Arabian Journal for Science and Engineering
https://doi.org/10.1007/s13369-021-06086-5

RESEARCH ARTICLE-COMPUTER ENGINEERING AND COMPUTER SCIENCE

# A New Ensemble-Based Intrusion Detection System for Internet of Things

Adeel Abbas[1] · Muzaazam A. Khan[1,2] · Shahid Latif[3] · Maria Ajaz[1] · Awais Aziz Shah[4] · Jawad Ahmad[5]

Received: 21 March 2021 / Accepted: 12 August 2021

**Abstract**

The domain of Internet of Things (IoT) has witnessed immense adaptability over the last few years by drastically transforming human lives to automate their ordinary daily tasks. This is achieved by interconnecting heterogeneous physical devices with different functionalities. Consequently, the rate of cyber threats has also been raised with the expansion of IoT networks which puts data integrity and stability on stake. In order to secure data from misuse and unusual attempts, several intrusion detection systems (IDSs) have been proposed to detect the malicious activities on the basis of predefined attack patterns. The rapid increase in such kind of attacks requires improvements in the existing IDS. Machine learning has become the key solution to improve intrusion detection systems. In this study, an ensemble-based intrusion detection model has been proposed. In the proposed model, logistic regression, naive Bayes, and decision tree have been deployed with voting classifier after analyzing model's performance with some prominent existing state-of-the-art techniques. Moreover, the effectiveness of the proposed model has been analyzed using CICIDS2017 dataset. The results illustrate significant improvement in terms of accuracy as compared to existing models in terms of both binary and multi-class classification scenarios.

**Keywords** Intrusion detection · IoT · Machine learning · Security · Anomaly detection · Ensemble learning

## 1 Introduction

Today, our planet is surrounded by a plethora of electronic devices that are transforming human lives. In this regard, Internet of Things (IoT) is emerging as an innovative technology that is transforming the industry and life smarter with intelligent devices having enhanced connectivity such as healthcare monitoring, environment monitoring, water management, smart agriculture, and smart home. More precisely in IoT, many heterogeneous physical devices can cooperate and communicate with one another for transferring the data over large number of networks without interference of human-to-human or human-to-device interfaces [1–4]. Figure 1 demonstrates the usage of IoT in different fields.

It is anticipated that by year 2025, 41.6 billion IoT devices will be interconnected, which poses many challenges for the practical realization of IoT [5]. Specifically in large IoT networks, where challenges related to the integrity and confidentiality of data exist. The number of security concerns, such as zero-day attacks aimed at internet users, has increased. As a result of the widespread use of the Internet in numerous nations, such as Australia and the USA, zero-day assaults had a considerable impact [6]. According to

✉ Awais Aziz Shah
awais.shah@poliba.it

Adeel Abbas
aabbas@cs.qau.edu.pk

Muazzam A. Khan
muazzam.khattak@qau.edu.pk

Shahid Latif
lshahid19@fudan.edu.cn

Maria Ajaz
mariaajaz@cs.qau.edu.pk

Jawad Ahmad
J.Ahmad@napier.ac.uk

[1] Department of Computer Science, Quaid-i-Azam University, Islamabad, Pakistan

[2] Pakistan Academy of Sciences, Islamabad, Pakistan

[3] School of Information Science and Engineering, Fudan University, Shanghai, China

[4] Department of Electrical and Informational Engineering (DEI), Politecnico di Bari, Bari, Italy

[5] School of Computing, Edinburgh Napier University, Edinburgh, UK

Published online: 30 August 2021

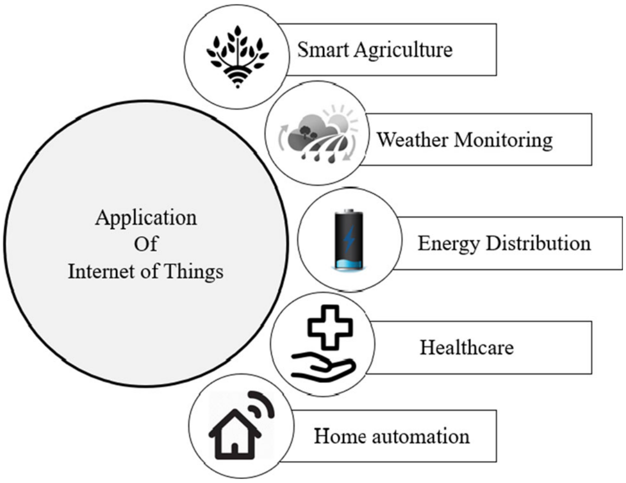

**Fig. 1** Applications of Internet of Things

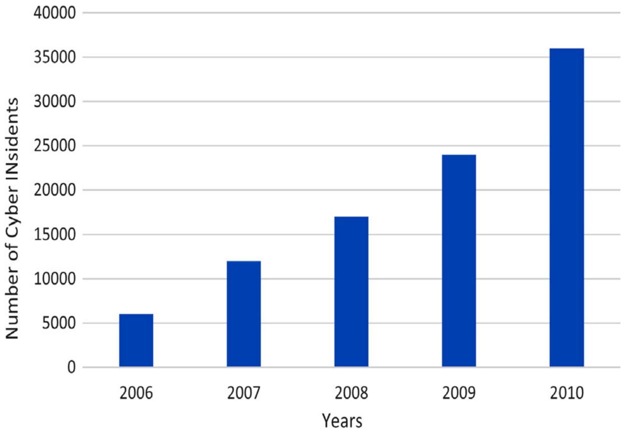

**Fig. 2** S-CERT: cyber incidents

United States Computer Emergency Readiness Team (US-CERT), cyber events have increased rapidly from 6000 to 36000 numbers in the year 2006 to the year 2010 duration [7]. Figure 2 (courtesy of US-CERT) depicts how cyber events have increased in the present internet network environment.

According to the statistics, the damages caused by the cyber attacks are expected to reach up to 3 Trillion by the year 2021 [8]. According to the Symantec report, on an average of IoT devices were attacked once after every two minutes [9]. Another analysis in [10] shows drastic increase in cyber-attacks incidents by approximately 2000% in just 6 years. In 2017, average costs caused by attacks reach to 482 million dollars in six months [10]. According to the 2017 data breach statistics, hackers have stolen or attacked about nine billion data records since 2013 [6].

Cyber-criminals from all around the world are driven to steal information, obtain unlawful profits, and discover new targets. To safeguard IoT devices or networks against assaults, it is critical to maintain a close check on them. It is crucial to analyze the sign of risks associated with IoT devices. Intrusions are the attempts to attack to the security structure of the IoT networks [11]. It can bypass the security layer of the network and become a major threat for the stability and confidentiality of the network [12].

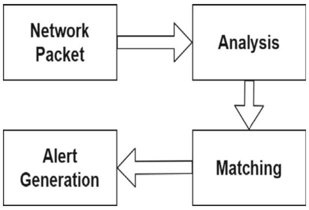

**Fig. 3** General architecture of an intrusion detection system

In order to secure the IoT systems, idea of intrusion detection was proposed in 1980 [13]. Intrusion detection is a process of monitoring and analyzing network traffic and respond when malicious attack occurs (also known as intrusions) with the signs on intrusion [14]. First intrusion detection system (IDS) was proposed in 1980 [15]. The purpose of IDS is to identify different types of harmful network traffic and computer activities that a regular firewall might miss. More precisely, we can say that IDS is very effective for detecting, identifying, and monitoring threats. This is critical for obtaining high levels of security against acts that jeopardize computer systems' availability, integrity, or secrecy [16].

Traditional approaches examine network packet by matching it with a predefined database where all types of attacks and signature patterns are already saved. The system was not sufficient for evaluating the traffic on the basis of this database, due to arrival of possible new zero-day, which will be distinct from signatures found in the file [9]. Existing IDS has shown inefficiency in detecting various attacks including zero-day attacks and reducing the false alarm rates (FARs) [17]. Therefore, it concludes that the IDS stability can be compromised due to malicious attacks no matter how accurate intrusion detection (ID) method. The IDS architecture is shown in Fig. 3.

IDS can be categorized based on how it is deployed or how it detects threats. There are several major categories of IDS [18,19].

- Detection-based ID methods
- Data-based ID methods
- Infrastructure-based ID methods
- Computing location-based ID methods
- Frequency usage-based ID Method

Five subcategories based on IDS traits have also been discovered in the literature [6]: statistical-based, pattern-based, rule-based, state-based and heuristic-based detection methods as shown in Table 1. Two major IDS categories,

**Table 1** Intrusion detection methodologies

| Methodology | Detail |
|---|---|
| Statistical | Analyzes traffic on the network using complicated statistical methods |
| Pattern | Recognizes the data characters, shapes and models |
| Rule | Detects a possible attack on suspicious network traffic using an attack "signature" |
| State | Review an event stream to detect probable attacks |
| Heuristic | Recognizes any aberrant activity that is not typical |

detection-based and data-based IDSs, are described in this study.

## 1.1 Detection-Based ID Methods

This technique is used to evaluate traffic on the basis of their attack type or their packet signature [20]. Detection-based IDS methods are functionally divided into three major categories (i.e., signature-based IDS, anomaly-based IDS and specification-based IDs) [7].

Signature intrusion detection systems (SIDS) are based on signature matching techniques to find a known attack. These are also known as rule-based detection or misuse detection [19,20]. In SIDS, signature- or pattern-matching methods are used to find a previous intrusion. For example, if 3 login attempts are failed in first 5 min, then alarm is generated for brute force password attack [19]. So, if there is a match found, an alarm will be generated. SIDS generally has a high detection accuracy for known intrusions and low false alarm rate (FAR) because an alarm is only generated if any pattern is matched [21]. On the other hand, it also requires frequent updates of signatures to ensure a good detection [22]. SIDS has several issues while identifying zero-day attacks since no matching signature exists in the database until the new attack's signature is retrieved and saved [6]. SIDS is resource-consuming approach due to huge signature database maintenance and comparison of possible intrusion [17].

Anomaly-based intrusion detection system (AIDS) has drawn interest from a lot of scholars due to its capacity to overcome the limitation of SIDS. The members' usual operations are profiled, and any divergence from the typical behavior is marked as an anomaly. This type of IDS is like a full-time job holder for detecting known and unknown attacks. Continuous checks are performed by the system for violations. If any case of violation or attempts exceeds from the threshold and if there is any deviation from baseline, data are notified as intrusion, and alarm is generated [23]. For this reason, it is also called behavior-based intrusion detection system [24]. AIDS has the capability to detect unknown or previously not encountered attacks because of its continuous learning ability [19]. AIDS is developed through two phases: training and testing. During the training stage, the typical traffic profile is utilized to learn the normal behavior model, followed by a fresh data set in the test phase to establish the ability of the system to generalize to unforeseen intrusions [6]. According to [25], AIDS gave better accuracy with low FAR and high false-positive rate. The continuous updation of profiles about attacks may increase the load on the system, which is a disadvantage of AIDS.

**Table 2** Comparison of signature- and anomaly-based IDS

| Signature-based IDS | Anomaly-based IDS |
|---|---|
| Identifies known attack types | Detects both known and unknown attack types |
| Depends upon operating system for identifying attacks | Less dependent upon operating system |
| Attack patterns and attack signatures should be updated regularly | Creates a profile of observed network communication for identification |
| Experts specified and programmed | Self-learning and self-programmed |
| Very effective in identifying intrusions with minimum false alarm rate (FAR) | Could be used to create intrusion signature and gave genuine intrusions |

AIDS is further divided into three categories [6,19].

- Statistical-based IDS
- Knowledge based IDS
- Machine learning-based IDS

Specification-based detection system (Sp-IDS) is responsible for process monitoring. It explains a system's intended behavior via its functions and the security policy [7]. Any operations or data packet carried out beyond the parameters of the system shall be regarded a security breach, and alarm will be generated. The cost and verification of defining the specifications are always remained a barrier for Sp-IDS. This concept was presented in 1996 [26]. Sp-IDS combines the advantages of SIDS and AIDS by manually developed specification and provides capability to detect previously unknown attacks with low FPR [19].

The comparison between IDS methods is illustrated in Table 2 [6,15].

## 1.2 Data-Based ID Methods

It also known as location-based IDS. Data-based IDS methods are divided into three main categories (i.e., host-based IDS, network-based IDS and hybrid-based IDS) [7,20].

The first category is network-based intrusion detection system (NIDS). It monitors the network traffic that is extracted from a network. This type of IDS is independent in operating system that is a reason they can be deployed in all types of environments [23]. These types of IDS can detect some specific attacks due to their monitoring capability. These IDSs have their specific network segment, and they only monitor those attacks which are passing through that segment to identify malicious activity such as denial of services (DoS) and brute force [27]. One of NIDS open-source example is SNORT [24].

Second category is host-based intrusion detection system (HIDS). This type of intrusion detection system has vast set of segments for monitoring. They can monitor the behavior of several objects of a host device [23]. can detect non-network traffic insider attack. Tripwire and AIDE (Advanced Intrusion Detection Environment) are examples of HIDS [6], which is one of its incapability to detect network attack types[15].

There are specific benefits and disadvantages of NIDS and HIDS. NIDS can be deployed easily and are less costly to buy and operate. Its performance nonetheless relies on familiar security features and signatures [7]. The system might simply fail to identify an attack if it uses a novel exploit that is ignorant of the IDS. HIDS is just as good as the security manager that keeps it up and monitors it. Therefore, the best optimal way is to combine a mixture of the best features of NIDS and HIDS to offer more flexibility [6]. This is generally known as hybrid IDS.

New and unknown attack types are main reason to improve IDS with technology of modern era. Machine learning has become a key solution for these types of problems.

In this study, six supervised ML techniques such as naive Bayes (Gaussian & multinomial), linear SVM, random forest (RF), logistic regression (LR), stochastic gradient descent and decision tree (DT), are deployed on CICIDS2017 dataset for individual performance on binary and multi-class classification. Comparison of four feature selection techniques has been done in this study. After individual performances, an ensemble model is proposed based on LR, naive Bayes (NB) and DT with voting classifier. These both binary and multi-class classifications will elaborate and differentiate that upcoming data packet is an attack or is a normal entry, and if it is any attack, then which type of attack is being happened. The proposed model provides significant improvement in accuracy and requires low computational power and resources.

### 1.3 Our Contributions

Main contributions of this work:

1. A novel ensemble-based learning-based ID model has been proposed.
2. Cross-comparison of several feature selection methods has been performed.
3. Performance of the proposed IDS has been evaluated for binary and multi-class classification scenarios.

### 1.4 Organization of Paper

The rest of the paper has organized as follows: Section 2 carries out the literature review of existing intrusion detection and ensemble learning methods. In Sect. 3, different methodologies of machine learning are described. The datasets used in this study with accuracy as an evaluation technique and proposed approach are briefly discussed in Sects. 4 and 5, respectively. Section 6 presents the results and discussion. Finally, Sect. 7 draws the conclusion of this work.

## 2 Literature Review

Due to the increase in the number of cyber attacks, the security of IoT devices is at high risk. The current state of the art proposes several solutions for the prevention of these attacks with the joint integration of machine learning techniques for the detection and identification of these attacks. This section discusses some of the work done in this direction.

An ensemble-based model for intrusion detection was established in [28] using multiple ML techniques of classification such as DT, J48 and SVM. Particle swarm optimization was used for selecting nine most relevant and important features in KDD99 dataset of intrusion detection. Proposed model's results produced higher accuracy of 90% with low FAR 0.9%.

Another hybrid IDS model based upon NB and SVM was presented in [29]. Real-time historical log dataset was normalized and preprocessing for this study. After enhancement, the proposed model produced 95% accuracy and precision. It is studied that classifier' performance was increased after adding session-based features.

A performance analysis of multiple classical ML algorithms on several ID-based datasets for detecting attack traffic has been performed in [30]. After normalization of datasets (CICIDS2018, UNSW-NB15, ISCX2012, NSLKDD and CIDDS001), three ML techniques such as SVM, KNN and DT were deployed. DT outperforms other classifiers by producing detecting accuracy rate between 99 and 100% for all datasets.

Another study of building an IDS using classification technique RF on NSL-KDD dataset is presented in [31]. Tree depth value was calculated by considering entropy score and Gini-index as z-score. Boruta technique was used for selecting important 34 important features from dataset. The proposed model [31] produced 99% accuracy for detecting attacks.

A lightweight IDS has been developed in [1] using SVM to detect unknown and misuse attempt in IoT network. This study conducted several experiments for DDoS attacks detection on different function such as linear, polynomial and radical basis. Processing time and complexity of SVM was reduced due to selected features as input. Main drawback of this proposed algorithm was the lack of ability to detect intrusions with zero effect of traffic flow rate.

A framework of machine learning-based botnet attack detection with sequential detection architecture for IDS has been introduced in [9]. Demand of processing resources reduced by adopting relevant feature selection method. N-BaIoT dataset was used in this study, and detection performance was 99% using three ML algorithms, including decision tree, NB and artificial neural network (ANN). Hybrid classification was used in each of sub-engine for achieving most accurate results among different classifiers. This classification gives an additional edge to extend detection mechanism with more sub-engines for new kind of attacks.

An ensemble-based AIDS model has been proposed in [32], which has DT, LR and gradient boosting as inputs of stacking classifier of ensemble learning. Chi-squared correlation method was deployed on CICIDS2018 dataset for extracting 23 important features. Proposed model produced 98.8% detection accuracy with 97.9% F-measure score and outperforms seven individual classifiers.

Anomaly detection system for cloud computing has been proposed in [14]. SVM is used as a prime machine learning algorithm with its different kernels. Important features of NSL-KDD dataset were selected on the basis of information gain ratio. The results show that the RBF kernel function gives the highest accuracy of 96.24% with minimum false alarm rate (FAR). Training and testing split was 80/20%. Study concludes that SVM has significant benefits for IDS evaluation on cloud computing.

A novel IDS with hybrid strategy on multi-agent system has been proposed in [33]. Deep neural network (DNN) was deployed for study around protocols of network and transport layer specially on transmission control protocol (TCP). Performance of DNN was investigated on training and detection agent. Proposed model was compared with different optimizers, Init_modes and activation functions on NSL-KDD dataset and got 98% performance for detecting anomalies and 97% for distinguishing different attack types.

Another study of building an IDS using classification technique RF on NSL-KDD dataset is presented in [31]. Tree depth value was calculated by considering entropy score and Gini-index as z-score. Boruta technique was used for selecting important 34 important features from dataset. The proposed model [31] produced 99% accuracy for detecting attacks.

An architectural model is presented in [12] for risk assessment (RA) of information system with CICIDS2017 dataset using ML algorithms. ML techniques including k-nearest neighbors (KNN), NB, gradient boosting tree, RF, and decision tree (DT) were evaluated for RA in this study. Performance of model was based on ML technique that have efficient predictivity of intrusion. Predictive model was the implementation of ML techniques that produced better results with CICIDS2017 dataset. For RA, risk matrix was analyzed by 15 model's predicted results.

A study presented in [34] proposed a model for detecting DDOS using ML algorithms. The performance of this model was analyzed on two datasets such as NSLKDD and KDD-Cup99 using DT and KNN classifiers. In this study, 8 features were extracted based on the approach of correlation. In this work, KNN outperformed DT with detection accuracy and error rate of 98.51% and 1.5%, respectively.

A performance analysis of ML algorithms in the context of anomaly-based intrusion detection in the field of IoT has been performed in [21]. Performance of different single and ensemble algorithms such as AdaBoost (AB), random forest (RF), multilayer perceptron (MLP) was compared for securing IoT from distributed denial-of-service (DDoS) attacks. Study aimed to identify the significance of a single classifier and that a classifier may perform significantly. The results revealed that XGB classifier shows good results for both classification and regression tree. Three popular data sets NSL-KDD, UNSW-NB15 and CIDDS-001 were used for benchmark in this study with Friedman and Nemenyi tests for statistical assessment and Raspberry Pi for calculating average response time.

A very comprehensive evaluation of the effectiveness of different ML algorithms including logistic regression (LR),NB, kNN, SVM, DT, and RF to detect MQTT (Message Queuing Telemetry Transport) protocol-based attacks on IoT has been conducted in [35]. An MQTT-based novel dataset was generated and then released for research community. This study also examines different needs of MQTT-based and other regular attack detection. Accuracy, true-positive (TP) and true negative (TN) metrics were used for fivefold cross validation to evaluate the experiments. Weighted average recall and precision rose up to 98.85% and 99.04%, respectively, for bidirectional flow feature as well as recall and precision rose up to 93.77% and 97.19%, respectively, for unidirectional flow feature. This study concludes that similar characteristics give flow-based features an upper hand to discriminate between human entry (Benign) and MQTT-based attack.

The basic need of feature selection is discussed in [36] by proposing an IDS of detecting DOS attacks using ML techniques such as NB, KNN, RF and SVM. Different sets of features such as 11, 12, 13 and 15 were extracted by multiple feature selection techniques. Experimental results prove that accuracy improves after reducing features of any dataset. In this study, RF outperforms other algorithms in terms of better results with 99.63% of accuracy.

Another study [37] also presented an ensemble IDS model having KNN, extreme learning machine and hierarchical extreme learning machine techniques. Proposed model produced 84.29% of detection accuracy with the 77.18% rate of detecting zero-day attacks. The study presented in [38] also evaluates four machine learning algorithms named as RF, decision tree C5.0, naive Bayes (NB) and support vector machine (SVM) on Canadian Institute for Cybersecurity Intrusion Detection System dataset (CICIDS2017). Detection of DDoS attacks and finding better performer ML algorithm were the basic needs of this study. Success probability of 99% with the average accuracy of 86.80% and 96.45% of RF and C5.0, respectively, surprises the others. They find out SVM was incorrectly classifying with 75% of false-positive rate (FPR). Algorithmic complexity was based upon number of features and number of training samples.

A review of 16 research methodologies for finding out most relevant and updated dataset and method for NIDS has been carried out in [11]. The pros and cons of existing methodologies has been discussed in detail here and has been concluded that the recent method by [39] of distance-based ML techniques including KNN, k-means clustering on CIDDS-001 dataset provides better results. Furthermore, [11] states that live or online data captured from real-time networks can give more accurate results.

The shortcomings of the available datasets for IDS developed since 1998 (such as unreliability, lack of traffic diversity and metadata) have been discussed in [40]. The study specifically focuses on CIC-IDS2017 effectiveness and feature selection using machine learning for detecting attack types. Moreover, the study also defines the concept of superfeatures using reduction algorithm. Seven classification algorithms including RF, decision tree (ID3), AdaBoost, MLP, NB, KNN, and quadratic discriminant analysis (QDA) were compared. It was concluded that random forest algorithm outperformed with superfeatures as compared to individual and top selected features.

Performance evaluation of Bayesian network and RandomTree classifiers is conducted in [25] with ensemble learning method vote. Ensemble IDS model is evaluated on KDDcup99 dataset and compared with base classifiers in terms of accuracy, precision and recall. This study concludes that proposed model has better effect on precision and recall instead of accuracy rate and claims that IDS presents a good effect for the whole dataset whatever big sample or small sample because of combined advantages of aforementioned classifiers. Bayesian network has an advantage of better effects on small datasets, while RandomTree performs better with big sample data.

In this study, several techniques are deployed for intrusion detection with multiple combinations of classification algorithms as mentioned before. Some ML algorithms provide better results with higher FPR, which is not bearable for any IDS. Additionally, the existing models consume high computational power and require expensive resources while deploying MLP, ANN, DNN and DL in comparison with ML techniques for better IDS system. These advanced techniques gave better results but utilize maximum resources to establish hidden layers and hidden units. As we increase the hidden layers or over-train the system, results will be optimum, but there will be overfitting issues in structure [41]. It is clear that there is a need for more effective models to cope with the future challenges of cyber security within the IoT domain. Ensemble learning can boost the performance of ML-based IDS [42]. According to [43], hybrid or ensemble models provide higher accuracy of detection and lower false alarm rate (FAR).

## 3 Methodologies

On the basis of literature, it was found that DL and ANN require substantial computing power for execution with multiple hidden layers. Moreover, DL might gave optimum/better results when hidden layers are increased but meanwhile, DL is complex in nature. This makes the systems fragile, and when errors are made, the errors can be very large [44]. Due to over-training, number of hidden layers increased when compared to the problem's complexity. This situation effects time & complexity. Moreover, it also effects resources very badly as well as loses its ability of generalization over testing dataset [41].

As discussed above, IDS can be categorized into two main detection systems: AIDS and SIDS. These two have several benefits mainly related to detecting the behavior of network packet, but there are several shortcomings of such detection system as well. SIDS uses detail knowledge of attacker's actions. Common signatures can improve the accuracy of SIDS and also gave limited number of false-positive alarm rate [45]. On the other hand, it also needs regular signature updates to ensure accurate detection, and it is a resource-intensive technique owing to the large signature database maintenance and comparison of potential intrusions [22]. AIDS can detect zero-day attacks with low false alarm rate, but its result will have high false-positive rate. Statistical-based, knowledge-based, and several ML algorithms such as fuzzy logic, SVM, NN, Markov models are used to enhance detection model's performance [46]. AIDS has the capability to overcome the limitation of SIDS. ML models are updated, in order to improve the IDS performance.

We have found in the literature that single classifier may not be strong enough to build a good AIDS model due to large and imbalanced data. The constraints of the use of a single AIDS classifier lead to the notion of constructing a more sophisticated, but less accurate and low FAR hybrid or ensemble model [43]. Whenever a hybrid or ensemble approach is introduced, the performance of individual algorithms can be enhanced, and some studies have been demonstrated that the application of ensemble paradigm can prove to be versatile and certainly boost the prediction accuracy and detection speed. With a proper voting system and weighting assignment, this approach seems to improve the classification rate [22]. With the help of an ensemble model, we can reduce the uncertainty in the generalization performance of using a single algorithm [43]. These are the reasons to choose "ensemble" approach for enhancing individual performance of ML classifiers in an AIDS model.

Ensemble learning is not limited to ML basic classification algorithms; it can also help to improve the performance of ANN, DNN and MLP. For example, in [33], authors deployed DNN for a hybrid classification through ensemble method.

In this study, we are working on a classification problem, and the dataset used is known as CICIDS2017. Six different supervised ML classification techniques for intrusion detection are chosen, in this work. Decision tree (DT), naive Bayes (NB), Gaussian & multinomial, random forest (RF), logistic regression (LR), linear SVM and stochastic gradient descent classifier (SGDClassifier) are the algorithms used with stacking classifier as an ensemble method. These six ML algorithms are chosen on the basis of optimum performance in the literature as discussed below.

Mirza [42] mentioned that LR and DT (CART) perform better with artificial neural network (ANN) in an ensemble model for intrusion detection at different thresholds because of less loss function value of LR. Decision tree performs better due to pre-pruning method, and stacking classifier was deployed as ensemble method. Yang [9] states that NB performed better with J48 and ANN in an ensemble algorithm because it identify labels faster. NB's accuracy improves from 62.52% to 99.10% for junk attack detection. Kelton [47] states that SVM with linear kernel performs better with 92% individual accuracy for forming an ensemble IDS model with RF and multinomial. SVM performs better with its kernel trick. S. Krishnaveni [14] also stated that linear SVM performs better with 92.65% accuracy and 5.92s time with less false alarm rate for anomaly detection IDS. Le Yang [13] states after experiments that RF is performing better with selected features in comparison of SVM and k-nearest neighbor (Knn).

Thus, these aforementioned six ML classifiers with one ensemble method are being used in this study for performing their individual and hybrid analysis in terms of intrusion detection.

### 3.1 Decision Tree

Decision tree is another ML algorithm. As per its name, it is a tree structure classifier consisting of two parts of decision leaves and nodes and breakdowns the data into smaller and smaller nodes. Leaves are the decision outcomes. It can be used for classification and regression problems [48]. Entropy is the measure of impurity or uncertainty data samples. It can be calculated as [48]:

$$H(S)=\sum_{y\in X} p(y)\log 2\frac{1}{p(y)} \tag{1}$$

In Eq. 2, information gain IG(S, A) for a set S is changed in entropy in particular feature A. Entropy and IG are calculated for which feature to split his nodes on to get closer to predicting target variable. It also tells when to stop splitting [48].

$$\mathrm{IG}(S,A)=H(S)-\sum_{i=0}^{n} P(y)\times H(y) \tag{2}$$

ID3 and C4.5 algorithms are usually used for building DT [49].

### 3.2 Naive Bayes

The naive Bayes (NB) classifier is the most common algorithm of ML, which was based on Bayesian theorem for classification problems [9]. Learning probabilistic knowledge from available features and using it for unknown features are its basic deeds. It also handles nonlinear parameters and usually robust to outliers. Bayes theorem states that [50]:

$$P(a\mid D)=\frac{P(D\mid a)P(a))}{P(D)} \tag{3}$$

In Eq. 3:

- $P\ (a) \longrightarrow$ hypothesis prior probability.
- $P\ (D) \longrightarrow$ data prior probability.
- $P\ (a\mid D) \longrightarrow$ Probability of hypothesis given data posterior probability
- $P\ (D\mid a) \longrightarrow$ Probability of data given hypothesis likelihood

With the number of classes a$_i$, where i = 1,......L. Probability of seeing D belonging to a$_i$ can be written as $P\ (D\mid a_i)$. The posterior probability of class a$_i$ can be calculated as [50]:

$$\begin{aligned} P\ (a_i\mid D)\ &= \frac{P\ (D\mid a_i)\ P\ (a_i)}{P(D)} \\ &= \frac{P\ (D\mid a_i)\ P\ (a_i)}{\sum_{i=1}^{L} P\ (D\mid a_i)\ P\ (a_i)} \end{aligned} \tag{4}$$

The NB classifier is based on the assumptions that attributes are independent on given target class [50].

$$v_{NB} = \text{argmax}_{v_j \in V} P\left(v_j\right) \prod_i P\left(a_i \mid v_j\right) \tag{5}$$

### 3.3 Random Forest

It is an ensemble algorithm introduced by Leo Breiman [13]. He integrated decision tree and bagging method for developing forest of decision trees (DT). These tree are created by random selection of attributes for separation at each node. Overfitting problem of DT is resolved by this ensemble algorithm. Sample sizes are extracted by the bootstrap method from original data set [13].

### 3.4 Logistic Regression

It is a classification algorithm and a variant of linear regression. It predicts the binary outputs. Logistic curve is produced, which is limited to values between 0 and 1. Curve is constructed using odds logarithm of target variable instead of probability. Sigmoid function is used in LR [42]. It is extended for multi-class classification as well with OVR and multinomial attributes. Overfitting can be faced with large-dimensional dataset but can be avoided by regularization methods. After taking log of odds ratio, LR can handle both categorical and continuous data with equation 6 [42]:

$$p = \frac{1}{1 + e^{-\left(b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_p x_p\right)}} \tag{6}$$

### 3.5 Linear SVM

SVM was developed from statistical learning theory concepts in 1970 [1]. Basically, it deals with two-class classification problems and regression. Hyper-plane creates a boundary between two classes for classification. Nearest point to the hyper-plane are called support vectors, and its technique is known as support vector machine (SVM). In Eq. 7, hyper-plane is expressed [1]:

$$w \cdot y + b = 0 \tag{7}$$

where y is an input vector, w and b represent its weight and bias, respectively. Equation 8 is a mathematical representation of SVM [1]:

$$h\left(x_i\right) = \begin{cases} +1 & \text{if } w \cdot y + b \geq 0 \\ -1 & \text{if } w \cdot y + b < 0 \end{cases} \tag{8}$$

Here, +1 and -1 represent classes A and B, respectively. Final decision equation is as follows [1]:

$$f(x) = \text{sign}\left(\sum_{i=1}^{N} \alpha_{o,i}\left(y^T y_i\right) + b\right) \tag{9}$$

Linear kernel of SVM is used when the data are linearly separable, that's why, it can be separated using a single line and preferable for large number of features. Final decision equation can be modified with the required kernel formula [1]. Linear kernel function is expressed as follows:

$$\text{Linearkernel} = y^T y_i \tag{10}$$

### 3.6 Stochastic Gradient Descent Classifier

Stochastic gradient descent (SGD) is performing well for large and sparse problems due to its linear complexity. This approach is used to fit linear classifiers such as SVM and LR under convex loss functions. Mathematical representation is as follows [51]: Let $(a_i, b_i)$ be a set of training instances, $a_i$ belongs to $Z^n$, $b_i$ belongs to -1, 1. The output of the classification is got by:

$$c(x) = v^u + j \tag{11}$$

where v belongs to $Z^m$ and j being the intercept which belongs to Z. The cost function will be

$$F(v, j) = \frac{1}{N} \sum_{i=1}^{N} L\left(b_i, c\left(a_i\right)\right) + \alpha Z(v) \tag{12}$$

where L represents loss function, Z represents regularization and $\alpha > 0$. L can take three types of values hinge, log and modified_huber for SVM, LR and smooth hinge loss.

SGD is not a ML classifier but scikit-learn API allows SGDClassifier to act as an estimator with modified_huber loss function. The below equation represents elasticnet regularization term of SGDClassifier.

$$\frac{\rho}{2} \sum_{i=1}^{n} v_i^2 + (1 - \rho) \sum_{i=1}^{n} |v_i| \tag{13}$$

where $p$ signifies ratio term [51].

### 3.7 Stacking Classifier

Stacking or voting classifier is a meta-classifier or ensemble learning method. Ensemble methods improve the performance of model [52]. Voting classifier combines different ML classifiers for classification.

**Table 3** Dataset labels

| BENIGN | BOT | DDoS |
|---|---|---|
| DoS GoldenEye | DoS Hulk | DoS Slowhttptest |
| DoS slowloris | FTP-Patator | Heartbleed |
| Infiltration | PortScan | SSH-Patator |
| Brute force | Sql injection | XSS |

Let us assume decision of the $t$th classifier as $d_{t,c} \in \{0,1\}$, t = 1, . . . , T and c = 1, . . . , C, where T is the number of classifiers and C is the number of classes. Hard voting is one of voting methods, and it has three scenarios, depending on unanimous voting, simple majority and plurality voting. Hard voting usually refers to plurality voting. A mathematical representation is as follows [52]:

$$\sum_{t=1}^{T} d_{t,c^*} = \max_{c} \sum_{t=1}^{T} d_{t,c} \tag{14}$$

Majority vote can be weighted by associating a weight $W_t$ to classifier $h_t$ for choosing c* class. Mathematical representation is given below [52].

$$\sum_{t=1}^{T} w_t d_{t,c^*} = \max_{c} \sum_{t=1}^{T} w_t d_{t,c} \tag{15}$$

Another voting mechanism is available called soft voting that works with a probability term. It takes average probabilities for each class and utilizes it for classifying data points.

## 4 Dataset

The dataset used in this study has been created by The Canadian Institute for Cybersecurity (CIC) in 2017 [53]. The CIC Intrusion Detection System dataset (CICIDS2017) contains common attacks, which are similar to the real-world data. This dataset consists of two files named: GeneratedLabelledFlows and MachineLearningCVE, the first file consists of 86 features, while the later consists of 79 features [54]. In this study, MachineLearningCSV data file consisting 8 traffic monitoring sessions of 5 days is implemented. These 8 files are merged into 1 CSV file for further study. Merged file has 2830743 rows, 78 features columns and 1 label column. Two features from this file have the same name Fwd Header Length that makes it as redundant feature; then, one of them is removed and only 77 features columns and 1 label column are available for experiments [55]. These 78 features contain 15 class labels; Table 3 represents 1 BENIGN (normal) and 14 Attack type labels [56]. Dataset is splitted into two portions: 70% for training and 30% for testing.

### 4.1 Accuracy

Accuracy is used to monitor the performance of the proposed approach. Accuracy determines the real performance, which allows to see the correct detection for different instances. The following equation states the general formula for calculating accuracy. The higher the accuracy is, the better the ML technique is [57].

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \times 100 \tag{16}$$

In the above equation

- TP stands for true positives, data points that have been accurately classified as normal.
- TN stands for true negative, data points that have been accurately classified as attack.
- FP stands for false positive, normal data points that have been incorrectly categorized as attack.
- FN stands for false negative, attack data points that have been incorrectly categorized as normal [57].

## 5 Proposed Model

Before deploying any ML technique, preprocessing of dataset is necessary. During preprocessing, 2867 rows consist of NAN and infinity values, which were removed. Then, the resultant dataset consists of 2827876 rows [58]. The work flow of the proposed IDS approach is described in Fig. 4. After preprocessing of the dataset, binary and multi-class classification by feature selection methods as well as without feature selection methods is performed. In our proposed model, four feature selection techniques were used. One of the optimum feature selection techniques was opted for comparison with all feature's results in both classification scenarios. On the basis of results, three optimum ML algorithms were selected for ensemble model. At the end, those 3 algorithms were deployed with stacking classifier for ensemble model, and resultant accuracy in both scenarios matches the literature work with less resources and FAR.

### 5.1 Multi-Class Classification

All the aforementioned algorithms were applied for multi-class classification with all 78 features of dataset. CICIDS2017 dataset contains 15 class labels. ML algorithms were applied on dataset for classifying these all 15 different types of labels. Multi-class classification will identify the exact type of attack.

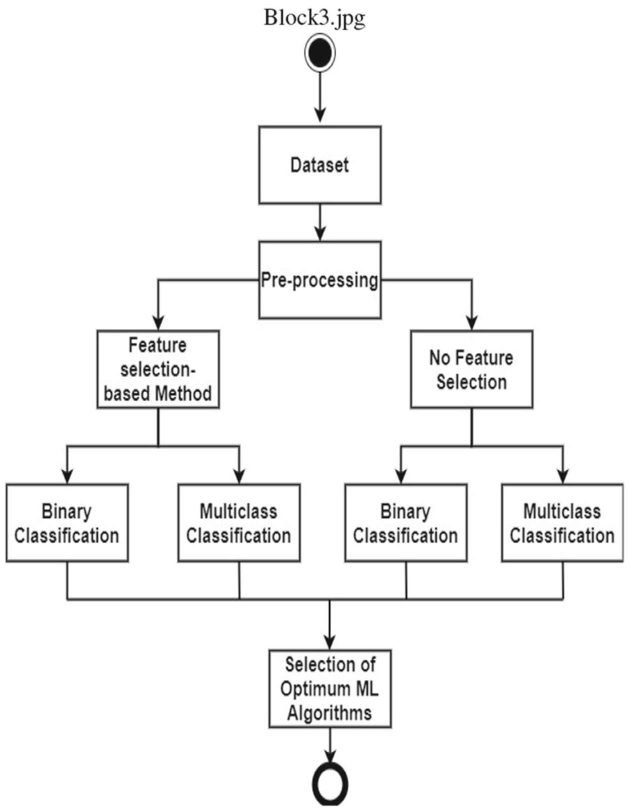
Block3.jpg
Dataset
Pre-processing
Feature selection-based Method
No Feature Selection
Binary Classification
Multiclass Classification
Binary Classification
Multiclass Classification
Selection of Optimum ML Algorithms

**Fig. 4** Flowchart of intrusion detection process

**Table 4** Individual performance on multi-class classification

| Classifiers | Accuracy (%) |
|---|---|
| DT | 91.22 |
| NB(G) | 80.26 |
| NB(M) | 84.89 |
| RF | 99.68 |
| LR | 91.56 |
| LinearSVM | 92.85 |
| SGDClassifier | 92.99 |

Table 4 shows accuracy results, and one can see that RF is performing better with 99.68% accuracy of detection. From the table, it is evident that NB Gaussian (NB(G)) performance is poor since it does not work for multi-class. The performance of NB multinomial (NB(M)) is also weak against DT and linear SVM. LR has minimum accuracy from all linear classifiers (LinearSVM and SGDClassifier), which uses SGD optimizer with 91.56% accuracy. Comparison of accuracy results for multi-class classification with all features is given in Table 4.

**Table 5** Individual performance on binary classification

| Classifiers | Accuracy (%) |
|---|---|
| DT | 98.68 |
| NB(G) | 80.65 |
| NB(M) | 85.34 |
| RF | 99.67 |
| LR | 92.45 |
| inearSVM | 89.61 |
| GDClassifier | 92.26 |

## 5.2 Binary Classification

For improving classification model, all attack types are replaced with one class label Attack and the dataset is converted into binary class. Now, we have two class labels BENIGN and ATTACK.

After conversion, binary classification algorithms were applied on all features of same dataset. LR increases detection accuracy in comparison with SGD optimization holder classifiers because of its basic binary classification nature. Accuracy comparison of the ML technique results is given in Table 5. The accuracy of DT increases from 91.22 to 98.68% because of less targeted values. NB(G) has least binary detection accuracy results in comparison with other ML techniques.

## 5.3 Feature Selection and Classification

Feature selection is a process of removal of redundant or useless features from the initial dataset. It decreases the number of dimensions in the dataset, reducing processing and memory utilization, making it easier to understand and examine data [59]. The most common feature selection methods are filtering, wrapper, embedding, and hybrid methods [60].

With large datasets, overfitting is a common problem, which can be overcome by the regularization method. In this study, we have analyzed four methods for important feature selection, and one of them is used for further classification process. CICIDS2017 data has numerical input and categorical targeted output. Wrapper method was not deployed in this study because it is slower than others [60]. Therefore, ANOVA (analysis of variance) correlation method and Chi-squared correlation methods from filtering and feature importance of random forest and features importance of LinearSVM from embedded or hybrid method are chosen. This study extracted 15 important features from whole dataset on the basis of their relevance by these aforementioned 4 techniques for deep understanding of dataset. ANOVA uses F-test to confirm any significant difference between the groups. ANOVA's f-test value will be 1, when there will not be any

**Table 6** Top 15 features extracted for multi-class classification

| Chi-2 | ANOVA | RF | LinearSVM |
|---|---|---|---|
| Flow duration | Bwd packet length max | Destination port | Destination port |
| Bwd packet length max | Bwd packet length mean | Fwd packet length max | Fwd packet length max |
| Bwd packet length mean | Bwd packet length Std | Fwd packet length mean | Fwd packet length min |
| Bwd packet length Std | Flow IAT max | Bwd packet length max | Fwd packet length mean |
| Flow IAT max | Fwd IAT Std | Bwd packet length mean | Fwd packet length Std |
| Fwd IAT total | Fwd IAT max | Bwd packet length Std | Flow IAT Std |
| Fwd IAT Std | Max packet length | Fwd IAT Std | Flow IAT max |
| Fwd IAT max | Packet length mean | Max packet length | Fwd IAT max |
| Packet length std | Packet length std | Packet length mean | Fwd packets/s |
| FIN flag count | Packet length variance | Packet length std | Min packet length |
| PSH flag count | Average packet size | Packet length variance | Packet length mean |
| Avg Bwd segment size | Avg Bwd segment size | Average packet size | packet length variance |
| Idle mean | Idle mean | Avg Bwd segment size | Down/up ratio |
| Idle max | Idle max | Subflow Fwd bytes | Average packet size |
| Idle min | Idle min | Init_Win_bytes forward | Idle max |

**Table 7** Top 15 features extracted for binary classification

| Chi-2 | ANOVA | RF | LinearSVM |
|---|---|---|---|
| Bwd packet length max | Bwd packet length max | Destination port | Fwd packet length max |
| Bwd packet length mean | Bwd packet length mean | Fwd packet length max | Fwd packet length Std |
| Bwd packet length Std | Bwd packet length Std | Fwd packet length mean | Flow bytes/s |
| Flow IAT max | Flow IAT max | Bwd packet length | Max flow IAT Std |
| Fwd IAT Std | Fwd IAT Std | Bwd packet length mean | Flow IAT max |
| Fwd IAT max | Fwd IAT max | Bwd packet length Std | Flow IAT min |
| Max packet length | Max packet length | Max packet length | Fwd IAT mean |
| Packet length mean | Packet length mean | Packet length mean | Fwd IAT max |
| Packet length Std | Packet length Std | Packet length Std | Fwd IAT min |
| Packet length variance | Packet length variance | Packet length variance | Max packet length |
| FIN flag count | Average packet size | Average packet size | Packet length mean |
| Avg Bwd segment size | Avg Bwd segment size Avg | Fwd segment size | Packet length variance |
| Idle mean | Idle mean | Avg Bwd segment size | Down/up ratio |
| Idle max | Idle max | Init_Win_bytes forward | Average packet size |
| Idle min | Idle min | Init_Win_bytes backward | Idle max |

significant difference between the groups. It shows that all variances are equal [61]. Chi-squared statistic is calculated for selecting features, which are highly dependent on the response in Chi-squared correlation method. Tables 6 and 7 represent top 15 relevant features extracted by four feature selection techniques for multi-class and binary classification scenarios, respectively.

## 5.4 Classification on Selected Features

Table 8 illustrates the details about the accuracy of ML algorithm on selected features for binary classification. ANOVA and Chi-squared (Chi-2) feature selection methods have good accuracy results for NB(M), but F-measure score was just 72%; therefore, NB(M) is ignored. Average accuracy results of LinearSVM feature selection method were better in comparison with other three methods. Then, top 15 features were extracted from LinearSVM method with average accuracy of 88.19%, which were chosen for binary classification. Selected features for further study are represented in Table 7 (see LinearSVM column).

ML techniques were applied on selected feature's dataset for multi-class classification. Table 9 illustrates that LinearSVM feature selection method's average accuracy is 85.56%, which outperforms the other selected methods on top 15 features for multi-class classification. Experimental

**Table 8** Binary class accuracy performances at top 15 features with 4 feature selection models

| | ANOVA (%) | CHI-2 (%) | RF (%) | LinearSVM (%) |
|---|---|---|---|---|
| DT | 83.85 | 90.72 | 90.47 | 88.45 |
| NB(G) | 83.74 | 82.78 | 85.23 | 85.24 |
| NB(M) | 80.32 | 80.32 | 87.97 | 81.33 |
| RF | 89.48 | 90.29 | 91.95 | 89.93 |
| LR | 87.99 | 87.86 | 88.03 | 88.58 |
| LinearSVM | 87.96 | 87.91 | 83.64 | 88.45 |
| SGDClassifier | 87.98 | 87.99 | 87.92 | 88.49 |
| Avg results | 86.83 | 87.21 | 87.87 | 88.19 |

**Table 9** Multi-class accuracy performances at top 15 features with four feature selection models

| | ANOVA (%) | CHI-2 (%) | RF (%) | LinearSVM (%) |
|---|---|---|---|---|
| DT | 81.17 | 93.36 | 88.97 | 93@.50 |
| NB(G) | 15.01 | 10.36 | 64.24 | 64.08 |
| NB(M) | 80.32 | 84.66 | 82.32 | 83.92 |
| RF | 88.81 | 89.92 | 97.46 | 93.51 |
| LR | 87.01 | 85.45 | 86.26 | 87.60 |
| LinearSVM | 86.25 | 85.50 | 85.72 | 88.93 |
| SGDClassifier | 87.01 | 86.58 | 87.00 | 87.40 |
| Avg results | 75.08 | 76.55 | 84.57 | 85.56 |

**Table 10** Comparison of multi-class and binary class accuracy with all and selected features (M = Multi-class, B = Binary-class)

| | M(All) (%) | M(Selected) (%) | B(All) (%) | B(Selected) (%) |
|---|---|---|---|---|
| DT | 91.22 | 93.50 | 98.68 | 88.45 |
| NB(G) | 80.26 | 64.08 | 80.65 | 85.24 |
| NB(M) | 84.89 | 83.92 | 85.34 | 81.33 |
| RF | 99.68 | 93.51 | 99.67 | 89.93 |
| LR | 92.99 | 87.61 | 92.45 | 88.58 |
| LinearSVM | 92.85 | 88.93 | 89.61 | 88.45 |
| SGDClassifier | 91.56 | 87.40 | 92.26 | 88.49 |

results prove that linear SVM performs better for feature selection in multi-class scenario also. Selected features for further study are represented in Table 6 (see LinearSVM column).

### 5.5 Ensemble Method

Table 10 presents the comparison of multi-class and binary class with all and selected features. In multi-class with all and selected features scenario, accuracy is decreasing for all classifiers except decision tree. Accuracy of DT is increasing from 91.22 to 93.50%. DT is targeted for our proposed ensemble model.

In binary classification with all and selected features scenario, detection accuracy of all classifiers is decreasing same as multi-class scenario, but NB(Gaussian) accuracy is increasing from 80.65 to 85.24%. Therefore, NB is also chosen for ensemble model. Highlighted values in Table 10 are representing increasing accuracy of detection. Linear SVM cannot be selected for ensemble model because it has been already used for feature selection. RF is an ensemble algorithm of DT, which is already selected for ensemble model, and therefore, it does not make sense to select RF. LR is preferred on SGDClassifier (an optimization method of linear classifiers) because LR works on SGD method.

Our proposed model in this study consists of three supervised classification algorithms such as decision tree, naive Bayes and logistic regression. Stacking classifier is used for ensemble learning with hard voting. All the results of these three classifiers for selected features are feed in the hard voting module, which out-turns efficient and more accurate predictions. The proposed model is illustrated in Fig. 5.

The proposed ensemble model distinctly increases the accuracy of detecting actual label in multi-class selected feature scenario. DT classifier helped to increase the hybrid accuracy of NB(M) and LR to 88.96% from 83.92% and

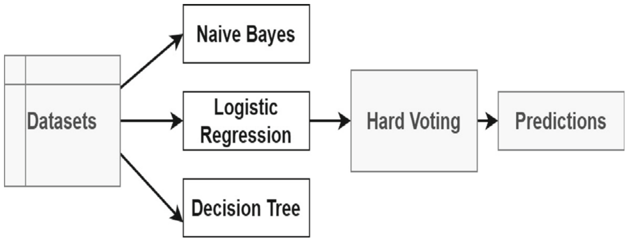

**Fig. 5** The proposed model

**Table 11** The proposed model's results

| Classification | Classifiers | Accuracy | Proposed model accuracy |
|---|---|---|---|
| Multi-class | NB(M) | 83.92% | **88.96%** |
| | DT | 93.50% | |
| | LR | 87.60% | |
| Binary-class | NB(G) | 85.24% | **88.92%** |
| | DT | 88.45% | |
| | LR | 88.58% | |

87.60%, respectively. In case of binary classification, our ensemble model increased the overall accuracy to 88.92% for selected features as given in Table 11.

The individual accuracy of each class label is represented in Table 12 for multi-class classification scenario. Table 12 shows that our proposed model is producing better accuracy for detecting several attack types.

## 6 Results Conclusion

After prepossessing of CICIDS2017 dataset, six ML algorithms such as DT, NB(G), NB(M), RF, LR, LinearSVM and SGDClassifier were deployed to all features where RF outperforms other techniques with average accuracy of 99.67% in both classification scenarios. For regularization, four feature selection techniques were applied to extract top 15 features. Linear SVM method of feature selection outperforms other techniques in both binary and multi-class scenarios with average accuracy of 88.19% and 85.56%, respectively. With the prominent improvement of accuracy, 3 classifiers are selected such as NB, DT and LR with accuracy of 83.93%, 93.50% and 87.60%, respectively, in multi-class. In binary classification, respective three algorithms outperform others with 85.24%, 88.45% and 88.58% accuracy, respectively. Table 11 elaborates the results of voting ensembles method where the accuracy of NB and LR increases due to regularization of DT and majority voting scheme of stacking classifier.

Ustebay [10] worked on single merged traffic file of same dataset CICIDS2017 using deep learning classifier “Deep Multilayer Perceptron (DMLP)” for intrusion detection resultant in 89% accuracy with selected features. This study provides better performance in terms of accuracy for just binary classification scenario and required more computational power and hardware cost due to several hidden layers of DL. In comparison with [10] work, our proposed approach uses complete dataset of 8 traffic files and produces average accuracy of 88.94% approximately equivalent to his work, with low computational power and resource using ML algorithms instead of DL. Our proposed model performs better to distinguish between benign and attack as well as benign and which type of attack in binary and multi-class classification scenarios, respectively.

**Table 12** Individual accuracy of all 15 labels in multi-class classification

| Labels | Accuracy (%) |
|---|---|
| BENIGN | 99.98 |
| Bot | 100 |
| DDoS | 59.85 |
| DoS goldenEye | 93.25 |
| DoS hulk | 71.67 |
| DoS slowhttptest | 82.54 |
| DoS slowloris | 79.26 |
| FTP-Patator | 100 |
| Heartbleed | 100 |
| Infiltration | 100 |
| PortScan | 99.99 |
| SSH-patator | 100 |
| Brute force | 99.78 |
| Sql injection | 100 |
| XSS | 100 |

## 7 Conclusion

This paper presents an ensemble learning-based intrusion detection model. Proposed model gives guarantee to detect all types of attacks. It provides significant accuracy with low computational power, resources and low false alarm rate by using ML algorithms instead of ANN and DL techniques with ensemble paradigm. Proposed model consists of LR, NB and DT with hard voting ensemble method and evaluated on CICIDS2017 dataset in both binary and multi-class scenarios.

In future, proposed ensemble model will be extended on deep and recurrent neural network with objective to increase the accuracy for detecting the intrusions in IoT.

**Funding** Open access funding provided by Politecnico di Bari within the CRUI-CARE Agreement.

## References

1. Jan, S.U.; Ahmed, S.; Shakhov, V.; Koo, I.: Toward a lightweight intrusion detection system for the internet of things. IEEE Access **7**, 42 (2019)
2. Nivaashini, M.; Thangaraj, P.: A framework of novel feature set extraction based intrusion detection system for internet of things using hybrid machine learning algorithms. In: 2018 International conference on computing, power and communication technologies (GUCON). pp. 44–49 (2018)
3. Tait, K.-A.; Khan, J. S.; Alqahtani, F.; Shah, A. A.; Khan, F. A.; Rehman, M. U.; Boulila, W.; Ahmad, J.: Intrusion detection using machine learning techniques: an experimental comparison. In: IEEE International congress of advanced technology and engineering (ICOTEN)
4. Khan, M.A.; Khan, M.A.; Latif, S.; Shah, A.A.; Rehman, M.U.; Boulila, W.; Driss, M.; Ahmad, J.: Voting classifier-based intrusion detection for IOT networks. In: 2nd International conference of advanced computing and informatics (ICACIN) (2021)
5. Abiodun, O.I.; Abiodun, E.O.; Alawida, M.; Alkhawaldeh, R.S.; Arshad, H.: A review on the security of the internet of things: challenges and solutions. Wireless Pers. Commun. (2021). https://doi.org/10.1007/s11277-021-08348-9.
6. Khraisat, A.; Gondal, I.; Vamplew, P.; Kamruzzaman, J.: Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity **2**, 12 (2019)
7. Rajasekaran, K.: Classification and importance of intrusion detection system. Int. J. Comput. Sci. Inf. Secur. **10**, 44 (2020)
8. Thakkar, A.; Lohiya, R.: A review of the advancement in intrusion detection datasets, In: Procedia Computer Science. **167**, pp. 636–645, international Conference on Computational Intelligence and Data Science (2020)
9. Soe, Y.N.; Feng, Y.; Santosa, P.I.; Hartanto, R.; Sakurai, K.: Machine learning-based iot-botnet attack detection with sequential architecture. Sensors **20**(16), 4372 (2020)
10. Ustebay, S.; Turgut, Z.; Aydin, M.A.; Intrusion detection system with recursive feature elimination by using random forest and deep learning classifier. In : International Congress on Big Data. Deep learning and fighting cyber terrorism (IBIGDELFT) **2018**, 71–76 (2018)
11. Rupa Devi, T.; Badugu, S.: A review on network intrusion detection system using machine learning. In: Satapathy, S.C., Raju, K.S., Shyamala, K., Krishna, D.R., Favorskaya, M.N. (eds.) Advances in Decision Sciences, Image Processing Security and Computer Vision. Springer, Cham (2020)
12. Pangsuban, P.; Wannapiroon, P.: A real-time risk assessment for information system with cicids2017 dataset using machine learning. Int. J. Machine Learn. Comput. **10**(3), 465–470 (2020)
13. Yang, L.; Cai, M.; Duan, Y.; Yang, X.: Intrusion detection based on approximate information entropy for random forest classification. In: Proceedings of the 2019 4th international conference on big data and computing. ser. ICBDC 2019. New York, NY, USA: Association for Computing Machinery, p. 125-129 (2019)
14. Krishnaveni, S.; Vigneshwar, P.; Kishore, S.; Jothi, B.; Sivamohan, S.: Anomaly-based intrusion detection system using support vector machine. In: Dash, S.S., Lakshmi, C., Das, S., Panigrahi, B.K. (eds.) Artificial Intelligence and Evolutionary Computations in Engineering Systems, pp. 723–731. Springer Singapore, Singapore (2020)
15. Liu, H.; Lang, B.: Machine learning and deep learning methods for intrusion detection systems: a survey. Appl. Sci. **9**(20), 4396 (2019)
16. Uikey, R.; Gyanchandani, M.: Survey on classification techniques applied to intrusion detection system and its comparative analysis. Int. Conf. Commun. Electron. Syst. **2019**, 1451–1456 (2019)
17. Ahmad, Z.; Shahid Khan, A.; Wai Shiang, C.; Abdullah, J.; Ahmad, F.: Network intrusion detection system: a systematic study of machine learning and deep learning approaches. Trans. Emerg. Telecommun. Technol. **32**(1), e4150 (2021)
18. Kim, G.; Lee, S.; Kim, S.: A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. Exp. Syst. Appl. **41**(4), 1690–1700 (2014)
19. Butun, I.; Morgera, S.D.; Sankar, R.: A survey of intrusion detection systems in wireless sensor networks. IEEE Commun. Surv. Tutor. **16**(1), 266–282 (2014)
20. Soe, Y.N.; Feng, Y.; Santosa, P.I.; Hartanto, R.; Sakurai, K.: Towards a lightweight detection system for cyber attacks in the IOT environment using corresponding features. Electronics **9**(1), 144 (2020)
21. Verma, A.; Ranga, V.: Machine learning based intrusion detection systems for IOT applications. Wireless Pers. Commun. **111**(4), 2287–2310 (2020)
22. Zainal, A.; Maarof, M.; Shamsuddin, S.M.: Ensemble classifiers for network intrusion detection system. J. Inf. Assur. Secur. **4**, 217–225 (2009)
23. Aksu, D.; Üstebay, S.; Aydin, M.A.; Atmaca, T.: Intrusion detection with comparative analysis of supervised learning techniques and fisher score feature selection algorithm. In: Czachórski, T., Gelenbe, E., Grochla, K., Lent, R. (eds.) Computer and Information Sciences, pp. 141–149. Springer, Cham (2018)
24. Chaudhari, R.; Patil, S.: Intrusion detection system: classification techniques and datasets to implement, (2017)
25. Wang, Y.; Shen, Y.; Zhang, G.: Research on intrusion detection model using ensemble learning methods. In: 2016 7th IEEE International conference on software engineering and service science (ICSESS) (2016)
26. Berthier, R.; Sanders, W.H.: Specification-based intrusion detection for advanced metering infrastructures. In: 2011 IEEE 17th Pacific rim international symposium on dependable computing, pp. 184–193, (2011)
27. Ullah, I.; Mahmoud, Q.H.: A two-level hybrid model for anomalous activity detection in IOT networks. In: 2019 16th IEEE Annual consumer communications networking conference (CCNC), pp. 1–6, (2019)
28. Kumari, A.; Mehta, A.: A hybrid intrusion detection system based on decision tree and support vector machine. In: 2020 IEEE 5th International conference on computing communication and automation (ICCCA), pp. 396–400, (2020)
29. Pokharel, P.; Pokhrel, R.; Sigdel, S.: Intrusion detection system based on hybrid classifier and user profile enhancement techniques. Int. Workshop Big Data Inf. Secur. **2020**, 137–144 (2020)
30. Kilincer, I.F.; Ertam, F.; Sengur, A.: Machine learning methods for cyber security intrusion detection: datasets and comparative study. Comput. Netw. **188**, 107840 (2021)
31. Fitni, Q.R.S.; Ramli, K.: Implementation of ensemble learning and feature selection for performance improvements in anomaly-based intrusion detection systems. In: 2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and ommunications Technology (IAICT), pp. 118–124. (2020)

32. Fitni, Q.R.S.; Ramli, K.: Implementation of ensemble learning and feature selection for performance improvements in anomaly-based intrusion detection systems. In 2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and ommunications Technology (IAICT), pp. 118–124. (2020)
33. Liang, C.; Shanmugam, B.; Azam, S.; Jonkman, M.; Boer, F.; Narayansamy, G.: Intrusion detection system for internet of things based on a machine learning approach (2019)
34. Kachavimath, A.V.; Nazare, S.V.; Akki, S.S.: Distributed denial of service attack detection using naïve bayes and k-nearest neighbor for network forensics, In 2020 2nd International conference on innovative mechanisms for industry applications (ICIMIA), pp. 711–717, (2020)
35. Hindy, H.; Bayne, E.; Bures, M.; Atkinson, R.; Tachtatzis, C.; Bellekens, X.: Machine learning based iot intrusion detection system: An mqtt case study (mqtt-iot-ids2020 dataset)' (2020)
36. Sah, G.; Banerjee, S.: Feature reduction and classifications techniques for intrusion detection system. Int. Conf. Commun. Signal Process. **2020**, 1543–1547 (2020)
37. Latah, M.; Toker, L.: An efficient flow-based multi-level hybrid intrusion detection system for software-defined networks. CCF Trans. Netw. **3**(3), 261–271 (2020)
38. Abdulrahman, A.; Ibrahem, M.K.: Evaluation of ddos attacks detection in a new intrusion dataset based on classification algorithms. Iraqi J. Inf. Commun. Technol. **1**, 49–55 (2019)
39. Verma, A.; Ranga, V.: Statistical analysis of cidds-001 dataset for network intrusion detection systems using distance-based machine learning. Proc. Comput. Sci. **125**, 709–716 (2017)
40. Sharafaldin, I.; Habibi Lashkari, A.; Ghorbani, A.A.: A detailed analysis of the cicids2017 data set. In: Mori, P., Furnell, S., Camp, O. (eds.) Information Systems Security and Privacy, pp. 172–188. Springer, Cham (2019)
41. Uzair, M.; Jamil, N.: Effects of hidden layers on the efficiency of neural networks. In: 2020 IEEE 23rd international multitopic conference (INMIC), pp. 1–6, (2020)
42. Mirza, A.H.: Computer network intrusion detection using various classifiers and ensemble learning. In: 2018 26th Signal processing and communications applications conference (SIU), pp. 1–4, (2018)
43. Pham, N.T.; Foo, E.; Suriadi, S.; Jeffrey, H.; Lahza, H.F.M.: Improving performance of intrusion detection system using ensemble methods and feature selection. In: Kim, D.S., Camtepe, S. (eds.) Proceedings of the Australasian computer science week multiconference 2018. United States of America: Association for Computing Machinery, pp. 1–6, (2018)
44. Zohuri, B.; Moghaddam, M.: Deep learning limitations and flaws. Modern Approaches Mater. Sci. **2**, 01 (2020)
45. Alaparthy, V.T.; Morgera, S.D.: A multi-level intrusion detection system for wireless sensor networks based on immune theory. IEEE Access **6**, 47 (2018)
46. Rajagopal, S.; Kundapur, P.P.; Hareesha, K.S.: A stacking ensemble for network intrusion detection using heterogeneous datasets. Secur. Commun. Netw. **2020**, 4586875 (2020)
47. da Costa, K.A.; Papa, J.P.; Lisboa, C.O.; Munoz, R.; de Albuquerque, V.H.C.: Internet of things: a survey on machine learning-based intrusion detection approaches. Comput. Netw. **151**, 147–157 (2019)
48. Bhavani, T.T.; Rao, M.K.; Reddy, A.M.: Network intrusion detection system using random forest and decision tree machine learning techniques. In: Luhach, A.K., Kosa, J.A., Poonia, R.C., Gao, X.-Z., Singh, D. (eds.) First International Conference on Sustainable Technologies for Computational Intelligence, pp. 637–643. Springer, Singapore (2020)
49. Sriavstava, R.; Singh, P.; Chhabra, H.: Review on Cyber Security Intrusion Detection: Using Methods of Machine Learning and Data Mining, pp. 121–132. Springer, Cham (2020)
50. Islam, M.J.; Wu, Q.M.J.; Ahmadi, M.; Sid-Ahmed, M.A.: Investigating the performance of naive- bayes classifiers and k-nearest neighbor classifiers, in *2007 International Conference on Convergence Information Technology (ICCIT 2007)*, pp. 1541–1546, (2007)
51. Mittal, D.; Gaurav, D.; Sekhar Roy, S.: An effective hybridized classifier for breast cancer diagnosis. In: 2015 IEEE International conference on advanced intelligent mechatronics (AIM). pp. 1026–1031 (2015)
52. Polikar, R.: Ensemble Learning, pp. 1–34. Springer, Boston (2012)
53. Sharafaldin, I.; Lashkari, A. H.; Ghorbani, A.: Toward generating a new intrusion detection dataset and intrusion traffic characterization. In ICISSP, (2018)
54. Hamid, Y.; Muthukumarasamy, S.; Journaux, L.: Machine learning techniques for intrusion detection: a comparative analysis **08**, 1–6 (2016)
55. Stiawan, D.; Idris, M.Y.; Bamhdi, A.M.; Budiarto, R.: Cicids-2017 dataset feature analysis with information gain for anomaly detection. IEEE Access **8**, 132911–132921 (2020)
56. Panigrahi, R.; Borah, S.: A detailed analysis of cicids2017 dataset for designing intrusion detection systems. Int. J. Eng. Technol. **7**(3), 479–482 (2018)
57. Vinayakumar, R.; Alazab, M.; Soman, K.P.; Poornachandran, P.; Al-Nemrat, A.; Venkatraman, S.: Deep learning approach for intelligent intrusion detection system. IEEE Access **7**, 525–550 (2019)
58. Faker, O.; Dogdu, E.: Intrusion detection using big data and deep learning techniques. In: Proceedings of the 2019 ACM Southeast Conference. ser. ACM SE '19. New York, NY, USA: Association for Computing Machinery, p. 86-93 (2019)
59. Karabulut, E.M.; Özel, S.A.; İbrikçi, T.: A comparative study on the effect of feature selection on classification accuracy. Proc. Technol. **1**, 323–327 (2011)
60. Jović, A.; Brkić, K.; Bogunović, N.: A review of feature selection methods with applications. In 2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO), 1200–1205 (2015)
61. Shamsaei, B.; Gao, C.: Comparison of some machine learning and statistical algorithms for classification and prediction of human cancer type. IEEE-EMBS Int. Conf. Biomed. Health Inf. **2016**, 296–299 (2016)