Decision Analytics Journal 7 (2023) 100233

Contents lists available at ScienceDirect

Decision Analytics Journal

journal homepage: www.elsevier.com/locate/dajour

A new two-phase intrusion detection system with Naïve Bayes machine learning for data classification and elliptic envelop method for anomaly detection

Monika Vishwakarma, Nishtha Kesswani *

Department of Computer Science, Central University of Rajasthan, Ajmer, India

ARTICLE INFO

Keywords:
Internet of Things
Intrusion detection system
Machine learning
Naive Bayes classifier
Unsupervised elliptic envelope

ABSTRACT

Technology is pivotal in the rapid growth of services and intensifying the quality of life. Recent technology, like the Internet of Things (IoT), demonstrates an impressive performance in fast-forward development. Intrusion Detection System (IDS) is used as a lifeline to prevent attacks by classifying the activities as normal and suspicious. In this paper, we propose a two-phase IDS for IoT. In the first phase, we categorize data into four sections according to the data types (i.e., nominal, integer, binary, and float). We then classify them using different versions of the Naive Bayes classifier. After that, we use majority voting to choose the final result of the classification. In the second phase, we pass those data which behave normally or are benign in the first phase and classify them using an unsupervised elliptic envelope. We validated our work using the standard NSL-KDD, UNSW_NB15, and CIC-IDS2017 datasets. We found the proposed method more efficient than existing IDS techniques and achieved reasonable accuracy in the first phase. Furthermore, the benign data is sent to the second phase of the analysis. After the second phase, we achieved a 97% accuracy in the NSL-KDD dataset, 86.9% in the UNSW_NB15 dataset, and 98.59% accuracy in the CIC-IDS2017 dataset.

1. Introduction

In the evolutionary era, the Internet has always been performing a most significant role. Globally, the total estimate of Internet users is projected to increase from 3.9 billion in 2018 to 5.3 billion by 2023, as stated by Cisco Annual Internet Report [1]. Furthermore, the Internet of Things (IoT) is becoming increasingly widespread. IoT integrates many heterogeneous objects (such as in a smart home: intelligent bulbs, refrigerators, fans, air conditioners, automated doors, and TVs.) with various connecting technologies such as Bluetooth Low Energy (BLE), WiFi, and ZigBee. There are also other domains and applications in which the IoT can play an important role and enhance our lives quality. These applications include smart transportation, industrial automation, agriculture, and healthcare [2].

The IoT model [3] has been emerging towards formulating a cyber–physical environment where everything can be found, operated, investigated, and modernized. Because of being connected, the chances of attacks on the network increase. Many attacks and malicious incidents can affect different layers of the IoT architecture, creating security concerns. Makhdoom et al. [4] discussed the commonly known attacks on different layers, depending on the anatomy of the malware, and IoT-enabled cyber-attacks are also illustrated in a survey [5]. Similarly, Zarpel et al. [6] elaborated on intrusion detection systems in IoT. They have classified IDS based on placement strategies, detection methods, security threats, and validation strategies. Zargar et al. [7] explained in detail about Distributed Denial of Service (DDoS) attacks and also classified the countermeasures.

Intrusion Detection System (IDS) secures communication between devices and detects intrusion on the IoT layers. Many IDS has been launched for secure communication over the internet. It actively monitors malicious activity on the network and sends an alert message to the system administrator when attacks are detected. IoT devices are small and can be easily deployed in remote areas. However, the computation power is relatively low due to their small size and low battery capacity. Moreover, they use lightweight protocols for communication. For these reasons, the algorithms presented for attack detection should be lightweight with low energy consumption.

These heterogeneous networks are more prone to attacks like information leakage, service interruption, spoofing, etc., due to inadequate security measures and reliable intrusion detection systems. These attacks can cause devastating effects, such as damaging hardware, manipulating information, blocking system availability, and harming people. Hence, this is evident that the range of the effect of attacks on IoT networks varies greatly. For instance, a comparatively harmless and straightforward attack may produce no severe damage. Still, it

* Corresponding author.
E-mail addresses: monika.vish98@gmail.com (M. Vishwakarma), nishtha@curaj.ac.in (N. Kesswani).

https://doi.org/10.1016/j.dajour.2023.100233
Received 17 January 2023; Received in revised form 17 April 2023; Accepted 18 April 2023
Available online 22 April 2023

may cause a warning to social life. There is an attack on a device of critical importance, such as loss of central control of the flight. The article [8] highlights how crucial it is to build security into every layer of the IoT system to guard against cyberattacks that might endanger the health and safety of people. The paper compares the advantages and disadvantages of signature-based, anomaly-based, specification-based, and hybrid IDS techniques in the IoT. The paper also examines current issues with IoT security and possible future trend orientations.

Limited computational capacity and equipment, software, and protocol variations are the major causes of vulnerable devices. Consequently, an important gap exists between safety conditions and the defense abilities of currently available IoT devices. In particular, IoT equipment with limited computational capability, memory, and battery resources cannot perform intensive computational and susceptive safety tasks that produce large computation and communication loads [9]. Additionally, it is not reasonable to operate complicated and strong security standards. As a result, given the heterogeneity of these devices, it is quite challenging to develop and deploy a security tool that strikes a balance between security and performance.

Most IDS detection strategies collect data and classify them based on specific characteristics. However, when IoT devices communicate with each other to share information in a real network, the probability of variation in data also increases. Moreover, most existing approaches classify the data using just one phase. Another problem is the imbalanced distribution of the available datasets. If we trained our model with imbalanced datasets, the model performs bias toward the class which has more samples. However, we have proposed a two-phase approach that detects misclassified attacks as normal in the first phase. We have also used the weight initialization method for each class to overcome the imbalanced problem at the time of training. This makes the proposed IDS more stronger as compared to the existing ones. The main contributions of this paper are summarized as follows:

- The proposed IDS consists of two phases:
  - The first phase involves categorizing data into four sections based on data types and classifying them using different versions of the Naive Bayes classifier, followed by majority voting to choose the final result.
  - The second phase involves passing benign data from the first phase through an unsupervised elliptic envelope for further classification.
- The proposed method is validated using three standard datasets: NSL-KDD, UNSW_NB15, and CIC-IDS2017.
- The proposed method achieves higher efficiency and accuracy than existing IDS techniques.
- The model achieves 97% accuracy in NSL-KDD, 86.9% in UNSW_NB15, and 98.59% in CIC-IDS2017.

In this paper, we have proposed machine learning-based IDS. The rest construction of this paper is arranged as follows. Section 2 illustrates the related work. Section 3 discusses the proposed work in detail, divided into two subsections corresponding to the classification's two stages. Section 4 shows the experimental setup and results. Section 5 summarizes the paper.

2. Related work

In this section, we have discussed the literature associated with IDS in IoT. This section classifies existing IDS according to the technologies used in it. The following subsections are discussed: machine learning and deep learning-based IDS, blockchain-based IDS, and rule-based IDS.

2.1. Machine Learning and deep learning based IDS

In the trending era of Machine Learning (ML) techniques, ML algorithms such as KNN, Random Forest, and Naive Bayes perform very well with low complexity and reasonable computation time. Anthi et al. [10] proposed three-layer IDS for smart homes IoT devices. They performed a testbed experiment by deploying 8 IoT devices in a network and continuously monitoring the network traffic. They performed several attacks on the network and saved the log files. After the data collection, all the attacks are categorized into four main attack categories: DoS, Man In The Middle, Reconnaissance, and Replay attack. The first layer identifies connected IoT tools via a scanning network. The second layer classifies the packets as malicious or normal. The third layer distributes spiteful packets into one of the four attacks using nine machine-learning classification schemes and selects the best one. However, they did not clearly define the data features. Wenjuan Li et al. [11] suggested a disagreement-based semi-supervised training mechanism that works for both labeled and unlabeled. They used outdated DARPA (KDD99) data for the model evaluation.

Horng et al. [12] presented clustering and SVM-based IDS. Birch clustering algorithm has been used for the relevant feature selection for each type of attack from the primary KDD Cup 99 dataset. For each type of attack, an SVM classifier is used separately and combined with all four classifiers or building an IDS. Similarly, Eesa et al. [13] introduced a novel feature separation technique based upon a cuttlefish algorithm, and a decision tree classifier has been used to classify the intrusion. The article [14] discusses the challenges of constructing data mining systems with intelligence to detect intrusion attacks due to the large datasets employed in the learning phase. The article proposes a feature selection algorithm called "the Highest Wins" (HW) to determine the suitable set of characteristics present in the training datasets. Li et al. [15] performed a two-step software-defined technique-enabled, AI-based intrusion detection. They used a weighted voting system with a random forest within an adaptive modification of the samples' weight to categorize the flow.

An anomaly-based IDS model was proposed by Aljawarneh et al. [16]. Initially, the data were filtered using a voting system incorporating information gain to select the essential attributes. They performed binary and multiclass classification using different machine learning techniques on a 20% NSL-KDD train dataset. And got high accuracy and low false-positive, but the method was assessed using only the NSL-KDD dataset. Similarly, Moustafa et al. [17] presented an IDS by implementing the AdaBoost ensemble method using Decision Tree (DT) techniques, Naive Bayes, and Artificial Neural Network to enhance the overall achievement in terms of precision and detection rate. They identify spiteful functions that endeavor to breach network applications. Similarly, Article [18] offers a distributed ensemble design-based IDS that uses fog computing by combining k-nearest neighbors, Naive Bayes, and XGBoost, as first-level individual learners. Random Forest uses the first level's prediction results at the second level to determine the final classification.

Prabavathy et al. [19] proposed IDS by implementing fog computing. They performed IDS at fog junctions using an online sequential extreme technique to detect the incoming attacks and send them to the cloud server. A new feature selection metric system called CorrAUC has been proposed by Shafiq et al. [20], which is based on the wrapper method to separate the characteristics and preferred relevant characteristics for the machine learning algorithm by applying the area under the curve (AUC) metric. They implemented the combined TOPSIS and Shannon entropy on a bijective simple set to verify chosen characteristics for malicious traffic classification in the IoT network. Hussain et al. [21] systematically examines the security conditions, current security solutions, and the attack vectors for the IoT networks to analyze the gap among IoT security requirements.

Like machine learning, deep learning (DL) techniques are also very effective. It takes more time to train a model but gives better results

with high accuracy. Shone et al. [22] use the DL approach for intrusion detection. They proposed a Non-Symmetric Deep Auto-Encoder (NDAE) based IDS. The proposed architecture applied the Random Forest classifier with stacked NDAEs to distribute network traffic into malicious incidents and normal behavior. Tian et al. [23] proposed a web attack detection system based on distributed DL. They applied several concurrent models to increase the stability of the detection system. To effectively identify any potential intrusions and unusual traffic behavior, this study [24] suggests a CNN-based technique for anomaly-based intrusion detection systems in the Internet of Things.

For dynamic and heterogeneous networks like IoMT (Internet of Medical Things), the proposed [25] framework employs a deployment architecture of software as a service (SaaS) in the fog and infrastructure as a service (IaaS) in the cloud. The study suggests a cyber-attack detection system for IoMT networks based on ensemble learning and fog–cloud architecture. Zhao et al. [26] proposed a hybrid intrusion detection system to boost classification performance utilizing a CFS-DE (Correlation-based feature Selection-Differential Evolution) method to reduce feature dimension and a weighted Stacking classification technique to raise the weights of base classifiers upon successful training. Similarly, In this research [27], an intelligent IDS is described that can defend IoT devices against cyber threats. The system employs a conditional generative adversarial network and a lightweight CNN model to produce attack data and address the lack of training data.

2.2. Blockchain based IDSs

Wenjuan Li et al. [28] proposed blockchain signature-based IDS that targets insider attacks. They performed experiments on Both Simulation and IoT real-world environments and mainly targeted worm and flooding attacks. The rule-based database auto-updates in a collaborative IoT environment. Alexopoulos et al. [29] presented a blockchain-based architecture, where they examined a collection of warnings generated by several IDS. All nodes operated consent rules to assure the transactions' legality before adding them to a block. Kumar et al. [30] served two modules as the foundation of the suggested framework. Initially, a security and privacy module for large-scale C-ITS (Cooperative Intelligent Transport System) data was created using blockchain technology and smart contracts. The second module used a deep learning approach.

In the review paper, Meng et al. [31] discussed IDS with blockchain technology. They believed blockchains could positively influence intrusion detection by facilitating alarm exchange, data co-operative, and estimation. Shah et al. [32] examines the IoT security risks and weaknesses and suggests a system architecture based on artificial intelligence to overcome them. The suggested model aims to distinguish harmful and non-malicious smart contracts using deep learning methods and identifying malicious people trying to breach the IoT ecosystem. For non-malicious IoT data, tamper-proof storage using blockchain technology is provided.

2.3. Rule based IDSs

Raza et al. [33] proposed the first real-time IDS by deploying a distributed mini-firewall protected from an external host. The IDS is easy to Extend to detect a large number of attacks and is suitable for resource-constrained devices. Sadikin et al. [34] introduced rule-based IDS in the Zigbee IoT system. They define some rules, such as the Received Signal Strength Indication (RSSI) pattern, frame counter, traffic rate, and packet frame format. Eskandari et al. [35] Passwan IDS is constructed to be hosted and fulfilled by a competitive IoT gateway. Passban is administered as a distinct add-on device autonomously connected to the network it is supposed to protect. Nonetheless, additional training sessions might be necessary in case of alterations to the underlying network topology.

The article [36] explains how the increased connection of modern human life through the Internet of Things (IoT) has increased the risk of cyberattacks on typical home appliances. The article suggests a method for managing rules dynamically to improve defenses against cyberattacks. Tests reveal that the suggested technique significantly decreases CPU and Memory usage, increases the number of packets handled per second, and significantly improves security. Mamvong et al. [37] explained an effective safety algorithm for constrained IoT tools. They presented a numerical justification for diminishing the AES-128 algorithm's complexity using the standard algorithm's core algebraic qualities.

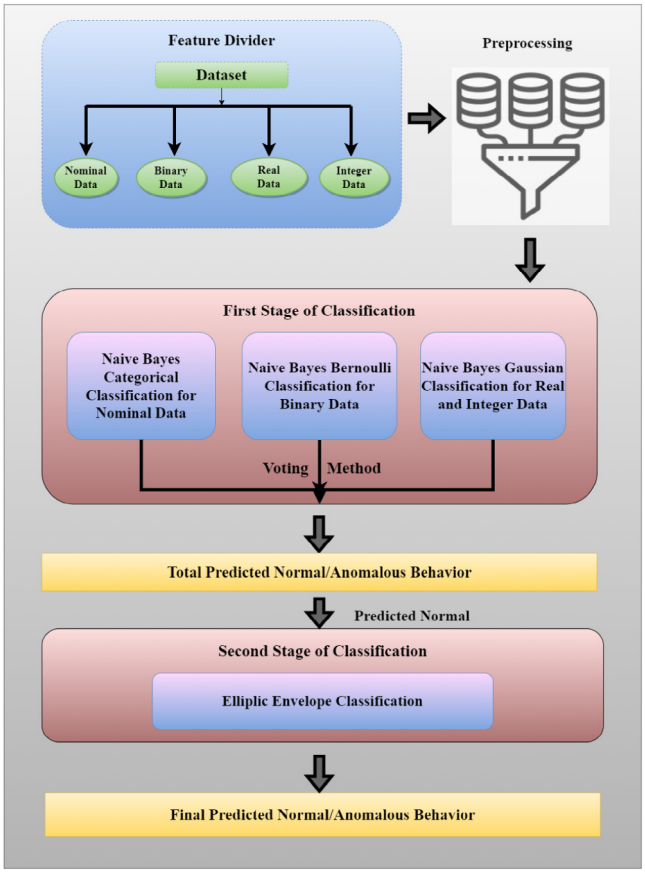


Fig. 1. The Proposed IDS.

The above literature and summary in Table 1 has explained the existing IDS, which has many issues that can affect performance. When IoT devices communicate through data sharing in an IoT environment, there is also the possibility of variation in data types failing to detect attacks. Most of the existing IDS do not examine all the features, and some classifiers may behave differently according to the data types. To address these issues, we proposed a model by considering all the features, dividing them into parts based on their data type, and processing them in parallel, reducing the computation time.

3. Proposed work

This paper has proposed a two-phase IDS to optimize efficiency without losing any relevant information related to the data. Fig. 1 shows the workflow of the proposed model.

3.1. First phase analysis

Before diving deep into the proposed method, we introduce the proposed approach briefly. We distributed our workflow of the proposed solution into five steps. The first step is to separate features into subparts according to the data type and then apply a suitable preprocessing

Table 1
Summary of existing work on IDS.

References	Experiments performed	Datasets	Advantages	Disadvantages
Anthi et al. [38]	Real-time	Benign Network Data	Real-time detection of fraudulent packets on networks.	Insufficient facts on the data characteristics.
Eskandari et al. [35]	Testbed	Generated	IoT gateway software solution with a minimal resource consumption.	If the underlying network topology is altered, it is possible that further training sessions may be necessary.
Wenjuan Li et al. [39]	Simulation and real-world	DARPA (KDD99)	It automatically takes advantage of unlabeled data.	Older data were utilized in the evaluation of the model.
Raza et al. [33]	Simulation (Contiki-Cooja)	–	Might detect unknown attacks.	The proposed model was not evaluated using real-world networking data.
Pajouh et al. [40]	–	NSL–KDD	The model suggested that it is effective for targeted assaults on minorities.	The accuracy as a whole is not up to the standard.
Hong et al. [41]	–	NSL-KDD	The article evaluates the quality of algorithms using confusion matrix, ROC curve, and AUC area.	Model was evaluated using only the NSL-KDD dataset.
Sun et al. [42]	Real-time	Network Data	Improved precision in detecting intrusions.	Detects a minimal number of assaults.
Liu et al. [43]	Simulation	–	A higher degree of precision in the identification of intrusions.	Identifying possible intrusions becomes less accurate as the amount of data grows.
Al et al. [44]	–	CICIDS2017 and UNSW-NB15	The article addresses the common issue of class imbalance in datasets.	The article does not discuss the limitations.
Engelen et al. [45]	–	CICIDS2017	The article highlights the effectiveness of machine learning techniques in network intrusion detection and the challenges associated with their adoption in large-scale network environments.	The article focuses only on the CICIDS2017 dataset and does not discuss other benchmark datasets.
Yin et al. [46]	–	UNSW-NB15	Hybrid feature selection method that combines filter and wrapper methods to manage the influence of less important features effectively.	The classification result using Multilayer Perceptron is not up to the standard.

technique in the second step. During the third step, compute weight based on the class (attacks) distribution. After the weight estimation, we classify sub-parts using different versions of Naive Bayes.

3.1.1. Feature separation based on their data type

In the internet era, heterogeneous devices are connected, share information, and communicate with each other. In the same way, the form of data on a network may be different. So the employed algorithms might have unusual behavior according to the fluctuations in the data set. Before classification, most scholars select features using techniques such as Information Gain (IG), Filter, and Wrapper. It is a great way to reduce the computation complexity, but somehow, we might be losing some important information related to the data. First, we have separated the incoming data features on their data types (i.e., object, binary, integer, float) to overcome this situation. This is essential to know the data behavior, what value is used to determine features, and how many unique values are stored in the features. For instance, the protocol_type feature has three nominal values, i.e., TCP, UDP, and ICMP.

3.1.2. Preprocessing of data

Preprocessing is another crucial step for better classification results, wherein data can be analyzed and transformed effectively. We have divided data into four categories based on their data types under the feature separation step. These data features have independent properties, like the object category containing nominal values. The binary category contains either zero or one value, and the integer category contains integer type values, the same as float having floating values.

However, before preprocessing, we need to check the feature value and range to apply the best method to transform the data without losing information.

Object features encoding. In this paper, we have used label encoding [47] techniques to transform the object-type data into numeric form. Label encoder converts all the given feature's nominal values into the numeric positive integer ranging from 0 to up to the feature's unique values; for example, protocol_type has three different values ICMP=0, TCP=1, and UDP=2. Eq. (1) shows the normalization of the data within the range of 0 to 1.

$$X_i\_normalize = \frac{X_i - X_{minimum}}{X_{maximum} - X_{minimum}} \tag{1}$$

X_i is the feature input value, $X_{minimum}$ and $X_{maximum}$ are the feature's minimum and maximum value respectively.

3.1.3. Weight initialization

Data have various properties; Somehow, the class label distribution of the data might be imbalanced. That is the most challenging task to predict the actual value with a lower false-positive rate. Sometimes accuracy is high, although the false alarm rate is also high. If the false rate of IDS is high, it means the system is not performing well, and this causes significant damage to the system or the user's privacy. So, we calculate weight based on class label distribution to reduce bias towards imbalanced data. The computed weight is as follows.

$$W_c = \frac{N}{N_c * N_s} \tag{2}$$

W_c = Weight of the class
N = Total number of samples in the data set
N_c = Total number of classes in the data set
N_s = Total number of samples in each class

Adding weight to the classifier gives priority to the class with fewer samples. So we can improve the accurate prediction rate with a low false rate.

3.1.4. Analysis methods

There are many machine learning algorithms for data classification for desired output. However, all the time, data does not have the same properties, and algorithms also behave individually depending on the data. Sometimes the same algorithm gives good results for one data and bad for another. This paper has used different versions of Naive Bayes; we have split data based on this. These separated data and weights are sent to the classifier to classify the intrusion and normal behavior.

Naive Bayes is a supervised classification method based on the Bayes theorem derived from conditional probability [48]. In this, we calculate the probability of an event when some information is already given. Here, Eq. (3) shows the probability of each class with respective features given, which is directly proportional to the product of the probability of each feature; when the class is C_i to the probability of a class, and this is divided by the product of the probability of each feature.

$$P(C_i|f_1, f_2, \ldots, f_n) \propto \frac{P(C_i) \times \prod_{j=1}^{n} P(f_j|C_i)}{P(f_1) \times P(f_2) \ldots \times P(f_n)} \tag{3}$$

$C_i = i$th class value from the no. of classes C

$f_j = j$th feature of the data set.

To find the output of the particular record, we need to add argmax, which will give us the highest probability of the class as shown in Eq. (4), which is most likely.

$$c = argmax_{i=1,2,..k} P(C_i) \times \prod_{j=1}^{n} P(f_j|C_i) \tag{4}$$

Naive Bayes categorical classification. We have done features encoding in a preprocessing step to convert the nominal value into that form in which the classifier can easily analyze the data. In the Naive Bayes categorical classification, all the data must be distributed in a discrete format, referring to nominal values. To calculate the $P(f_j|C_i)$ in Eq. (4) for the category l in the feature j given class c is estimated as:

$$P(f_j = l|C_i = c; \alpha) = \frac{N_{ljc} + \alpha}{N_c + \alpha n_r} \tag{5}$$

Where $N_{ljc} = |s \in S|f_{js} = l|C_i = c|$ is the number of times to appears in the feature f_i that is belongs to class c.

S = is the number of samples.

α = is the smoothing parameter, here it is zero.

N_c = Number of samples in the class c.

n_r = Number of categories in feature j.

Naive Bayes binary classification. In binary classification, data must be either 0 or 1. The Naive Bayes classifier calculates the likelihood probability by using the Bernoulli theorem.

$$P(f_j|C_i) = P(j|C_i)f_j + (1 - P(j|C_i)f_j)(1 - f_j) \tag{6}$$

Naive Bayes integer and float classification. For the integer and float type feature, we have implemented a Gaussian Naive classifier to classify the data to the desired output. To calculate the likelihood of $P(f_j|C_i)$ is described in Eq. (7).

$$P(f_j|C_i) = \frac{1}{\sqrt{2\pi\sigma_y^2}} e^{-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}} \tag{7}$$

After the training phase, we predicted the probability of the testing data. Calculated probability is classified into normal and anomaly by a selected threshold value, which defines the incoming event's probability as normal or anomaly.

3.1.5. Voting technique

As we applied different versions of the Naive Bayes classification algorithm based on the features data types. All the collected predictions must be arranged in that format to choose the appropriate predicted data. So we apply here the voting technique. It means we selected those predicted data that have the highest vote for belonging to a particular class. Let us suppose p_1, p_2, p_3, and p_4 are the predicted values of the data types nominal, binary, integer, and float, respectively. Then for each record, we calculate the final prediction P as follows:

$$P = mode(p_1, p_2, p_3, p_4) \tag{8}$$

Algorithm 1: First stage of classification

Input : **Class:** Labeled data set; **LA:** Learning algorithm
Output: Normal = 0; Anomaly = 1

for $i = 1$ *to* n_c ; // n_c: no. of classes
do
  $w_i = (\frac{n_s}{n_c \times n_{s_i}})$;
  /* w_i: weight of i^{th} class; n_s: total no. of samples; n_{s_i}: no. of samples in i^{th} class */
endfor
for $i = 1$ *to* n_f **do**
  if *type*(F_i) == *"object"* **then**
    $C_f = F_i$
  end
  /* C_f is the categorical classification */
  else if *type*(L_i) == *"boolean"* **then**
    $B_f = F_i$
  end
  /* B_f is the Bernoulli classification */
  else if *type*(L_i) == *"integer"* **then**
    $G_f = F_i$
  end
  else
    $G_f = F_i$
  end
  /* G_f is the Gaussian classification */
endfor
D[4] = [C_f,B_f,G_f,G_f]; λ = [0.3*to*0.5]
for $k = 1$ *to* 4 **do**
  $P_k = LA(D_k, w)$;
  /* P: Predicted probability */
  if $P < \lambda$ **then**
    $P_k = 0$;
  end
  else
    $P_k = 1$;
  end
endfor
$p = mode(P_1, P_2, P_3, P_4)$ /* p is the final predicted values 0 and 1 */

3.2. Second phase of analysis

Sometimes an attack is detected as normal due to the model's training over the unbalanced data. This can be challenging as it can affect the entire network or the user's privacy. We have implemented the second analysis phase to only those mimicking normal for accurate detection.

3.2.1. Linear discriminant analysis (LDA)

LDA [49] reduces the dimension by forming new features, a linear combination of original features using Eigen-decomposition. LDA is supervised; it reflects the class labels and finds components that depart the classes most. E.g., N_c is the number of classes, and then it converts into the $N_c - 1$ feature components. It gets optimality by minimizing the variance within the class and maximizing the variance between the classes. LDA diminishes the features into a single feature for the second stage of the classification. As a result, the computation complexity is significantly less.

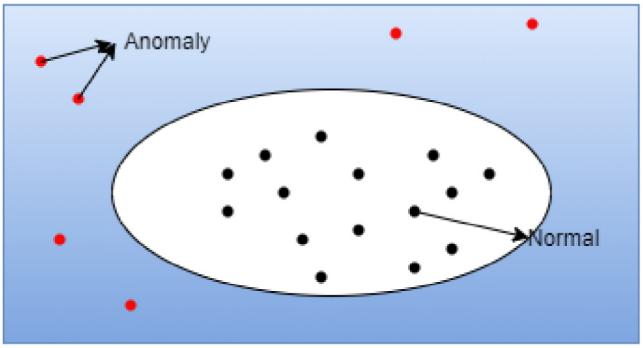


Fig. 2. Elliptic Envelope.

Table 2
NSL–KDD and UNSW_NB15 data set distribution.

NSL–KDD				UNSW_NB15	
	Train dataset	Test dataset	Train 20% dataset	Train dataset	Test dataset
Normal	67343	9711	13449	56000	9711
Anomaly	58630	12833	11743	37000	45332

3.2.2. Elliptic envelope classification

Elliptic Envelope [40] is an unsupervised learning technique that fits well if data has a Gaussian distribution. The Elliptic Envelope method is an outlier identification technique predicated on specific data points in the center of a dense elliptical area as shown in Fig. 2. However, outliers are placed distant from this region. An elliptical envelope is fitted to the data points that are thought to be normal for the technique to function. The mean and covariance matrices of the normal data points are estimated, and the Mahalanobis distance of each data point from this elliptical area is computed. Outliers are defined as data points with a high Mahalanobis distance. The elliptic envelope approach has the benefits of being resilient to outliers, handling data with several dimensions, and being effective for massive datasets. In our case, we use data that comes into doubt due to its behavior in the first analysis phase as normal.

4. Experimental setup and results

4.1. Dataset

The experiments were performed on NSL-KDD and UNSW_NB15 datasets. The NSL-KDD data set [50] is an evolved version of the KDD 99 data set, which removes many issues related to KDD data. There is no duplication in the data set, and data recorded in the train and test sets are in the relevant numbers. The data set is divided into three distinct categories: Train, Test, and Test_20%, as shown in Table 2. NSL-KDD data contains 39 different types of attacks. Test data includes 17 attacks that are unknown for the train data.

UNSW_NB-15 data set [51] was formed in the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS) by the IXIA Perfect-Storm tool. The data set is mainly split into two distinct categories: Train and Test, as shown in Table 2.

All the labeled records are divided into two classes, one for normal behavior of the data and the second class for all the attacks named anomaly for the binary classification. The accuracy, precision, recall, and f–score of different approaches on NSL–KDD data set are shown in Fig. 3

Table 3
Binary classification comparison result in percentage using NSL-KDD data set.

Models	Detection rate	False alarm rate
Proposed	**97**	5.34
TIDS [52]	86.46	–
TDTC [40]	84.82	5.83
Two-tier [53]	83.24	**4.83**
fuzziness [54]	84.12	–
Multi_CNN [55]	86.95	–

Table 4
Binary classification comparison result in percentage using CIC IDS2017 data set.

Models	Accuracy	Precision	Recall	F-score
Proposed	98.59%	95.40%	97.51%	96.44%
[56]	81.83%	–	–	90.01%
[57]	–	77%	84%	77%

Table 5
Comparison of the proposed model with existing approaches (NSL-KDD Dataset).

Models	Normal	DoS	Probe	U2L	R2L
Proposed	94.65	93.05	87.69	40.29	58.40
TIDS [52]	92.1	92.3	90.5	52.2	49.2
TDTC [40]	94.43	88.20	87.32	70.15	42
DBN [22]	95.64	87.96	72.97	0	0
S-NDAE [22]	97.73	94.58	95.67	2.70	2.82
Two-tier [53]	94.56	84.68	79.76	67.16	34.81
Multi CNN [55]	91.19	86.63	82.73	23.5	35.15

4.2. Result analysis

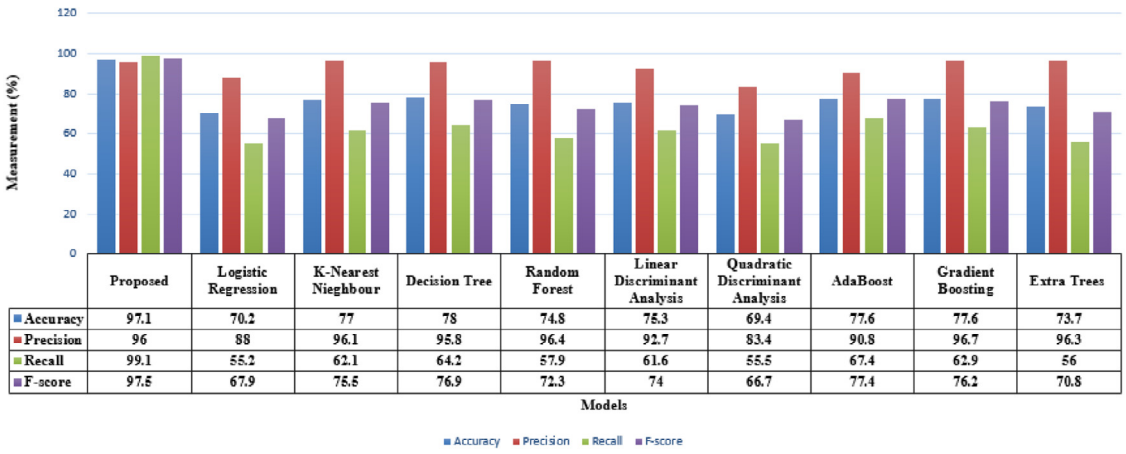
The result analysis evaluated the proposed model using different calculating performance measurements: accuracy, precision, recall, and f–score. These are the standard measurement techniques that show model performance. This section presents the overall performance of the proposed work. Various versions of Naive Bayes have worked effectively for their respective features in the first analysis phase. Following that, the second phase of analysis applies only to those behaviors which behave normally. Eventually, we got an overall 97% accuracy in the NSL-KDD dataset, as shown in Table 3, 86.9% accuracy in the UNSW_NB15 dataset, and 98.59% in CIC-IDS2017 dataset as shown in Fig. 4 and Table 4.

The accuracy, precision, recall, and f–score of different ML approaches on the UNSW_NB15 data set are shown in Fig. 4.

4.3. Comparative analysis

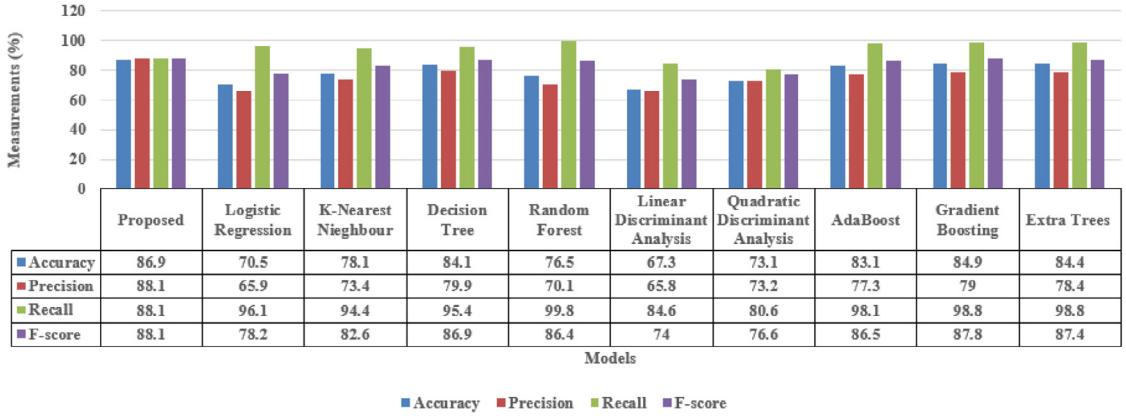
We have applied the voting method for multiclass classification after the all-learning Naive Bayes classifiers give a vote to a particular attack class. The multiclass classification of the NSL-KDD dataset results compared with the existing IDSs as shown in Table 5.

We evaluate the proposed model's performance using NSL-KDD, UNWS_NB15, and CIC-IDS2017 benchmark data sets and compare them with the existing machine learning algorithms. We have also used different machine-learning techniques to check the proposed model's performance. We trained these ML models using the training dataset and computed the accuracy, precision, recall, and f-score by passing the testing dataset to the trained model. After that, we compare our proposed model's performance with the different machine learning models. Moreover, the results show that the proposed model recall is 99.1% of the NSL-KDD dataset, which is higher than the other models. In the UNSW_NB dataset, we achieved higher accuracy of 86.9%, which is comparatively reasonable. For the CIC-IDS2017 dataset, we compare with research article [56,57].



	Proposed	Logistic Regression	K-Nearest Nieghbour	Decision Tree	Random Forest	Linear Discriminant Analysis	Quadratic Discriminant Analysis	AdaBoost	Gradient Boosting	Extra Trees
Accuracy	97.1	70.2	77	78	74.8	75.3	69.4	77.6	77.6	73.7
Precision	96	88	96.1	95.8	96.4	92.7	83.4	90.8	96.7	96.3
Recall	99.1	55.2	62.1	64.2	57.9	61.6	55.5	67.4	62.9	56
F-score	97.5	67.9	75.5	76.9	72.3	74	66.7	77.4	76.2	70.8

Fig. 3. Accuracy, precision, recall and f-score of different methods on NSL-KDD data set.



	Proposed	Logistic Regression	K-Nearest Nieghbour	Decision Tree	Random Forest	Linear Discriminant Analysis	Quadratic Discriminant Analysis	AdaBoost	Gradient Boosting	Extra Trees
Accuracy	86.9	70.5	78.1	84.1	76.5	67.3	73.1	83.1	84.9	84.4
Precision	88.1	65.9	73.4	79.9	70.1	65.8	73.2	77.3	79	78.4
Recall	88.1	96.1	94.4	95.4	99.8	84.6	80.6	98.1	98.8	98.8
F-score	88.1	78.2	82.6	86.9	86.4	74	76.6	86.5	87.8	87.4

Fig. 4. Accuracy, precision, recall and f-score of different methods on UNSW_NB15 data set.

5. Conclusion

This paper presents a machine learning-based two-phase IDS. Firstly, we categorize data into four sections according to the data types (e.g., nominal, integer, binary, and float). Then classify them using different versions of the Naive Bayes classifier. After that, with the help of majority voting, we choose the final result of the classification. In the second phase, we pass those data which behave like normal in the first stage, and these data are classified using an unsupervised elliptic envelope. It draws an imaginary envelope and assigns value 1, which lies inside the Envelope, and −1 outside the Envelope. Our proposed model is also performing very well in the imbalanced distribution of the data by providing the weight initialization to each class. Finally, we got an overall 97% accuracy with a meager false positive rate. The drawback of this model is that it does not work pretty in multiclass classification.

In future work, we will improve the multiclass classification and feature engineering techniques model, expand this procedure in real-time for network traffic analysis, and evaluate performance. We will further attempt to capture network data by deploying IoT devices in the real world. Moreover, the IDS system will detect malicious incidents in real-time and immediately take appropriate action to prevent damage.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article

References

[1] U. Cisco, Cisco annual internet report (2018–2023) white paper, Cisco, San Jose, CA, USA, 2020.
[2] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, Internet of things: A survey on enabling technologies, protocols, and applications, IEEE Commun. Surv. Tutor. 17 (4) (2015) 2347–2376.
[3] M. Nitti, V. Pilloni, G. Colistra, L. Atzori, The virtual object as a major element of the Internet of Things: A survey, IEEE Commun. Surv. Tutor. 18 (2) (2015) 1228–1240.
[4] I. Makhdoom, M. Abolhasan, J. Lipman, R.P. Liu, W. Ni, Anatomy of threats to the Internet of things, IEEE Commun. Surv. Tutor. 21 (2) (2018) 1636–1675.
[5] I. Stellios, P. Kotzanikolaou, M. Psarakis, C. Alcaraz, J. Lopez, A survey of iot-enabled cyberattacks: Assessing attack paths to critical infrastructures and services, IEEE Commun. Surv. Tutor. 20 (4) (2018) 3453–3495.
[6] B.B. Zarpelão, R.S. Miani, C.T. Kawakani, S.C. de Alvarenga, A survey of intrusion detection in Internet of Things, J. Netw. Comput. Appl. 84 (2017) 25–37.
[7] S.T. Zargar, J. Joshi, D. Tipper, A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks, IEEE Commun. Surv. Tutor. 15 (4) (2013) 2046–2069.
[8] A. Heidari, M.A. Jabraeil Jamali, Internet of Things intrusion detection systems: A comprehensive review and future directions, Cluster Comput. (2022) 1–28.
[9] L. Xiao, X. Wan, X. Lu, Y. Zhang, D. Wu, IoT security techniques based on machine learning: How do IoT devices use AI to enhance security? IEEE Signal Process. Mag. 35 (5) (2018) 41–49.
[10] E. Anthi, L. Williams, M. Słowińska, G. Theodorakopoulos, P. Burnap, A supervised intrusion detection system for smart home IoT devices, IEEE Internet Things J. 6 (5) (2019) 9042–9053.
[11] W. Li, W. Meng, M.H. Au, Enhancing collaborative intrusion detection via disagreement-based semi-supervised learning in IoT environments, J. Netw. Comput. Appl. (2020) 102631.

[12] S.-J. Horng, M.-Y. Su, Y.-H. Chen, T.-W. Kao, R.-J. Chen, J.-L. Lai, C.D. Perkasa, A novel intrusion detection system based on hierarchical clustering and support vector machines, Expert Syst. Appl. 38 (1) (2011) 306–313.
[13] A.S. Eesa, Z. Orman, A.M.A. Brifcani, A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems, Expert Syst. Appl. 42 (5) (2015) 2670–2679.
[14] R.M.A. Mohammad, M.K. Alsmadi, Intrusion detection using highest wins feature selection algorithm, Neural Comput. Appl. 33 (2021) 9805–9816.
[15] J. Li, Z. Zhao, R. Li, H. Zhang, Ai-based two-stage intrusion detection for software defined iot networks, IEEE Internet Things J. 6 (2) (2018) 2093–2102.
[16] S. Aljawarneh, M. Aldwairi, M.B. Yassein, Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model, J. Comput. Sci. 25 (2018) 152–160.
[17] N. Moustafa, B. Turnbull, K.-K.R. Choo, An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of Internet of Things, IEEE Internet Things J. 6 (3) (2018) 4815–4830.
[18] P. Kumar, G.P. Gupta, R. Tripathi, A distributed ensemble design based intrusion detection system using fog computing to protect the Internet of Things networks, J. Ambient Intell. Humaniz. Comput. 12 (2021) 9555–9572.
[19] S. Prabavathy, K. Sundarakantham, S.M. Shalinie, Design of cognitive fog computing for intrusion detection in Internet of Things, J. Commun. Netw. 20 (3) (2018) 291–298.
[20] M. Shafiq, Z. Tian, A.K. Bashir, X. Du, M. Guizani, Corrauc: A malicious bot-iot traffic detection method in iot network using machine learning techniques, IEEE Internet Things J. (2020).
[21] F. Hussain, R. Hussain, S.A. Hassan, E. Hossain, Machine learning in IoT security: Current solutions and future challenges, IEEE Commun. Surv. Tutor. 22 (3) (2020) 1686–1721.
[22] N. Shone, T.N. Ngoc, V.D. Phai, Q. Shi, A deep learning approach to network intrusion detection, IEEE Trans. Emerg. Top. Comput. Intell. 2 (1) (2018) 41–50.
[23] Z. Tian, C. Luo, J. Qiu, X. Du, M. Guizani, A distributed deep learning system for web attack detection on edge devices, IEEE Trans. Ind. Inform. 16 (3) (2019) 1963–1971.
[24] T. Saba, A. Rehman, T. Sadad, H. Kolivand, S.A. Bahaj, Anomaly-based intrusion detection system for IoT networks through deep learning model, Comput. Electr. Eng. 99 (2022) 107810.
[25] P. Kumar, G.P. Gupta, R. Tripathi, An ensemble learning and fog-cloud architecture-driven cyber-attack detection framework for IoMT networks, Comput. Commun. 166 (2021) 110–124.
[26] R. Zhao, Y. Mu, L. Zou, X. Wen, A hybrid intrusion detection system based on feature selection and weighted stacking classifier, IEEE Access 10 (2022) 71414–71426.
[27] K.-H. Le, M.-H. Nguyen, T.-D. Tran, N.-D. Tran, IMIDS: An intelligent intrusion detection system against cyber threats in IoT, Electronics 11 (4) (2022) 524.
[28] W. Li, S. Tug, W. Meng, Y. Wang, Designing collaborative blockchained signature-based intrusion detection in IoT environments, Future Gener. Comput. Syst. 96 (2019) 481–489.
[29] N. Alexopoulos, E. Vasilomanolakis, N.R. Ivánkó, M. Mühlhäuser, Towards blockchain-based collaborative intrusion detection systems, in: International Conference on Critical Information Infrastructures Security, Springer, 2017, pp. 107–118.
[30] R. Kumar, P. Kumar, R. Tripathi, G.P. Gupta, N. Kumar, M.M. Hassan, A privacy-preserving-based secure framework using blockchain-enabled deep-learning in cooperative intelligent transport system, IEEE Trans. Intell. Transp. Syst. 23 (9) (2021) 16492–16503.
[31] W. Meng, E.W. Tischhauser, Q. Wang, Y. Wang, J. Han, When intrusion detection meets blockchain technology: A review, Ieee Access 6 (2018) 10179–10188.
[32] H. Shah, D. Shah, N.K. Jadav, R. Gupta, S. Tanwar, O. Alfarraj, A. Tolba, M.S. Raboaca, V. Marina, Deep learning-based malicious smart contract and intrusion detection system for IoT environment, Mathematics 11 (2) (2023) 418.
[33] S. Raza, L. Wallgren, T. Voigt, SVELTE: Real-time intrusion detection in the Internet of Things, Ad Hoc Netw. 11 (8) (2013) 2661–2674.
[34] F. Sadikin, T. van Deursen, S. Kumar, A hybrid Zigbee IoT intrusion detection system using secure and efficient data collection, Internet of Things (2020) 100306.
[35] M. Eskandari, Z.H. Janjua, M. Vecchio, F. Antonelli, Passban IDS: An intelligent anomaly based intrusion detection system for IoT edge devices, IEEE Internet Things J. (2020).
[36] P. Nespoli, D. Díaz-López, F.G. Mármol, Cyberprotection in IoT environments: A dynamic rule-based solution to defend smart devices, J. Inform. Secur. Appl. 60 (2021) 102878.
[37] J.N. Mamvong, G.L. Goteng, B. Zhou, Y. Gao, Efficient security algorithm for power constrained IoT devices, IEEE Internet Things J. (2020).
[38] E. Anthi, L. Williams, M. Słowińska, G. Theodorakopoulos, P. Burnap, A supervised intrusion detection system for smart home IoT devices, IEEE Internet Things J. 6 (5) (2019) 9042–9053.
[39] W. Li, W. Meng, M.H. Au, Enhancing collaborative intrusion detection via disagreement-based semi-supervised learning in IoT environments, J. Netw. Comput. Appl. 161 (2020) 102631.
[40] H.H. Pajouh, R. Javidan, R. Khayami, D. Ali, K.-K.R. Choo, A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in IoT backbone networks, IEEE Trans. Emerg. Top. Comput. (2016).
[41] R.-F. Hong, S.-C. Horng, S.-S. Lin, Machine learning in cyber security analytics using NSL-KDD dataset, in: 2021 International Conference on Technologies and Applications of Artificial Intelligence, TAAI, 2021, pp. 260–265, http://dx.doi.org/10.1109/TAAI54685.2021.00057.
[42] H. Sun, X. Wang, R. Buyya, J. Su, CloudEyes: Cloud-based malware detection with reversible sketch for resource-constrained Internet of Things (IoT) devices, Softw. - Pract. Exp. 47 (3) (2017) 421–441.
[43] L. Liu, B. Xu, X. Zhang, X. Wu, An intrusion detection method for Internet of Things based on suppressed fuzzy clustering, EURASIP J. Wireless Commun. Networking 2018 (2018) 1–7.
[44] M. Al Olaimat, D. Lee, Y. Kim, J. Kim, J. Kim, A learning-based data augmentation for network anomaly detection, in: 2020 29th International Conference on Computer Communications and Networks, ICCCN, 2020, pp. 1–10, http://dx.doi.org/10.1109/ICCCN49398.2020.9209598.
[45] G. Engelen, V. Rimmer, W. Joosen, Troubleshooting an intrusion detection dataset: The CICIDS2017 case study, in: 2021 IEEE Security and Privacy Workshops, SPW, 2021, pp. 7–12, http://dx.doi.org/10.1109/SPW53761.2021.00009.
[46] Y. Yin, J. Jang-Jaccard, W. Xu, A. Singh, J. Zhu, F. Sabrina, J. Kwak, IGRF-RFE: A hybrid feature selection method for MLP-based network intrusion detection on UNSW-NB15 dataset, J. Big Data 10 (1) (2023) 1–26.
[47] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.
[48] H. Zhang, The optimality of naive Bayes, Aa 1 (2) (2004) 3.
[49] A.A. Aburomman, M.B.I. Reaz, Ensemble of binary SVM classifiers based on PCA and LDA feature extraction for intrusion detection, in: 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference, IMCEC, IEEE, 2016, pp. 636–640.
[50] L. Dhanabal, S. Shantharajah, A study on NSL-KDD dataset for intrusion detection system based on classification algorithms, Int. J. Adv. Res. Comput. Commun. Eng. 4 (6) (2015) 446–452.
[51] N. Moustafa, J. Slay, UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set), in: 2015 Military Communications and Information Systems Conference, MilCIS, IEEE, 2015, pp. 1–6.
[52] M. Vishwakarma, N. Kesswani, A two-stage intrusion detection system (tids) for Internet of Things, in: Advances in Deep Learning, Artificial Intelligence and Robotics: Proceedings of the 2nd International Conference on Deep Learning, Artificial Intelligence and Robotics, ICDLAIR 2020, Springer, 2022, pp. 89–97.
[53] H.H. Pajouh, G. Dastghaibyfard, S. Hashemi, Two-tier network anomaly detection model: A machine learning approach, J. Intell. Inf. Syst. 48 (1) (2017) 61–74.
[54] R.A.R. Ashfaq, X.-Z. Wang, J.Z. Huang, H. Abbas, Y.-L. He, Fuzziness based semi-supervised learning approach for intrusion detection system, Inform. Sci. 378 (2017) 484–497.
[55] Y. Li, Y. Xu, Z. Liu, H. Hou, Y. Zheng, Y. Xin, Y. Zhao, L. Cui, Robust detection for network intrusion of industrial IoT based on multi-CNN fusion, Measurement 154 (2020) 107450.
[56] A. Yulianto, P. Sukarno, N.A. Suwastika, Improving adaboost-based intrusion detection system (IDS) performance on CIC IDS 2017 dataset, in: J. Phys. Conf. Ser., 1192 (1) (2019) 012018.
[57] I. Sharafaldin, A.H. Lashkari, A.A. Ghorbani, Toward generating a new intrusion detection dataset and intrusion traffic characterization, ICISSp 1 (2018) 108–116.