.

Sydney in [28] proposed an intrusion detection to secure IIoT networks from cyberattacks. The proposed method used the genetic algorithm in the feature selection phase. A set of classification techniques such as Extreme Gradient Boosting (EGB), Linear Regression, NB, ET, RF, and DT. The combination of genetic algorithm with RF achieved ten features for binary classification, and seven features for multi-class classification. The UNSW-NB15 was used to evaluate and validate the effectiveness of the proposed model. The proposed intrusion detection method obtained 87.61% as an accuracy rate, and the overall experimental results were acceptable in contrast with the existing intrusion detection models. Several limitations have been observed. For instance, using synthetic datasets rather than real-time IIoT traffic may affecting the detection performance. Another problem is that outdated datasets are limited to specific types of attacks and cannot detect modern cyberattack scenarios. Additionally, many anomaly-based IDS approaches do not use a suitable intrusion dataset for IIoT, which reflects the nature of such an environment to design and develop an effective anomaly detection approach.

3 Proposed Method

The proposed framework of anomaly-based ensemble learning for IIoT networks is divided into three phases: pre-processing, feature selection, and classification (see Fig. 2). In the pre-processing phase, data cleaning, missing values compensation, and normalization are performed. In the feature selection phase, the best importance features are selected using two feature selection techniques: MRMR and PCA. In the classification phase, the following ensemble learning models are used to perform the predication: bagging, LPBoost, RUSBoost, and Adaboost. Further details concerning each phase will be illustrated in the following subsection.

3.1 Preprocessing Phase

The most significant aspects of the pre-processing stage are data cleansing, missing value compensation, and normalization. If the datatype is numeric, the process for this phase includes replacing the missing data with the mean value of that feature. If the data is nominal, the missing value is replaced with the mode value. The categorical values are encoded as integers. Following the cleaning of the dataset, the normalization process is performed to convert numeric values into new integer values ranging from 0 to 1. The Z-score algorithm is used to perform the normalization step, which is defined as follows:

$$f_k^{norm} = \frac{f_k - \mu_f}{\sigma_f} \tag{1}$$

Given a set of attributes: $\{f_1, f_2, f_3 \ldots, f_k, \ldots.., f_N\}$, $1 \leq k \leq N$, in which each attribute is represented by f_3 *and* N represents the total number of features, Z-Score f_k^{norm} is computed as described in Eq. (1).

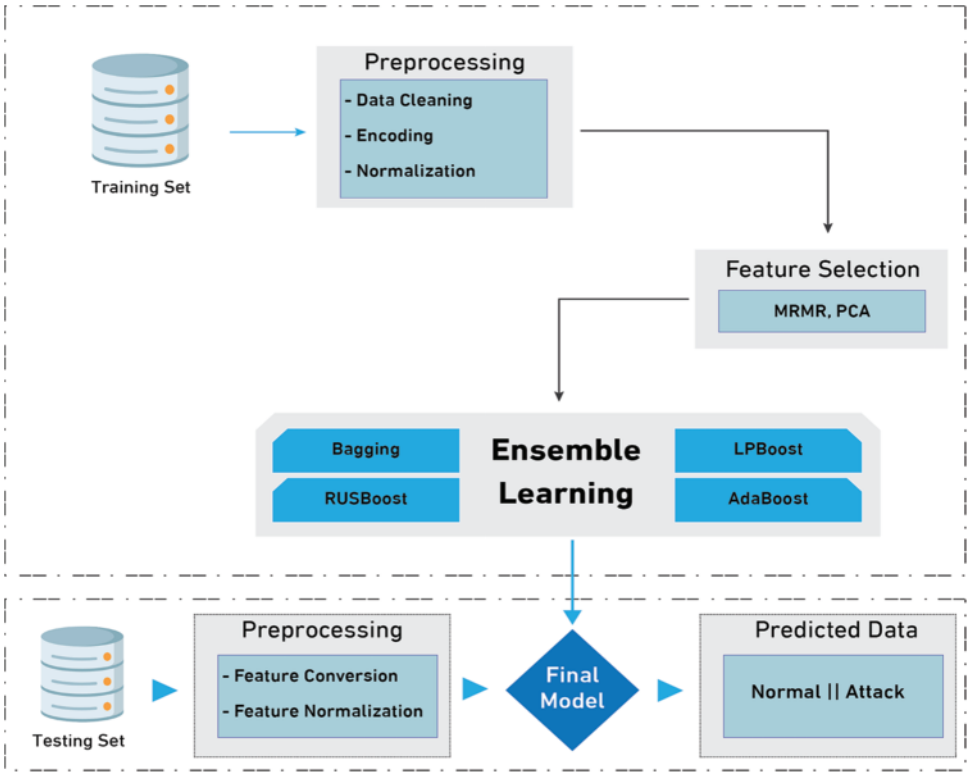


Figure 2: The architecture of proposed anomaly-based ensemble learning for IIoT

3.2 Feature Selection Phase

The feature selection stage plays a vital role in reducing data dimension, removing unnecessary features, and improving detection efficiency. We utilize MRMR and PCA as feature selection methods in our proposed anomaly detection framework. The following subsection details the MRMR and PCA methods.

3.2.1 Minimum Redundancy Maximum Relevance

MRMR [29] is used to find the "minimal–optimal" set of features. MRMR requires the user to choose only one option by indicating how many features they want to keep. MRMR is an iterative process that determines the optimal features based on rules. The selected features are added to the optimal feature list [29]. When a feature is added to the list, it never comes out. The score is determined for every feature to be assessed at each iterative process (f). Let F represents a set of data features as such f_k $\{f_1, f_2, f_3 \ldots, f_k\}$, and $FI \subseteq F$ where FI is initially empty to store the best importance features which are selected by the algorithm (see algorithm 2). The MRMR technique choose the next feature between the set of unchosen features $(F - FI)$. The score can be computed by using Eq. (2).

$$score_i(f) = \frac{relevance(f|target)}{redundancy(f|features\ selected\ until\ i-1)} \tag{2}$$

The relevance of dataset D to a target class T, represented as $R(D,\ T)$, is calculated by the mean value of the total mutual information (MI) of all data features in relation to the target class is as follows [30]:

$$R(D,\ T) = \frac{1}{F}\sum_{i=1}^{F} MI(f_i,\ T) \tag{3}$$

where F denotes the total number of data features, and T indicates the target class of the proposed ensemble models. The MI between data features and a target class T as follows [30]:

$$MI(f_i,\ T) = \sum_{x=1}^{S} p\left(f_{i,x},\ T_x\right) \log\left(\frac{p\left(f_{i,x},\ T_x\right)}{p\left(f_{i,x}\right)\ p(T_x)}\right. \tag{4}$$

where S represents the number of samples and $f_{i,x}$ denotes the xth element of feature vectors f_i, T_x of a target class T, $p\left(f_{i,x}\right)$ and $p(T_x)$, x represents the marginal probability T_x and $f_{i,x}$, respectively. The joint probability distribution of the T_x and $f_{i,x}$ is represented as $p\left(f_{i,x},\ T_x\right)$.

To minimize the redundancy among data features, V(s) depicts discrepancies between data features, which can be expressed in the Eq. (5) [30].

$$D(S) = \frac{1}{|F|^2}\sum_{i=1}^{F} MI\left(f_i,\ f_j\right) \tag{5}$$

where f_i and f_j represent ith and jth data features, and the MI is expreesed in Eq. (6) [30].

$$MI\left(f_i,\ f_j\right) = \sum_{x=1}^{S} p\left(f_{i,x},\ f_{i,\,y}\right) \log\left(\frac{p\left(f_{i,x},\ f_{i,\,y}\right)}{p\left(f_{i,x}\right)\ p(f_{i,\,y})}\right. \tag{6}$$

where $p\left(f_{i,x},\ f_{i,\,y}\right)$ computes the joint probabilistic distribution of $f_{i,x},\ f_{i,\,y}$.

Algorithm 1: Feature Selection Phase: MRMR

Input: $f_{k\ \ normalized}\ \{f_1^{norm}, f_2^{norm}, f_3^{norm},\ \ldots f_i^{norm}\}$

Output: $FI_{\ optimal\ set}$: the selected features

1: Load the candidate feature vectors

2: Create an empty basket FI to store the best selected features

3: Initializing *Minimum Redundancy Maximum Relevance*

4: **for** each iteration i **do**

5: Compute the score: $score_i(f) = \dfrac{relevance(f|target)}{redundancy(f|features\ selected\ until\ i-1)}$

6: $\text{Select}_{highest\ score}$: Maximum relevance && minimum redundancy features

7: Append the selected features f into FI basket

8: **end for**

3.2.2 PCA

The PCA is a common feature extraction and dimensionality reduction technique that is frequently used in the data science field. The PCA method aims to minimize the dimensionality of a given dataset, which includes different variables correlated with each other, and projects the dataset into a new set of variables with fewer dimensions than the original [31]. Such an anomaly detection

method collects many data features from the network, resulting in high computation complexity and increased time for attack detection, which does not fulfil the requirement of early detection [32]. The PCA technique can be an effective solution when combined with anomaly detection to reduce data features and detect anomalies in less time [32]. Let S represents the original dataset with r rows and f columns $\{s_1, s_2, \ldots, s_f\}$. The dataset is rxf matrix in which the transform data points are independent. Let M be a fxf correlation matrix of $\{s_1, s_2, \ldots, s_f\}$. Let (λ_1, e_1), (λ_2, e_2), $\ldots$ (λ_f, e_f), be f eigenvalue and eigenvector of such a matrix M. The PCA transforms S into a new dataset of variables N using the transformation matrix M as follows [32]:

$$N = M.S \tag{7}$$

We have the ith principal component as follows

$$n_i = e_i^T (s_i - \bar{s}) \tag{8}$$

where $i = 1, 2, 3, \ldots, f$.

$$e_i = \left(e_{i1}, e_{i2}, e_{i3}, \ldots e_f\right)^T \text{ is the } i^{th} \text{ eigenvector} \tag{9}$$

$$\overline{s_i} = \frac{1}{r} \sum_{i=1}^{n} si \tag{10}$$

where the $\bar{s}$ indicates the sample mean of observation s.

The first principal component will have the highest variance, the second principal component will have the next highest variance, and so on. The total variance of all PCs added together equals the total variance of the original variables $\{s_1, s_2, \ldots, s_f\}$. The principal components are sorted by eigenvalue order of importance as such: $\lambda_1 <= \lambda_2 <= \lambda_3 \ldots <= \lambda_f > 0$. Algorithm 3 illustrates the process of PCA techniques.

Algorithm 2: Feature Selection Phase: PCA

Input: A set of data feature f_k $\{f_1, f_2, f_3 \ldots, f_k\}$, where $1 \leq i \leq N$
Output: *FI*: Important feature vectors
1: Standardize the data set
2: Construct the covariance matrix
3: Compute the eigenvectors and eigenvalues
4: Select the most important features
5: Obtain the new feature vectors

3.3 Ensemble-Based Scheme for Anomaly Detection

The modeling phase receives the most essential feature representations determined by the MRMR and PCA techniques after the feature selection step. To evaluate if a particular data flow is normal or an attack, several ensemble learning models, such as bagging, LPBoost, RUSBoost, and Adaboost, are used. Each ensemble learning model is described in depth in the subsections that follow.

3.3.1 Bagging

The bagging [33] technique, also known as bootstrap aggregation, is one of the ensemble learning techniques used to minimize the variance on such a training dataset. In bagging, random samples of training dataset are chosen with replacement; in other words, the individual training set can be

selected more than once. Following the generation of several data samples, these weak learners will be trained independently for the classification task. In the bagging model, the decision is made from a group of learner models rather than an individual model to accomplish a more accurate estimate [34]. Therefore, a weak learner model may not produce an accurate decision; however, when such a weak learner is aggregated, it can become a strong learner. Such a model collectively works with other strong learners to produce the final prediction decision. To begin, a number of bootstrap samples are created, each serving as a separate independent dataset derived from a real distribution. Aggregation is the final step in producing the final prediction decision by combining the votes from each model; the final decision is based on the majority vote.

Input:

Give a training data set $S = \{(x_1,\ y_2),\ \ldots\ (x_n,\ y_n)\}$, where L is samples of sets of n elements from S

$S_1,\ S_2,\ S_3,\ \ldots\ S_L \rightarrow L$ from training sets

Train a model on each $S_i,\ i = 1,\ \ldots,\ L$ and get a sequence of L outputs $f_1(x),\ \ldots\ f_L(x)$.

Output:

Obtain the majority vote for classification problem.

$$\bar{F} = Sign\left(\sum_{i=1}^{L} Sign\ \left(f_i\left(x\right)\right)\right) \tag{11}$$

3.3.2 LPBoost

The LPBoost is an ensemble learning technique that belongs to the boosting family of classifiers. The LPBoost works by maximizing the margin between training sets of several classes and considering a classification function that predicts samples from a space X, which contains both normal and attack traffic into one of two classes, either normal or an attack. The LPBoost model learns from a given set of training examples with known class labels.

$$f : x \rightarrow\ \{-1, 1\} \tag{12}$$

As in all boosting classifiers, the final classification function is of the form:

$$f(x) = \sum_{j-1}^{j} \infty_j h_j\left(x\right) \tag{13}$$

$$h_j : x \rightarrow \{-1, 1\} \tag{14}$$

where σ_j represents a non-negative weighting for a *weak* classifier.

Any weak classifier *hj* can be better than random; however, combining many weak learners can result in a strong learner and improve classification performance. The LPBoost starts F with an empty pool of weak classifiers. Iteratively, a weak learner will become one of the other weak classifiers in such a group that includes all weak learners. The totally corrective property refers to the property that all learner weights are adjusted in each iteration.

3.3.3 RUSBoost

RUSBoost is a class labeled data boosting-based sampling strategy that tackles class imbalance [35]. It reduces the time it takes to generate a model, which is useful when creating an ensemble of models. Information loss, which is the primary drawback of random under-sampling, is greatly

minimized when random under-sampling is paired with boosting. Although some information may be absent from one of the models during creation, it will almost definitely be incorporated in the other models in the boosting ensemble. RUSBoost combines data sampling and boosting, providing a simple and efficient method for improving classification performance when training data is imbalanced. The RUSBoost Algorithm works as follow:

- First step, the weights of each example are initialized to $\frac{1}{m}$
- Then, T weak hypotheses are iteratively trained.
- After T iterations of step 2, the final hypothesis, $H(x)$, is returned as a weighted vote of the weak hypotheses.

3.3.4 Adaboost

Adaptive Boost is an ensemble method that combines several weak learners (decision trees) into a single powerful learner [36]. As a result, the system employs bagging and boosting techniques to create a more accurate predictor. The Adaboost technique is described in the following steps [37,38]:

- To separate the records, choose the best attribute using attribute selection measures.
- Compute the sample weights for each sample as follows:

$$w(xi,\ yi) = \frac{1}{L},\ i = 1,\ \ldots,\ L \tag{15}$$

where L is number of samples.

Calculate the sample weight for each sample using the equation above. The sample weight will be equal in the first round.

- Creating a decision stump out of the feature with the lowest Gini index. The Gini Impurity will be calculated for each variable.

$$Gini\ Impurity = 1 - (the\ probability\ pf\ True)^2 - (the\ probability\ pf\ False)^2 \tag{16}$$

Once you calculate the Gini Impurity of each node, the total Gini Impurity for each variable is the weighted average of the impurities of each node.

- Evaluate the performance of such a classifier in classifying the data using Eq. (17).

$$Significance = \frac{1}{2}\log\left(\frac{1 - Total\ Error}{Total\ Error}\right) \tag{17}$$

- Update the weights of the samples

$$New\ sample\ weight = sample\ weight * e^{amount\ of\ say\ \alpha} \tag{18}$$

- Normalize the weights and make a new data set.
- Repeat the previous step until we get enough number of models.
- Final prediction is performed by majority votes.

4 Deployment of Proposed Anomaly Detection as DaaS at Fog Layer and MaaS at Cloud Layer

This section presents the framework for deploying anomaly-based ensemble learning detection in IIoT environments, which can identify cyberattacks in scalable and heterogeneous IIoT networks. The

IIoT network is made up of numerous devices placed at different locations. As a result, the anomaly detection system should be capable of managing the large amount of traffic generated from these devices to effectively mitigate cyberattacks in minimal time. In this case, the anomaly detection-based centralized approach is insufficient for providing a high-quality mitigation system against different IIoT cyberattacks owing to computational resources. The proposed anomaly detection framework employs a decentralized approach that is distributed through fog nodes to improve the robustness and effectiveness of anomaly detection system against different cyber threats in IIoT networks. As shown in Fig. 3, the proposed anomaly detection framework comprises three layers: IIoT devices, fog layer, and cloud layer.

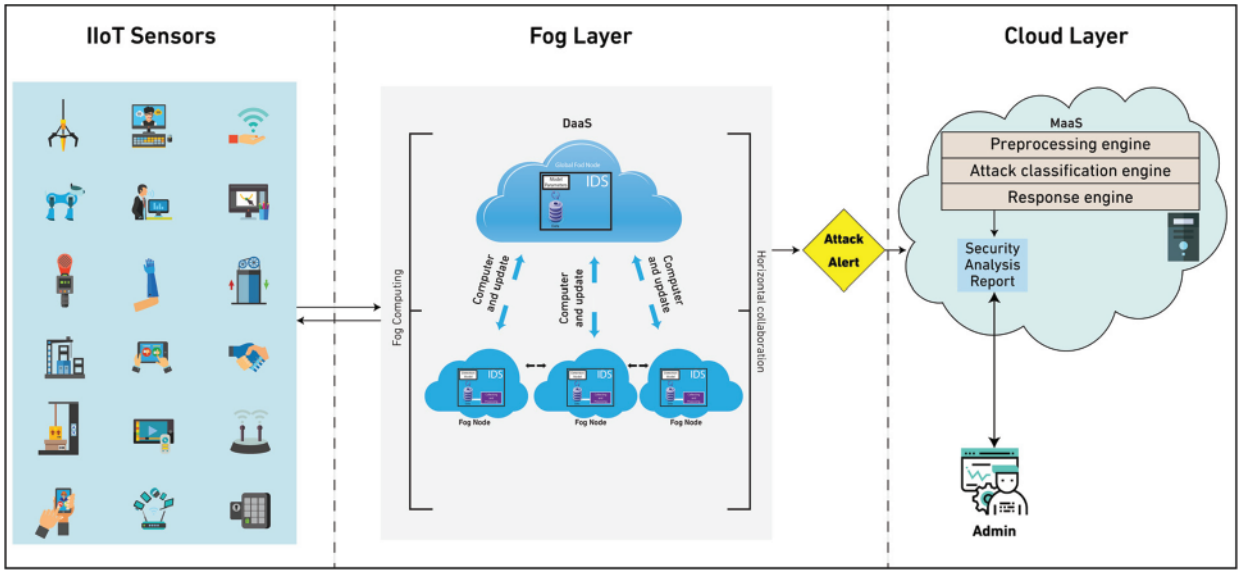


Figure 3: Deployment of anomaly detection as DaaS at fog layer and MaaS at cloud layer

The proposed anomaly-based ensemble learning framework leverages fog computing to provide a distributed anomaly detection system that utilizes a decentralized approach rather than a centralized architecture. Fog computing improves the security and privacy functions of data transmitted in such a network by providing scalable, resilient, and intelligent IIoT environments. Fog computing addresses many issues in the cloud including resource limitations, degradation, scalability, and high latencies. However, fog computing is not a replacement for the cloud; rather, it functions as an extension of the could that is located close to the end users. Therefore, our proposed anomaly detection framework follows fog-to-things architecture to provide end-to-end security protection. The collaborative anomaly-based ensemble detection as DaaS at the fog side contains a preprocessing engine and an anomaly detection engine. The preprocessing engine includes data cleansing, missing value compensation, and normalization. The anomaly detection engine is used to identify anomalies in the incoming traffic. The collaborative anomaly-based ensemble detection system is distributed at each fog node, in which each node hosts the detection parameters locally, exchanging data and learned parameters with its neighbors. The master node computes, updates, and distributes the ensemble learning parameters to each node. The architecture reveals that rather than having all operations—such as storing, training, and testing—computed at a central unit, the cloud, it distributes computation resources to each individual node, allowing it to detect anomalies efficiently and effectively near

IoT devices. This is because such an approach allows to build a scalable and a lightweight detection mechanism that is compatible with the nature of IIoT networks.

Because the fog layer is responsible for decreasing the computation load and solving the resource limitations of IoT devices, allowing for the design and development of a robust and lightweight anomaly detection system near to the applications, the cloud layer can adapt less-sensitive tasks without increasing computational overhead. The anomaly detection system at the fog layer will alert any detected anomalies and pass the traffic to the cloud for further security analysis. The MaaS in the cloud side can implement a second level of protection, such as classifying the type of anomalies or cyberattacks detected. When the type of cyberattacks is identified, the response unit applies appropriate mitigation mechanisms to prevent cyberattacks and provide information to the security analysis unit. The administrator evaluates, analyzes, and applies security policies as needed. Based on the information reported in the security analysis unit, the administrator evaluates the system‘s capabilities and performs required changes as needed to enhance future service quality.

5 Performance Analysis

This section presents the performance results of our proposed anomaly detection model. The experimental evaluation was obtained using Matlab software to create the model and using machine learning libraries to extract features, classify data, and obtain results. The tests were performed on an Intel Core i5 processor with 8 GB of RAM and Microsoft Windows 10 OS. To evaluate and analyze the proposed anomaly detection method, several performance metrics were used. These performance metrics include accuracy rate, sensitivity rate, specificity rate, F1-score, false positive rate (FPR), positive predicative value (PPV), and negative predicative value (NPV).

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative} \tag{19}$$

$$Specificity = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{20}$$

$$Sensitivity = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{21}$$

$$F1 - score = \frac{2 * True\ Positive}{2 * True\ Positive + False\ Positive + False\ Negative} \tag{22}$$

$$PPV = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{23}$$

$$NPV = \frac{True\ Negative}{True\ Negative + False\ Negative} \tag{24}$$

$$FPR = \frac{False\ Positive}{False\ Positive + True\ Negative} \tag{25}$$

5.1 Dataset

A critical step in validating our anomaly-based IDS method was selecting an appropriate dataset, which includes real-time IIoT traffic. The proposed anomaly based-IDS method was analyzed and evaluated using the X-IIoTID dataset [25]. The X-IIoTID dataset was created to fit the heterogeneity and interoperability of IIoT networks. The dataset framework is comprises of three tiers for accurately simulating IIoT networks in real-time scenarios. The first tier represents the foundation for traffic

generation to IIoT networks. The second tier analyzes, processes, and labels data. The last tier presents the final IIoT dataset. The X-IIoTID dataset includes different IIoT connectivity protocols, including machine-to-machine, machine-to-human, and human-to-machine, as well as the most recent cyberattack strategies in industrial control systems, such as command and control, brute force, malicious insider, man-in-the-middle, and DDoS attacks.

5.2 Results and Discussion

5.2.1 Evaluation Results of Ensemble Models When Using MRMR

Tab. 1 presents the performance results of several ensemble learning techniques when MRMR is employed as a feature selection technique. As shown, the Adaboost and bagging ensemble models obtained the same accuracy rate of 99.91%, achieving the best accuracy result of all the ensemble learning models used. The LPBoost ensemble model achieved a similar accuracy result of 99.38%. The RUSBoost model obtained a lower accuracy rate of 86.83%, which is the lowest performance result of all ensemble models. The bagging model obtained good performance results compared with other ensemble methods in terms of sensitivity and specificity, with scores of 99.95% and 99.86%, respectively. The Adaboost model obtained similar sensitivity and specificity results of 99.91% and 99.90%, respectively. The RUSBoost achieved lower performance results in terms of sensitivity and specificity, with scores of 92.22% and 81.52%, respectively. In terms of the F1-score, the Adaboost and bagging ensemble models achieved the same results of 99.91%, whereas the LPBoost model obtained a score of 99.37%. The RUSBoost obtained the lowest F1-score of the ensemble models with 87.42%. Fig. 4 shows the performance evaluation of different ensemble learning techniques when using MRMR.

Table 1: Performance results of different ensemble learning techniques when using MRMR method

Model	Accuracy	Sensitivity	Specificity	F1-score	FPR	PPV	NPV
RUSBoost	86.83	92.22	81.52	87.42	18.48	83.09	91.40
Adaboost	99.91	99.91	99.90	99.91	0.10	99.90	99.91
LPBoost	99.38	99.30	99.45	99.37	0.51	99.44	99.32
bagging	99.91	99.95	99.86	99.91	0.14	99.86	99.95

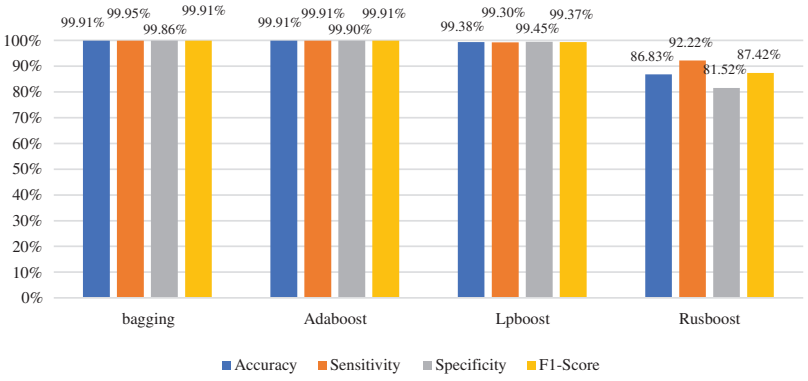


Figure 4: Performance evaluation of different ensemble learning techniques when using MRMR method

Fig. 5 depicts the NPV and PPV metrics for various ensemble learning models when MRMR is used in the feature selection phase. As Fig. 5 shows, the bagging ensemble model achieved the best NPV performance with 99.95%, while Adaboost and LPBoost achieved similar NPV results of 99.91% and 99.32%, respectively. The RUSBoost ensemble model yielded a lower NPV of 91.40%. In the case of the PPV measure, the Adaboost ensemble model outperformed other techniques with 99.90%, and the bagging model obtained a similar PPV result of 99.86%. The RUSBoost ensemble model produced a lower PPV of 83.09%. Fig. 6 presents the confusion matrix of ensemble models when using MRMR.

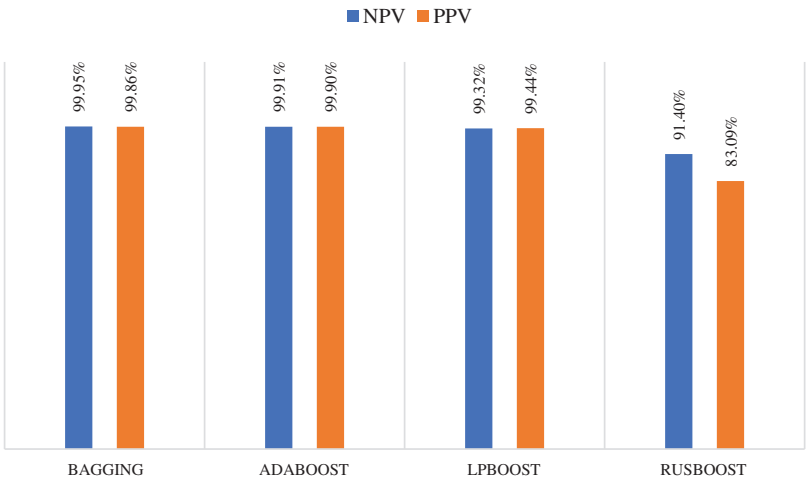
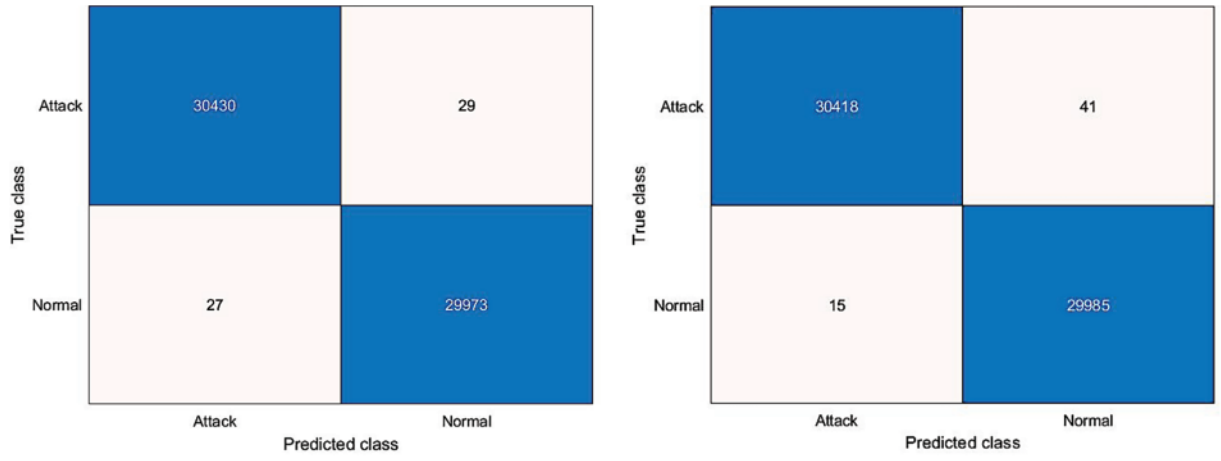


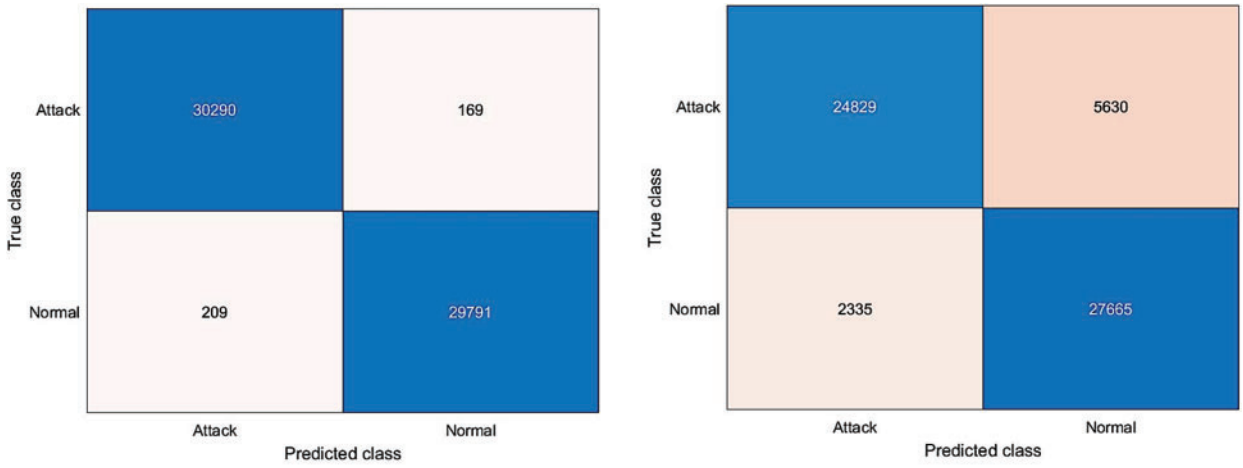
Figure 5: Negative predictive and positive predictive values of different ensemble learning models when using MRMR method



(a) Confusion matrix for AdaBoost model

(b) Confusion matrix for Bagging model

Figure 6: (Continued)



(c) Confusion matrix for LPBoost model (d) Confusion matrix for RUSBoost model

Figure 6: Confusion matrix of different ensemble learning models when using MRMR

5.2.2 Evaluation Results of Ensemble Models When Using PCA

Tab. 2 presents the performance results of several ensemble learning techniques when PCA is employed as a feature selection technique. As shown, the Adaboost and bagging ensemble models obtained similar accuracy rate with 99.46% and 99.68%, respectively. The bagging model outperformed all other ensemble models in terms of accuracy rate (99.68%). The LPBoost model achieved similar accuracy result of 99.07%, while the RUSBoost model obtained a lower accuracy rate of 88.75%, which is the lowest performance result of all ensemble models. The bagging model obtained a good performance results compared with other ensemble methods in terms of sensitivity and specificity, with scores of 99.79% and 99.58%, respectively. Whereas the Adaboost model obtained similar sensitivity and specificity results with 99.38% and 99.53%, respectively. The RUSBoost achieved lower performance results in terms of sensitivity and specificity with 87.51% and 89.98%, respectively. In terms of the F1-score, the bagging model obtained 99.68%, whereas the Adaboost and LPBoost obtained a similar performance result with 99.45% and 99.06%, respectively. The RUSBoost obtained the lowest F1-score of the ensemble models with 88.54%. Fig. 7 shows the performance evaluation of different ensemble learning techniques when using PCA.

Table 2: Performance results of different ensemble learning techniques when using PCA

Model	Accuracy	Sensitivity	Specificity	F1-score	FPR	PPV	NPV
RUSBoost	88.759	87.513	89.987	88.54	0.100	89.592	87.976
Adaboost	99.46	99.383	99.537	99.456	0.463	99.529	99.394
LPBoost	99.072	99.123	99.022	99.066	0.978	99.008	99.136
bagging	99.684	99.79	99.58	99.682	0.42	99.574	99.793

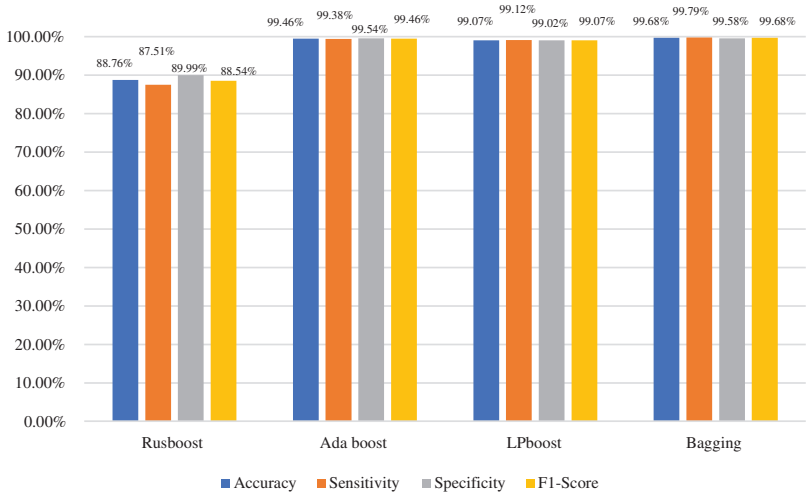


Figure 7: Performance evaluation of different ensemble learning models when using PCA technique

Fig. 8 depicts the NPV and PPV metrics for various ensemble learning models when PCA is used in the feature selection phase. As Fig. 8 shows, the bagging ensemble model achieved the best NPV performance with 99.79%, while Adaboost and LPBoost achieved similar NPV results of 99.39% and 99.14%, respectively. The RUSBoost ensemble model yielded a lower NPV of 87.98%. In the case of the PPV measure, the bagging ensemble model outperformed other techniques with 99.57%, and the Adaboost model obtained a similar PPV result of 99.53%. The RUSBoost ensemble model produced a lower PPV of 89.59%. Fig. 9 presents the confusion matrix of ensemble models when using PCA.

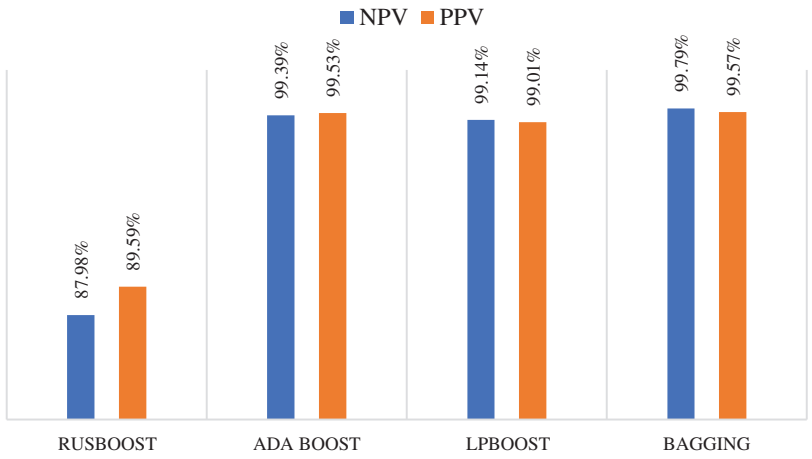
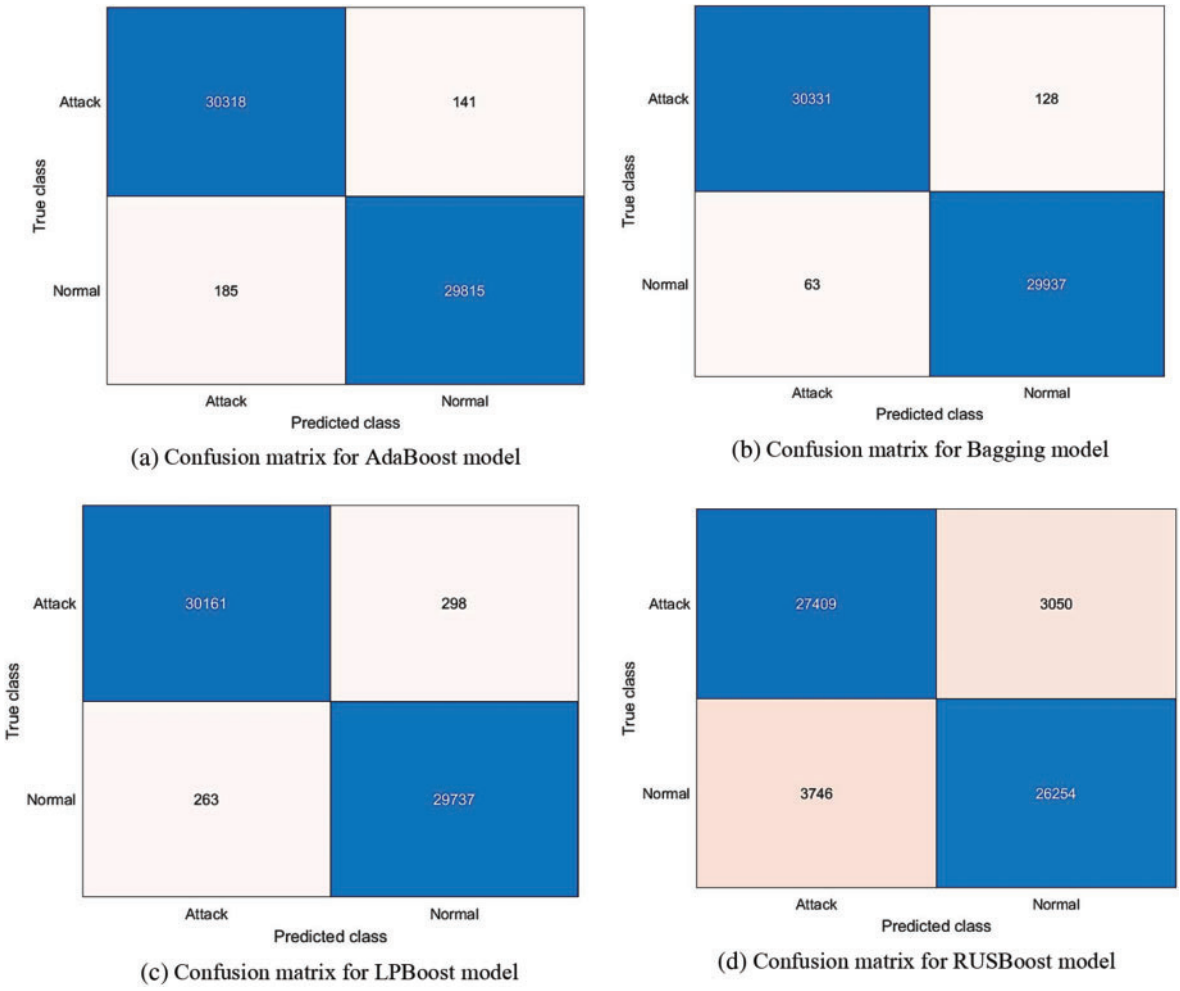


Figure 8: Negative predictive and positive predictive values of different ensemble learning models when using PCA technique



(a) Confusion matrix for AdaBoost model

(b) Confusion matrix for Bagging model

(c) Confusion matrix for LPBoost model

(d) Confusion matrix for RUSBoost model

Figure 9: Confusion matrix of different ensemble learning models when using PCA

5.2.3 Comparative Study

Fig. 10 depicts the accuracy rates of ensemble learning models when PCA and MRMR are employed in the feature selection phase. As Fig. 10 shows, the MRMR method achieved a higher accuracy than the PCA method in all ensemble learning techniques except for the RUSBoost model. The Adaboost ensemble model obtained a higher accuracy rate than all other ensemble techniques when MRMR is used as a feature selection method. However, the bagging model achieved a higher accuracy rate than all other ensemble models when using PCA as the feature selection method. Overall, the Adaboost and bagging models achieved good performance results in contrast to the LPBoost and RUSBoost models.

Tab. 3 compares the proposed method to existing methods. Our anomaly detection method outperformed a recently proposed approach in [28], who employed the genetic algorithm in the feature selection phase and the random forest model for attack classification; our method improved the accuracy rate by 12.3%. In comparison to the proposed method in [26], our proposed anomaly model improved the accuracy rate by 7.42% while significantly lowering the FPR by 14.6%. In comparison to

Al-Hawawreh et al. [22], who used deep learning techniques for attack classifications in IIoT networks, our proposed method increased the accuracy rate by 1.31%, and reduced the FPR by 1.7%. Another study by Awotunde et al. [27], who used a feedforward neural network model for attack detection in IIoT environments, our method enhanced accuracy rate by 1.01% and decreased FPR by 1.0%. Tab. 3 compares the proposed method's classification accuracy and FPR with existing anomaly detection techniques.

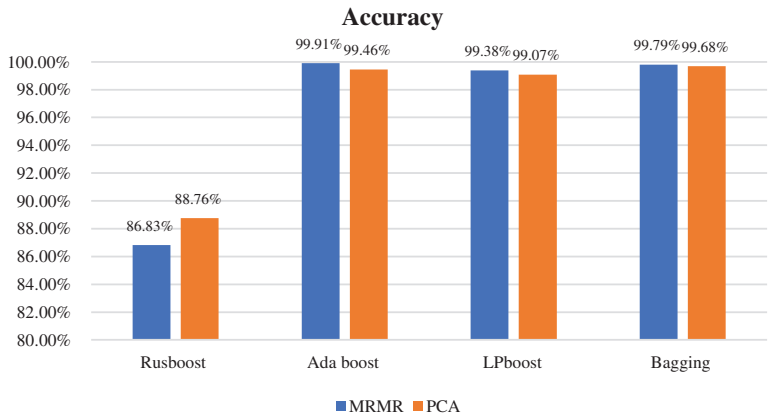


Figure 10: Comparison of MRMR and PCA with different ensemble models

Table 3: Comparison of the proposed method with existing methods

Ref	Accuracy	FPR
Kasongo [28]	87.61%	N/A
Ludwig [26]	92.49%	14.71%
AL-Hawawreh et al. [22]	98.6%	1.8%
Awotunde et al. [27]	98.9%	1.1%
Al-Hawawreh et al. [25]	99.54%	N/A
Proposed method	**99.91%**	**0.1%**

6 Conclusion

In this paper, we proposed an anomaly-based IDS to overcome different types of cyberattacks in IIoT environments. The proposed method used two feature selection techniques, MRMR and PCA, to improve the anomaly detection performance for efficiently detecting several IIoT cyberattacks. In the classification stage, several ensemble learning models were used to determine whether a given traffic flow is normal or an attack. At the validation step, we evaluated, analyzed, and compared our proposed anomaly detection method with recent previous studies. With an accuracy rate of 99.91% and a reduced false alarm rate of 0.1%, our proposed method outperformed other techniques. In the future, we will extend our method to classify different types of cyberattacks for a multiclassification problem. Also, we will investigate more feature-selection techniques and use deep learning algorithms for cyberattack classifications in IIoT networks.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] A. Khanna and S. Kaur, "Internet of Things (IoT), applications and challenges: A comprehensive review," *Wireless Personal Communications*, vol. 114, no. 2, pp. 1687–1762, 2020.

[2] S. Lee, A. Abdullah, N. Jhanjhi and S. Kok, "Classification of botnet attacks in IoT smart factory using honeypot combined with machine learning," *PeerJ Computer Science*, vol. 7, pp. 1–23, 2021.

[3] M. Aazam, S. Zeadally and K. A. Harras, "Deploying fog computing in Industrial Internet of Things and Industry 4.0," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4674–4682, 2018.

[4] M. Zolanvari, M. A. Teixeira, L. Gupta, K. M. Khan and R. Jain, "Machine learning-based network vulnerability analysis of Industrial Internet of Things," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6822–6834, 2019.

[5] P. Jayalaxmi, R. Saha, G. Kumar, N. Kumar and T. -H. Kim, "A taxonomy of security issues in Industrial Internet-of-Things: Scoping review for existing solutions, future implications, and research challenges," *IEEE Access*, vol. 9, pp. 25344–25359, 2021.

[6] J. Sengupta, S. Ruj and S. D. Bit, "A comprehensive survey on attacks, security issues and blockchain solutions for IoT and IIoT," *Journal of Network and Computer Applications*, vol. 149, pp. 1–20, 2020.

[7] M. Serror, S. Hack, M. Henze, M. Schuba and K. Wehrle, "Challenges and opportunities in securing the Industrial Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 2985–2996, 2021.

[8] F. Amin, A. Ahmad and G. Sang Choi, "Towards trust and friendliness approaches in the social Internet of Things," *Applied Science*, vol. 9, no. 1, pp. 1–25, 2019.

[9] A. Aljuhani, "Machine learning approaches for combating distributed denial of service attacks in modern networking environments," *IEEE Access*, vol. 9, pp. 42236–42264, 2021.

[10] T. Alharbi, A. Aljuhani and H. Liu, "Holistic DDoS mitigation using NFV," in *2017 IEEE 7th Annual Computing and Communication Workshop and Conf. (CCWC)*, Las Vegas, USA, pp. 1–4, 2017.

[11] M. Al-Hawawreh, F. d. Hartog and E. Sitnikova, "Targeted ransomware: A new cyber threat to edge system of brownfield Industrial Internet of Things," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 7137–7151, 2019.

[12] M. Kuzlu, C. Fair and O. Guler, "Role of artificial intelligence in the Internet of Things (IoT) cybersecurity," *Springer Discover Internet of Things*, vol. 1, no. 7, pp. 1–14, 2021.

[13] P. Wlazlo, A. Sahu, Z. Mao, H. Huang, A. Goulart *et al.,* "Man-in-the-middle attacks and defense in a power system cyber-physical testbed," *arXiv:2102.11455*. [Online]. Available: http://arxiv.org/abs/2102.11455. 2021.

[14] M. Eskandari, Z. H. Janjua, M. Vecchio and F. Antonelli, "Passban IDS: An intelligent anomaly-based intrusion detection system for IoT edge devices," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 6882–6897, 2020.

[15] H. H. Al-Maksousy, M. C. Weigle and C. Wang, "NIDS: Neural network based intrusion detection system," in *2018 IEEE International Symposium on Technologies for Homeland Security (HST)*, Boston, USA, pp. 1–6, 2018.

[16] A. Verma and V. Ranga, "Machine learning based intrusion detection systems for IoT applications," *Wireless Personal Communications*, vol. 111, no. 4, pp. 2287–2310, 2020.

[17] M. Alanazi, A. Aljuhani, "Anomaly detection for Internet of Things cyberattacks," *CMC-Computers, Materials & Continua*, vol. 72, no. 1, pp. 261–279, 2022.

[18] N. Koroniotis, N. Moustafa, F. Schiliro, P. Gauravaram and H. Janicke, "The SAir-IoT cyber testbed as a service: A novel cybertwins architecture in IIoT-based smart airports," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2021.

[19] P. Kumar, G. P. Gupta and R. Tripathi, "An ensemble learning and fog-cloud architecture-driven cyber-attack detection framework for IoMT networks," *Computer Communications*, vol. 166, pp. 110–124, 2021.
[20] P. Kumar, G. P. Gupta and R. Tripathi, "A distributed ensemble design based intrusion detection system using fog computing to protect the Internet of Things networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 10, pp. 1–18, 2020.
[21] A. Muna, N. Moustafa and E. Sitnikova, "Identification of malicious activities in Industrial Internet of Things based on deep learning models," *Journal of Information Security and Applications*, vol. 41, pp. 1–11, 2018.
[22] M. Al-Hawawreh, E. Sitnikova and F. Hartog, "An efficient intrusion detection model for edge system in brownfield Industrial Internet of Things," in *Proc. of the 2019 Conference on big Data and Internet of Things*, Melbourne Australia, pp. 83–87, 2019.
[23] H. Vargas, C. Lozano-Garzon, G. Montoya and Y. Donoso, "Detection of security attacks in Industrial IoT networks: A blockchain and machine learning approach," *Electronics*, vol. 10, no. 21:2662, pp. 1–18, 2021.
[24] L. Zhou and H. Guo, "Anomaly detection methods for IIoT networks," in *2018 IEEE Int. Conf. on Service Operations and Logistics, and Informatics (SOLI)*, Singapore, pp. 214–219, 2018.
[25] M. Al-Hawawreh, E. Sitnikova and N. Aboutorab, "X-IIoTID: A connectivity-and device-agnostic intrusion dataset for Industrial Internet of Things," in *IEEE Internet of Things Journal*, vol. 9, no. 5, pp. 3962–3977, 2021.
[26] S. A. Ludwig, "Intrusion detection of multiple attack classes using a deep neural net ensemble," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, Hawaii, USA, pp. 1–7, 2017.
[27] J. Awotunde, J. Bamidele, C. Chakraborty and A. Adeniyi, "Intrusion detection in Industrial Internet of Things network-based on deep learning model with rule-based feature selection," *Wireless Communications and Mobile Computing*, vol. 2021, pp. 1–17, 2021.
[28] S. M. Kasongo, "An advanced intrusion detection system for IIoT based on GA and tree based algorithms," *In IEEE Access*, vol. 9, pp. 113199–113212, 2021.
[29] H. Peng, F. Long and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and minredundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.
[30] F. Darbandi, A. Jafari, H. Karimipour, A. Dehghantanha, F. Derakhshan *et al.,* "Real-time stability assessment in smart cyber-physical grids: A deep learning approach," *IET Smart Grid*, vol. 3, no. 4, pp. 454–461, 2020.
[31] R. Chalapathy, N. Khoa and S. Chawla, "Robust deep learning methods for anomaly detection," in *the 26th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining*, CA, USA, pp. 3507–3508, 2020.
[32] D. H. Hoang and H. D. Nguyen, "A PCA-based method for IoT network traffic anomaly detection," in *2018 20th Int. Conf. on Advanced Communication Technology (ICACT)*, Chuncheon, South Korea, pp. 381–386, 2018.
[33] L. Breiman, "Bagging predictors," *Machine Learning*, no. 24, pp. 123–140, 1996.
[34] M. M. Rashid, J. Kamruzzaman, M. M. Hassan, T. Imam and S. Gordon, "Cyberattacks detection in IoT-based smart city applications using machine learning techniques," *International Journal of Environmental Research and Public Health*, vol. 17, no. 24, pp. 1–21, 2020.
[35] M. Adil, N. Javaid, U. Qasim, I. Ullah, M. Shafiq *et al.,* "LSTM and bat-based RUSBoost approach for electricity theft detection," *Applied Science*, vol. 10, no. 12, pp. 1–21, 2020.
[36] M. Mazini, B. Shirazi and I. Mahdavi, "Anomaly network-based intrusion detection system using a reliable hybrid artificial bee colony and AdaBoost algorithms," *Journal of King Saud University-Computer and Information Sciences*, vol. 31, no. 4, pp. 541–553, 2018.
[37] H. Blomquist and J. Moller, "Anomaly detection with machine learning–quality assurance of statistical data in the aid community," 2015. [Online]. Available: http://www.utn.uu.se/sts/cms/node/908.
[38] W. Hu, W. Hu and S. Maybank, "AdaBoost-Based algorithm for network intrusion detection," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 2, pp. 577–583, 2008.