I. J. Computer Network and Information Security, **2023, 3, 30-46**
Published Online on June 8, 2023 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2023.03.03

Ensemble Learning Approach for Classification of Network Intrusion Detection in IoT Environment

Priya R. Maidamwar*
Department of Computer Science & Engineering, G H Raisoni University, Amravati, India
E-mail: priya.maidamwar@raisoni.net
ORCID iD: https://orcid.org/0000-0002-8711-1138
*Corresponding Author

Prasad P. Lokulwar
Department of Computer Science & Engineering, G H Raisoni College of Engineering, Nagpur, India
E-mail: prasad.lokulwar@raisoni.net
ORCID iD: https://orcid.org/0000-0001-7608-6080

Kailash Kumar
College of Computing and Informatics, Saudi Electronic University, Riyadh, Kingdom of Saudi Arabia
E-mail: k.kumar@seu.edu.sa
ORCID iD: https://orcid.org/0000-0003-2916-719X

Received: 16 May 2022; Revised: 04 July 2022; Accepted: 13 October 2022; Published: 08 June 2023

Abstract: Over the last two years,the number of cyberattacks has grown significantly, paralleling the emergence of new attack types as intruder's skill sets have improved. It is possible to attack other devices on a botnet and launch a man-in-the-middle attack with an IOT device that is present in the home network. As time passes, an ever-increasing number of devices are added to a network. Such devices will be destroyed completely if one or both of them are disconnected from a network. Detection of intrusions in a network becomes more difficult because of this. In most cases, manual detection and intervention is ineffective or impossible. Consequently, it's vital that numerous types of network threats can be better identified with less computational complexity and time spent on processing. Numerous studies have already taken place, and specific attacks are being examined. In order to quickly detect an attack, an IDS uses a well-trained classification model. In this study, multi-layer perceptron classifier along with random forest is used to examine the accuracy, precision, recall and f-score of IDS. IoT environment-based intrusion related benchmark datasets UNSWNB-15 and N_BaIoT are utilized in the experiment. Both of these datasets are relatively newer than other datasets, which represents the latest attack. Additionally, ensembles of different tree sizes and grid search algorithms are employed to determine the best classifier learning parameters. The research experiment's outcomes demonstrate the effectiveness of the IDS model using random forest over the multi-layer perceptron neural network model since it outperforms comparable ensembles analyzed in the literature in terms of K-fold cross validation techniques.

Index Terms: Feature Selection, Intrusion Detection System, IDS, N_BaIoT Dataset, Random Forest (RF), Multilayer Perceptron Neural Network(MLP NN), UNSW NB15 Dataset.

1. Introduction

There have been a number of attacks recently due to vulnerabilities in network connectivity. Researchers are observing an extraordinary rise in the number and variety of cyberattacks, which is quickly turning into an unmanageable problem. A second reason for concern is that the growth of Internet-connected computer devices (often referred to as "things" or "IoT") makes it easier for cybercriminals to conduct attacks. Just a few examples include smart grids, smart cities, smart homes as well as other Internet of Thing's platforms and protocols. But when it comes to integrating IoT technology, security is the most crucial aspect. The speedy development of Internet of Things technology has resulted in inadequate safety standards. In addition, the IoT generates a vast amount of data that might be hijacked by an attacker during network transmission.

Figure 1 depicts the IoT Security framework [1]. This is a three-tiered structure which includes Perception,Transportation and Application Layer. Acquisition and management of data is handled by the Perception layer.

This work is open access and licensed under the Creative Commons CC BY 4.0 License.

Following this, the data is transferred to a gateway for additional analysis. In this layer, RFID, GPS, wireless sensor networks and other technologies are present. A malicious attacker might exploit this layer to obtain and analyze data while posing security hazards, such as eavesdropping and malicious routing, for their own purposes. The Transportation layer provides access to the Perception and Application layers from any location. Heterogeneous networks include local area networks, wireless ad hoc networks, WiFi, and 3G. Data leakage and network damage are the most common outcomes of a DDoS attack [2]. Attack detection and prevention technologies like Intrusion Detection Systems (IDS) are frequently employed to counteract these types of attacks at the transport layer. IDS often employ one or more of three detection methods: Signature-based, Anomaly-based, or a Hybrid approach. Due to its ability to identify cutting-edge threats, anomaly detection has traditionally received the greatest attention of the three methods discussed above. Anomaly detection relies on a classification model that has been properly trained to anticipate such coming dangers of some sort. In addition to this benefit, anomaly-based detection is occasionally linked to a high rate of false alarms (FAR). Because of this, it makes sense to build a high-precision classification model based on the lower FAR.

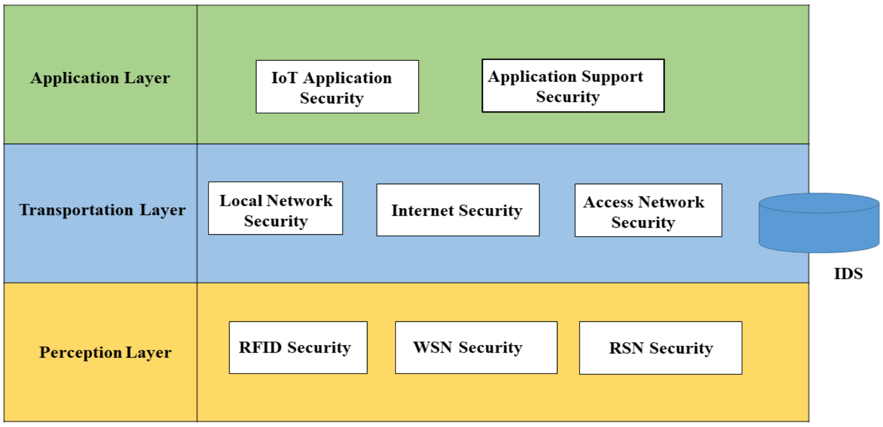


Fig.1. IoT security framework

A single classifier or a collection of classifiers can be used to build classifier models for anomaly-based IDS. A significant improvement in prediction accuracy has been made by the most effective classifier groups or ensemble learners when compared to the best individual classifiers. Serial and parallel classifier ensemble topologies [3] are the two most used configurations. In serial topologies, single classifiers are arranged sequentially, and a method ordering is defined for each. According to this criterion, Adaboost is an excellent example of a classifier ensemble. The parallel design, on the other hand, considers each classifier as if it had been given the exact same input data, so the combined classifier's final prediction is based on the results of the individual classifiers. In Ref. [4], the majority of classifier ensembles have this structure. According to this criterion, Random Forest as well as Multilayer Perceptron Neural Network are the suitable ensemble learners [5]. Using these ensemble classifiers, this research proposes an anomaly-based IDS that is effective and efficient. Random forest and Multilayer Perceptron Neural Network models are ensembled in such a way that the final accuracy is improved as well as the testing time is reduced. As a proof of concept, we evaluated the classifier performance on UNSWNB15 [6] and N_BaIoT [7] datasets. The following are the goals of this study.

- To find the ideal training parameters for Random Forest and Multilayer Perceptron Neural Network, a comprehensive search method such as grid search is performed.
- The results of ensemble classifiers on the current benchmark datasets, UNSWNB15 and N BaIoT, have been reported and examined. It was found that our proposed model is 99 percent accurate compared to 97 percent before feature selection, as demonstrated by the testing findings.
- To eliminate ranking bias often associated with the selection of overfitting features, we analyse the intrusion detection dataset and remove duplicate data.
- In pre-processing, we delete classes with less than 2% of the complete training set's total data. Oversampling is also utilized to ensure that the normal and abnormal classes are evenly distributed.

The following is the outline for the research paper. Section II extends the overview of literature work done on UNSW-NB-15 and N_BaIoT datasets and their comparative analysis is presented in tabular form. Section III presents proposed methodology. The experimental results are the main topic of Section IV, and a summary of the results is covered in Section V.

2. Related Works

Over the past two decades, numerous intrusion detection technologies have been reported and suggested. In order to create IDS that are both reliable and efficient, so many of the studies utilizing machine learning have been presented, and various enhancements have been created [8]. Authors in [9] have presented comparative studies of current ensemble ML algorithms for unbalanced datasets. There will be only a small number of intrusion events, so finding the most effective intrusion detection ensemble classifier is critical to this project's success. It was compared to the UNSWNB15 reference dataset and various boosting techniques to see how well the trees in the bag were learning. It wasn't just about the algorithms, either. The findings collected demonstrated that in the environment specified by Bagged Tree and Gentle Boost work with the highest accuracy and ROC value under the examined parameters, while RUS Boost performance is the lowest. The two publicly accessible labelled intrusion detection assessment datasets NSLKDD along with UNSWNB15 for various integration testing environments were analyzed using an ensemble approach proposed by researchers [10]. Most important features were extracted during pre-processing by removing duplicate and extraneous elements from the data set. It was then used to construct a data mining strategy that had high rate of detection and few false alarms. The experiments showed that the suggested ensemble approach performed better than other well-known cutting-edge IDS algorithms in successfully identifying a variety of novel threats. A new algorithm for an intrusion detection system that recognizes attacks based on the UNSW-NB15 dataset has been proposed and created by the authors in [11]. Based on prior research, it has been demonstrated that the k average clustering approach is effective at locating the centre of a cluster and its immediate neighbors. The feature selection strategy is used to construct a one-dimensional dataset with distance as the sole feature. Several kinds of attacks can be detected by the training and testing of ensemble classifiers. This approach successfully classifies attacker’s 90 percent of the time. An effective random forest classifier for anomaly detection in IoT networks was presented by Primartha R. and Tama B.A. There was a total of ten different classifiers constructed and evaluated. The experiment employed three datasets: NSLKDD, UNSWNB15, and GPRS, each with a different number of trees. In compared to other classifiers, the RF800 was proven to be statistically significant. According to existing methods, the suggested model outperformed them in terms of accuracy and FAR measurement. It was given by Moustafa N. et al. [12] to minimize hostile occurrences, especially botnet attacks. The protocols used in Internet of Things networks are DNS, HTTP and MQTT. As a result of analyzing their potential qualities, new statistical flow properties are created. By employing different machine learning methods namely Artificial Neural Networks, Decision Trees and Naive Bayes it was successful to identify detrimental occurrences (ANN). Two datasets are utilized to test selected significant features and the ensemble classifiers: UNSWNB15 and NIMS. Correlation coefficients were used to test the hypothesized properties of both normal and malicious actions, and the results were consistent. The proposed ensemble method outperforms the system's classification algorithms including three additional novel methods in terms of detection rates as well as the proportion of false positives.

With the help of the BoT-IoT dataset, Alsamiri, Jadel, and Khalid Alsubhi [13] identified IoT network attacks using machine learning techniques. Seven machine learning algorithms were used throughout the implementation phase, and the bulk of them performed effectively. New features from the dataset were discovered and compared to earlier research using the Random Forest Regressor technique. In terms of F-measure, new characteristics were found to produce superior results. RFSVM, RFNBKNN, and RFKNNLR are three hybrid models proposed by Pandey, Amritanshu, et al. [14]. AUC, recall, precision, and classification accuracy are just a few of the various performance indicators examined. RFSVM outperforms the other two models in almost every metric: accuracy,AUC, and classification errors. RFSVM outperforms other models as a result of this. Because the ensemble model outperforms individual classifiers, supervised classifier integration is used for improved performance. Another efficient and successful IoT botnet based attack detection solution was provided by Alqahtani, Mnahi, et al. [15]. The system employs a fisher score based feature selection technique as well as genetic based extreme gradient boosting (GXGBoost) model to pinpoint the highly crucial characteristics and recognize IoT botnet threats. Also, the fisher score is used to determine important characteristics and eliminate unimportant ones by minimising intra-class distance and increasing inter-class distance. GXGBoost, on the other side is an excellent and successful methodology for classifying IoT botnet attacks. On a publicly available botnet dataset of IoT devices, several experiments were conducted. Holdout and a 10-fold technique were used to develop the proposed method. Only three out of 115 traffic types had high detection rates, and the IoT botnet attack detection procedure' overall performance was enhanced. Kanimozhi, V., and Thangavel Prem Jacob [16] developed a framework with a variety of classifiers, including Naive Bayes Classifier, KNearset Neighbour, support vector machine classifiers,random forest classifiers,Adaboost with decision trees as well as artificial intelligence to distinguish the percentage of botnet threats against a current realistic cyber dataset CSECICIDS 2018. The accuracy of a given classifier was reported as a consequence of classification. In the provided framework, the curve of calibration is also employed as a common approach to statistical methodologies. It creates a reliability plot to see if the prediction probabilities of the separate classifiers are properly calibrated. Finally, the shown graphs demonstrated that artificial intelligence technology outperforms all other classifiers in terms of producing reliability graphs for determining whether or not the various classifiers prediction probabilities are properly adjusted. A light weight intrusion detection system was suggested by zer, Erman et al. [17]. For this, ten machine learning methods and the current BoTIoT dataset were chosen. The producers of this dataset recommended 12 top attributes for use in this investigation. From 12 high-quality feature pairs, 66 distinct feature pairs were created. Following that, ten full-featured intrusion detection systems with 12 full-featured features were constructed by training ten machines. Similar to the 660, which was developed by training 10 machine learning algorithms in each of the 66 feature pairs, it was a small functional pair-based intrusion detection system. Additionally, using a variety of machine learning techniques, researchers looked at 10 intrusion detection systems trained having the 12 best features and 660 intrusion detection systems trained with 66 feature combinations. Following that, a feature pair basis lightweight intrusion detection system with ten full-featured intrusion detection accuracy levels was chosen. This results in a functional pair that is both optimal and efficient, as well as a lightweight intrusion detection system. It's been decided. The smallest intrusion detection system has a detection accuracy of above 95%. The following is an overview of relevant work that used the ensemble approach to construct IDS utilizing various datasets, which is shown in Table 1.

Table 1. An overview of related work

Reference	Year	Ensemble Method	Algorithm	Data Set Used	Performance Metric
(Timčenko and Gajin, 2017) [9]	2017	Bagged trees, AdaBoost, USBoost, LogitBoost and Gentle Boost	C4.5,K-Nearest Neighbours	UNSW-NB15	DR is 100, ROC value is 0.999 for Gentle Boost
(Kamarudin et al., 2017) [10]	2017	Ensemble classifier Logitboost,	Random Forests	NSL-KDD and UNSW NB15	Accuracy is 99.45% DR is 99.10%, FAR is 0.18%
(Aravind and Kalaiselvi, 2017) [11]	2017	Ensemble classifier	K-means	UNSW-NB15	Accuracy is 90%
(Primartha and Tama, 2018) [1]	2017	Random Forests	Decision tree	GPRS NSL-KDD, UNSW-NB15,	Accuracy is 95.5%, FAR is 7.22%
(Moustafa et al., 2019) [12]	2018	AdaBoost ensemble learning	Decision Trees, AdaBoost, Naive Bayes, Artificial Neural Networks	NIMS botnet and UNSW NB15	Accuracy is 99.54%, DR is 98.93%, FPR is 1.38%
(Alsamiri and Alsubhi, 2019) [13]	2019	Adaboost	Random Forest, Quadratic discriminant analysis, Multilayer Perceptron, K-Nearest Neighbours Naıve Bayes	BoTIoT	Accuracy is 98% Precision is 97% Recall is 97% F1-Score is 97%
(Pandey et al., 2021) [14]	2019	Bagging	Linear Regression, Random Forest, K-Nearest Neighbor, Support Vector Machine, Naive Bayes	N_BaIoT	Accuracy is 85.34%
(Alqahtani et al., 2020) [15]	2020	Genetic-based extreme gradient boosting (GXGBoost)	Decision tree	N_BaIoT	Accuracy is 99.96%
(Kanimozhi and Jacob, 2020) [16]	2021	Adaboost with Decision Tree	KNearset Neighbor Classifier, Naïve Bayes,Support Vector Machine Classifier, Random Forest Classifier, Artificial Intelligence	CSE CIC IDS2018	Accuracy is 99% Precision is 99% Recall is 100% F1-Score is 99%
(Özer et al., 2021) [17]	2021	AdaBoost	Linear SVM, Naive Bayes, K-Nearest Neighbors, Decision Tree, Random Forest, Neural Network, Quadratic Discriminant Analysis	BoTIoT	Accuracy is 99%

3. Proposed Methodology

3.1. Dataset Characteristics

This study relies on the datasets UNSWNB-15 and N BaIoT-18. Data from the UNSWNB15 dataset includes both real-world network traffic and simulated traffic derived from a variety of attack vectors, training datasets, and testing. The likelihood of a certain behaviour occurring in each dataset is the same. There are less cases in each minority class than there is overall, thus this dataset can be termed as an imbalanced dataset. From 100 GB of raw TCPdump traffic, researchers had produced the dataset. There are 49 features utilized to arrange and produce traffic instances, including flow (5 features), base (13 features), content (8 features), time (9 features), additional (12 features), attack categories (1 feature), and the label (1feature). DoS, Fuzzers, Analysis, Backdoors, Shellcodes, Worms, Exploit and Generic attacks are some of the nine types of attacks that can be used against a computer system [18,19]. The NBaIoT dataset is made available to the general public in order to identify botnet assaults on IoT devices. The concept was hatched by Y. Meidan for the purpose of determining whether or not this dataset contained a real public IoT botnet, researchers analysed over 5 million real-world traffic datapoints. It came from a network of nine IoT devices that had been compromised by two botnets. There are two major botnet families: Bashlite and Mirai. In IoT devices, these are the most common botnets that are malicious. In this dataset, each device has a different number of attacks for each type of attack. Many files make up the NBaIoT dataset. 115 attributes are associated with each file along with the binary classification "benign" or "TCP (Transmission Control Protocol) assault." For the purposes of multiclass classification, TCP attacks can be separated into Mirai and Bashlite attacks [20,21].

3.2. Methodology

Figure 2 displays a novel recommended model for IDS categorization based on the UNSW-NB15 as well as N BaIoT datasets. The following steps are part of the recommended strategy.

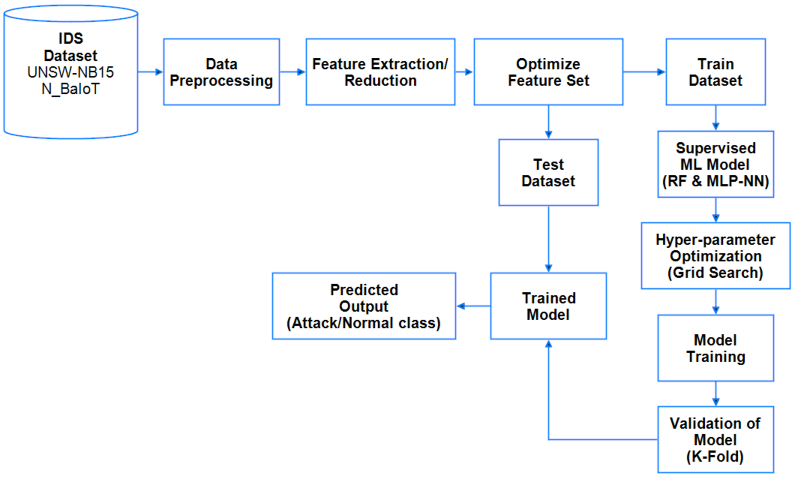

IDS Dataset UNSW-NB15 N_BaIoT
Data Preprocessing
Feature Extraction/ Reduction
Optimize Feature Set
Train Dataset
Test Dataset
Supervised ML Model (RF & MLP-NN)
Hyper-parameter Optimization (Grid Search)
Predicted Output (Attack/Normal class)
Trained Model
Model Training
Validation of Model (K-Fold)

Fig.2. Proposed model

A. Dataset Splitting

The IDS dataset should be divided into train as well as test data in order to train a model using a machine learning approach. In our implementation, we identified a split ratio of 70:30, meaning that 70% of the dataset is utilized to train the model and the remaining 30% is utilized to test the model that was created from it.

B. Preprocessing of Data

Before training can begin, the data must be cleaned and formatted properly due to their sheer volume and size. Consider the fact that certain data can be missing or misinterpreted. An incorrect conclusion could be drawn when raw data is fed into a modelling tool. Categorical feature coding, feature selection, and feature scaling were just the data preprocessing procedures used on the dataset because there were no nulls or duplicate values.

C. Selection and Extraction of Features

After the datasets have been prepared and cleaned, choosing an initial level feature set is necessary to direct the test for horizontal complexity. The training along with testing datasets are preprocessed, and feature extraction methods are employed to extract features.

D. Training of models, Tuning the Hyperparameter and Classifications

The model is then trained using supervised machine learning methods such as Random Forest (RF) along with multi-layer perceptron Neural Network Model (MLP NN). Tenfold cross-validation is also employed in the random forest method (RF). Classifiers in random forests can be used to address classification or regression problems. This is yet another iteration of Brayman's bagging outfit. There are times when bagging or boosting is faster and more effective than boosting. Random Forest can be taught as a bagging version using a random tree as the classifier. It is called ensemble learning when a decision tree operates as the learning model's principal classifier. Another set of classifiers using trees is called Random Forest. Each tree grows in accordance with a random vector that is dispersed randomly throughout the tree's growth. Each tree in the ensemble will cast a vote for the most prevalent input vector type. Random forest variation can be generated by varying some decision tree settings or selecting a random sample from a collection of attributes or a dataset. The tunable random forest's two parameters are the number of parameters that can be selected on each node and the number of nodes that make up the forest.

Figure 3. shows the classification of random forests used in this study. Random forests are collections of individual trees that can be used to categorize data as "normal" or "abnormal," depending on the data's characteristics. Votes are cast on each tree's classification forecast in order to arrive at an overall classification. In order to give a thorough baseline, different numbers of trees were selected. An extra set of parameters can be obtained by grid search such as the minimum and maximum depths, nbins and sample rates, colour sample rates per tree and histogram type [22].

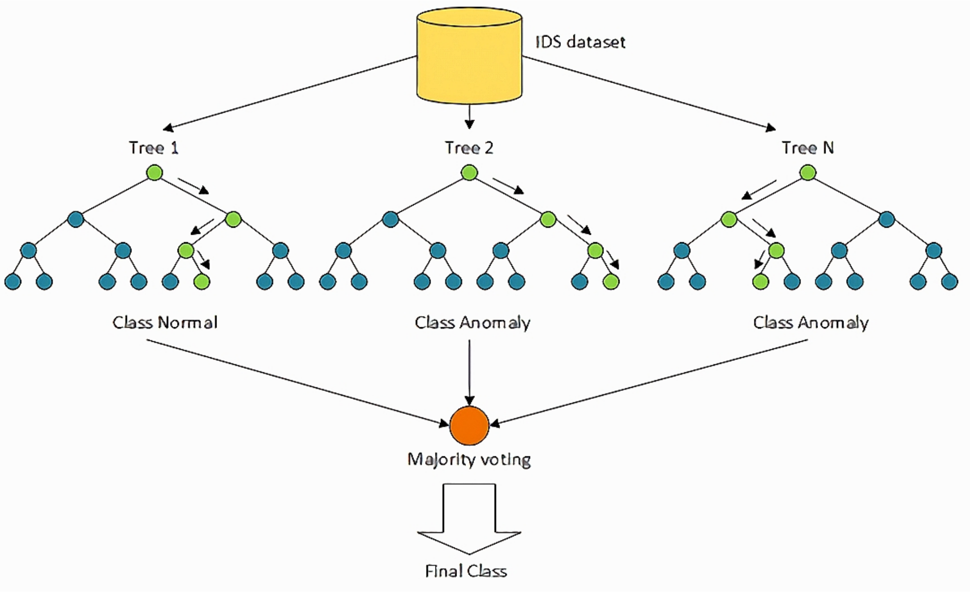


Fig.3. Illustration of random forest classifier

The multilayer perceptron (MLP) is one of the most widely used artificial neural networks. The basic MLP Model is shown in figure 4. With output neurons equal to the number of classes and input neurons equivalent to the number of characteristics, MLPs are capable of solving classification issues. Backpropagation is frequently used to train intermediate layers between input and output. The network takes the activation function of the preceding layer, weight values, and bias values as inputs to calculate the output of each layer [23].

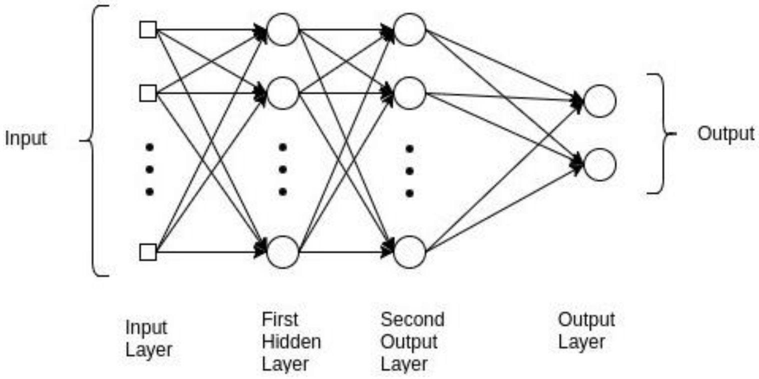


Fig.4. Basic MLP model

Finally, when training, effective parameters has to be changed or established for the good classification outcome of the set using hyper-parameter tuning. Grid search can be utilized to identify the optimal model parameters for any machine learning model; hence it is not restricted to a single model type. The grid search technique on input data yielded the search range and optimum values of model parameters. i.e. training data displayed in Table 2. As shown below, the algorithm for the suggested methodology.

Algorithm 1: UNSW-NB15 Dataset

Input:

A vector of feature values, the training and testing instance set S, and a class as well as label value
Feature Set F(i) = f1(i), f2(i), etc.
Attack(1), NORMAL(0) are part of the label set L(i).

Initialization:

Step1: Feature and label data should be prepared from raw dataset values from the UNSW Dataset.

Preprocessing Phase:

Step2: For every feature data
Make a calculation of the features set's normalised value.
Specify a range for scaling all feature data.
Parameter Stage of hypertuning

Step3: Establish the RF along with MLP models.
Step4: List all of the hyperparameters for ML algorithms along with the range of feasible values.
RF: "C," "random_state," "penalty," and "n_jobs"
MLP: 'hidden_layer_sizes','max_iter', 'activation','solver', 'alpha', 'learning_rate'
Step5: Hyperparameter values are sampled using the Grid Search CV Function.
Step6: Analyse each hyper parameter's value, and choose the one with the best score.
Step7: Utilise the K-Fold Validation Learning Method to validate the model.

Training Phase:

Step8: Set up the parameter that was tuned for the RF and MLP ML models.
Step9: Set up the label and feature information of the training dataset.
Step10: Prepare the model for the appropriate ML algorithms.
Step11: Utilise the K-fold cross validation technique to validate the model's performance.
Step12: Save the trained model (TMrf, TMmlp) if validation was successful; otherwise, go back to step 8 and repeat.

Testing Phase:

Step13: Set up the attribute values in the testing dataset.
Step14: Import the learned ML algorithm model.
Step15: Decide whether Attack (1) or Normal (0) will occur.
Step16: To test the system's accuracy, plot a confusion matrix against the real label data and the predicted label values.

Evaluation Phase:

Step 17: Confusion matrix parameters according to TP, FP, TN, and FN are used to assess the effectiveness of classification model C.

Algorithm 2: N_BaIoT Dataset

Input:

D: A collection of n data objects
C: For instance, "Normal; Attack" in a set of classes.
X: To classify data record be classified
H: (That X is within category C) Hypothesis

Output:

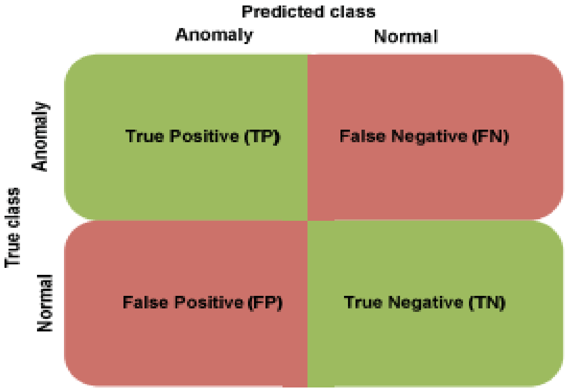
Predicted IDS category in which X should be categorised.

```
Pseuodocodes:
// Learning //
For j =1 to the number of classes

// Step 1: Preprocessing Phase //
Each feature's data is Fi.
Locate the feature variable values that are missing.
Determine the normalised value of the entire feature collection.
Scale all feature data to a particular range.
EndFor

// Step 2: Feature Reduction Phase //
Scale the dataset samples using MinMaxScaler()
Define autoencoder architecture using encoder and decoder
Train the autoencoder model with mean absolute error and adam optimization function for train and test data
Get the encoder layer and use the method predict to reduce dimensions in data
Split reduce feature data into train and test set
EndFor

// Step 3: Parameter Tuning Phase //
For feature and target data, training loss     Min Error
Specify the range of acceptable values for all ML algorithm hyperparameters.
Hyper parameter values are sampled using the Grid Search CV Function.
Analyse each hyper parameter value, and choose the one with the best score.
Utilise the K-Fold Validation Learning Method to validate the model.
EndFor
```

```
// Step 4: ML Model Training //
Initialize the parameter tuned for ML model.
Create the feature as well as label data from scratch required for training dataset.
Prepare the predictive model for the correct ML algorithms.
Maintained the trained model for the relevant ML algorithms
// Testing //

// Step 5: Prediction of Test Dataset //
Create the attribute data from scratch for the test dataset.
Upload the trained ML algorithm model.
Decide whether Attack (1) or Normal (0) will occur.
To test the system's accuracy, generate a confusion matrix comparing the real label data and the expected label data.
Analyse the effectiveness of the classification model C using the ROC and the TP, FP, TN, and FN Confusion Matrix Parameters.
Endfor
```

3.3. Performance Measures

It is based on a confusion matrix which indicates the amount of correctly as well as incorrectly detected cases by event type (normal vs. aberrant) (Figure 5). Because of these statistically defined metrics, the comparative comparison of classifiers is done using statistically determined measurements. Performance metrics are defined using the confusion matrix, which relies on four fundamental variables. Positive in every sense (TP). This number represents the number of test predictions that were correct. As the name suggests, True Negatives indicate a group of correctly predicted negative outcomes inside an experimental dataset (TN). Positives that aren't true (FP). In the test set, this figure represents the number of incorrect positive predictions. As an indicator of the number of incorrectly predicted results, the False Negative Prediction metric is used (FN) [24].



True class	Predicted class: Anomaly	Predicted class: Normal
Anomaly	True Positive (TP)	False Negative (FN)
Normal	False Positive (FP)	True Negative (TN)

Fig.5. Confusion matrix

In addition to the measures listed above, a classifier's quality can be evaluated using a variety of performance metrics. The most widely used metrics are sensitivity, specificity, accuracy, and ROC value [25]. The detection rate as well as Receiver Operating Characteristics were examined in this experiment (ROC). The ROC curve depicts the system's overall performance by examining at its ROC curve, which shows how well detection rates and false positive rates correlate. Accuracy (ACC), Precision (or DR), Recall (or DR), FAR, F-score, and the confusion matrix [26] are extensively used metrics in the literature to evaluate NIDS approaches, as shown in (1), (2), (3), (4), and (5).

$$ACC = \frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$

$$Pr\ e\,cision = \frac{TP}{TP+FP} \tag{2}$$

$$Re\ c\ all = \frac{TP}{TP+FN} \tag{3}$$

$$FAR = \frac{FP}{FP+TN} \tag{4}$$

$$F - score = \frac{2*Pr\ ecision*Re\ call}{Pr\ ecison+Re\ call} \tag{5}$$

Where TP, TN, FP, and FN stand for True Positive, True Negative, False Positive and False Negative respectively.

4. Summary and Explanation

4.1. Experimental Setup

For intrusion detection classification, UNSW-NB15 as well as N_BaIoT datasets are utilized in our research. On a computer with an Intel i5 clocked at 2.80 GHz, 16 GB associated with RAM plus Windows 10 (64 bit) as the operating system, the classification was performed using Pycharm IDE with Anaconda distribution as well as Scikit Learn Machine. We process data, identify features, and present findings for our studies using Scikit-Learn, Numpy, pandas, matplotlib, and other software.

4.2. Experimental Observations and Performance Analysis

Several tests were run on the provided data set for detecting harmful events to assess the effectiveness and utility of the Random Forest (RF) with Multi-layer Perceptron Neural Network (MLPNN).

A. Result Evaluation of UNSW-NB15 Dataset

Table 2 displays the training parameters for Random Forest as well as Multilayer Perceptron classification models.

Table 2. HyperParameter details for UNSW-NB15

Classifiers	Model Parameters	Search Range	Binary Classifier Selected Range	Multclass Classifier Selected Range
Random Forest	max_features	[10, 100, 500]	1000	10
	n_estimators	[10, 100, 1000]	500	100
	max_depth	[10, 40, 70, 100]	70	40
	min_samples_split	[2, 5, 10]	10	5
	max_leaf_nodes	[50, 100, 200]	100	50
	random_state	[10, 40, 70, 100]	100	100
Multilayer Perceptron	hidden_layer_sizes	[(100, 50, 10), (50,), (100,)]	(100, 50, 10)	100
	max_iter	[500, 1000]	1000	500
	activation	[‘tanh’, ‘relu’]	Tanh	Tanh
	solver	[‘ sgd ’ , ‘ adam ’, ‘ lbfgs ’]	sgd	sgd
	alpha	[0.0001, 0.05]	0.0001	0.0001
	learning_rate	[‘constant’, ‘adaptive’]	Adaptive	Adaptive

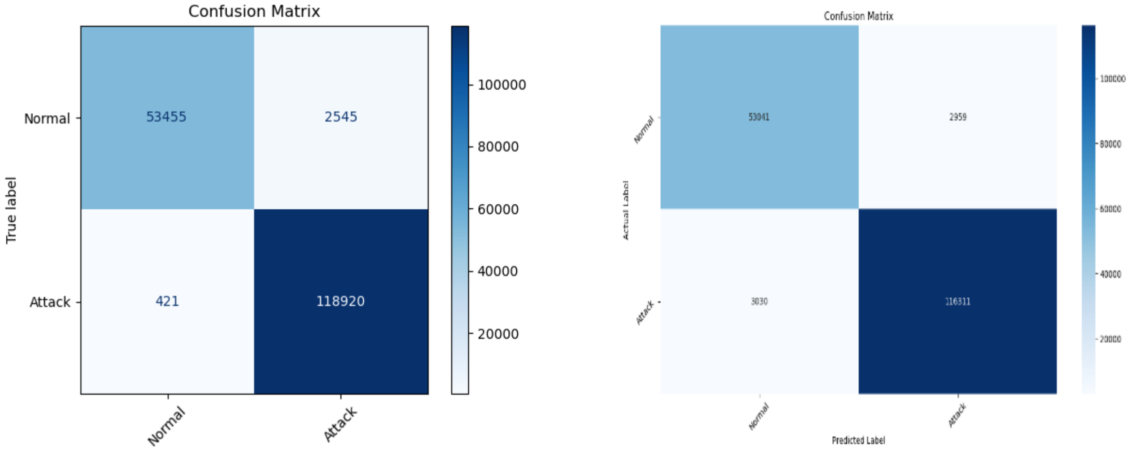


Fig.6. Binary classification confusion matrix utilising a) RF and b) MLP classifier

Figure 6-7 displays the confusion matrix of two classifiers performing binary as well as multiclass classification. The suggested method is employed for UNSW-NB15 dataset and the distinction between legitimate and malicious assaults is determined by comparing the expected result to the actual label for each attack.

The parameters of Classification report for the binary classification were evaluated for normal as well as attack output categories according to test results's precision, recall, and f-score. In terms of all parameter values, Random Forest algorithm is definitely superior to Multi-layer Perceptron methods. Ten different sorts of attacks have been produced as classification report parameters for multiclass categorization. Of all the types of attacks, the Generic and Normal types are successfully and accurately classified by both classifiers.

The performance in terms of Accuracy, Precision, F-score, Recall, Matthews' Correlation Coefficient (MCC), and Kappa Score for binary and multiclass classification were obtained. All performance parameters are efficient for Random Forest algorithms for binary classification system. The overall performance of the suggested system is depicted graphically in Figures 8 and 9.

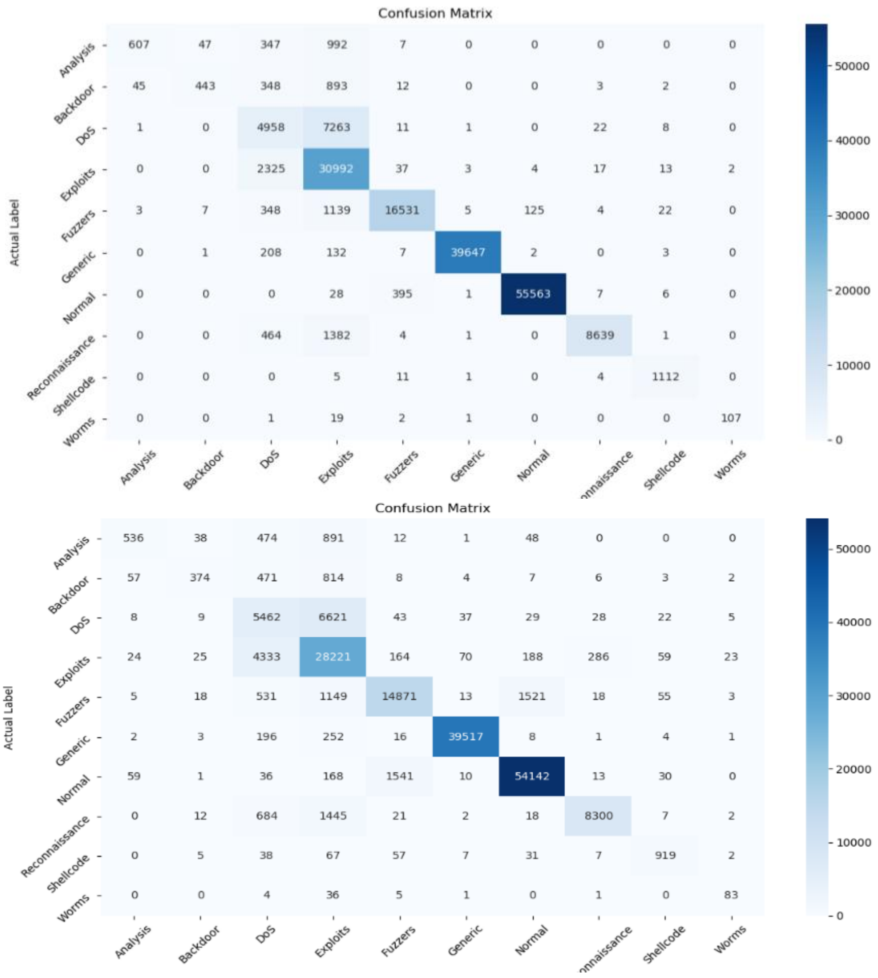


Fig.7. Multiclass classification confusion matrix utilising a) RF and b) MLP classifier

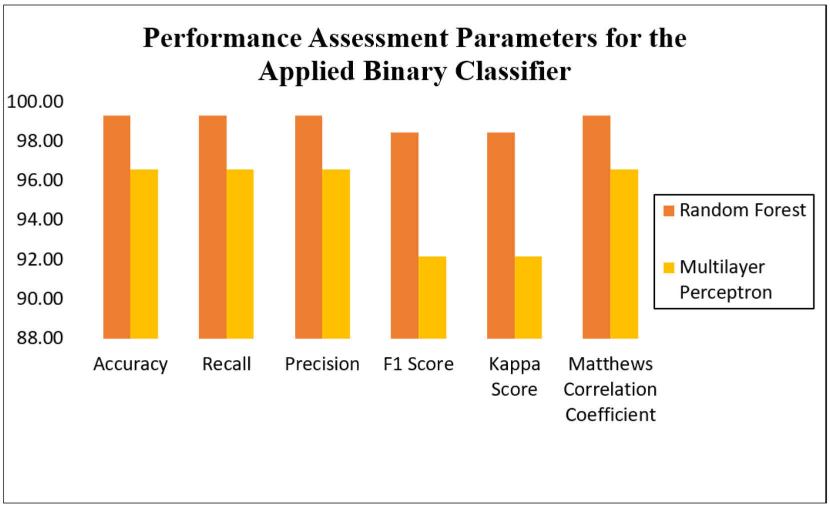


Fig.8. Entire system performance metric for binary classification

The system performance parameters for each class for binary classification using RF and MLP were obtained. This result shows that sensitivity and positive predictive value is significantly best using RF classifier than MLP. From this table, Generic, Normal and Shellcode are the best performing category using RF classifier. Figure 10, 11 and 12 shows the graphical representation of attack wise performance parameters in multiclass classification system. system for both type of classification system.

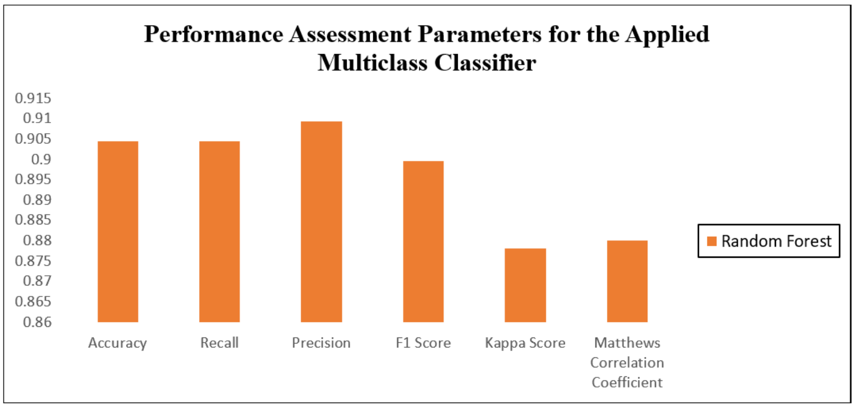


Fig.9. Entire system performance metric for multiclass classification

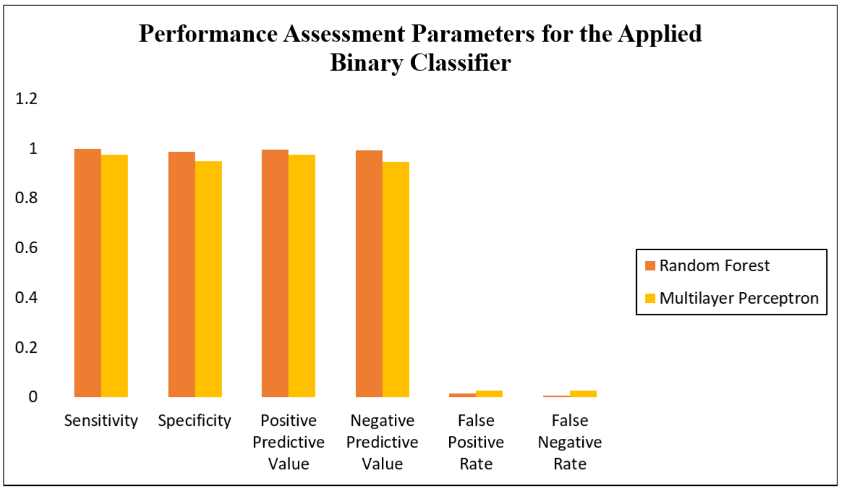


Fig.10. Performance characteristics of the binary classification system

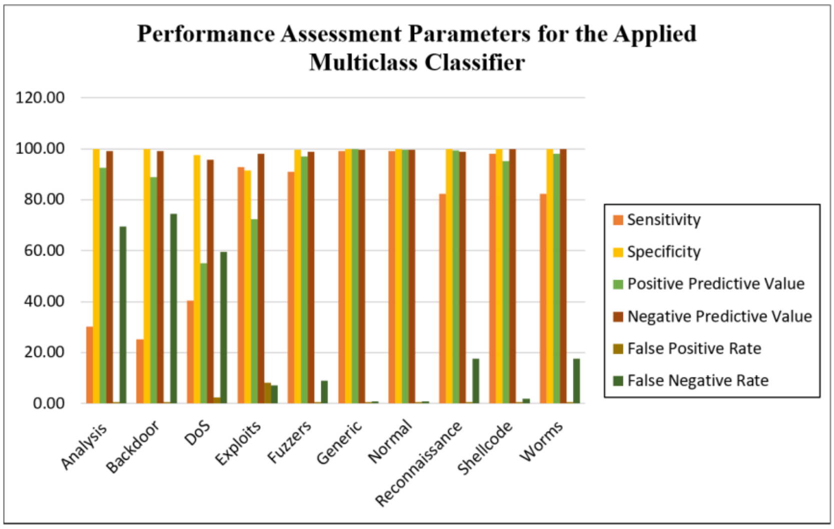


Fig.11. Performance characteristics of the multiclass classification system

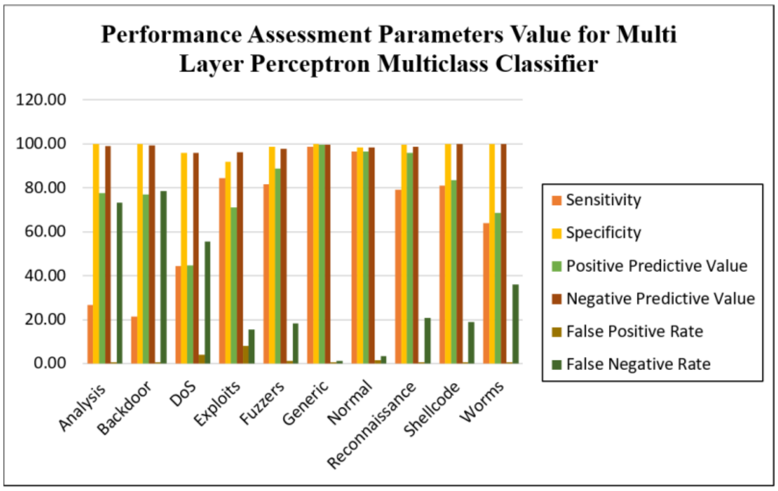


Fig.12. System performance parameter for each class for multiclass classification using MLP

B. *Result Evaluation of N_BaIoT Dataset*

This dataset comprises of 9 IoT devices from which 5 unique devices as mentioned beow are selected for classification.

- 'Danmini_Doorbell' – Dev A,
- 'Ecobee_Thermostat' – Dev B,
- 'Philips_B120N10_Baby_Monitor' – Dev C,
- 'Samsung_SNH_1011_N_Webcam' – Dev D,
- 'SimpleHome_XCS7_1002_WHT_Security_Camera' – Dev E

Table 3 displays the parameters for all 5 devices utilized to train the Random Forest as well as Multilayer Perceptron classification models. The evaluation time required for training, testing and hypertuning of five devices namely A, B, C, D and E are summaried in table 5. Performances of 5 IoT devices is depicted in figure 14, 15, 16, 17 and 18. Binary and Multiclass classification has been performed for both normal and attack activities. For Multiclass classification 2 attacks namely Mirai and Bashlite are used as labels.

Table 3. HyperParameter details for N_BaIoT dataset

Classifiers	Model Parameters	Search Range	A	B	C	D	E
Random Forest	n_estimators max_features max_depth min_samples_split max_leaf_nodes random_state	[10, 100, 1000] [10, 100, 500] [10, 40, 70, 100] [2, 5, 10] [50, 100, 200] [10, 40, 70, 100]	10 100 40 5 50 10	10 100 40 5 50 10	100 100 50 5 50 10	10 100 40 5 50 10	100 100 70 10 100 70
Multilayer Perceptron	hidden_layer_sizes max_iter activation solver alpha learning_rate	[(50, 30, 10), (50,), (100,)] [500, 1000] [‘tanh’, ‘relu’] [‘sgd’, ‘adam’, ‘lbfgs’] [0.001, 0.05] [‘constant’, ‘adaptive’] S	(50,) 500 Tanh Adam 0.001 adaptive	(50,) 500 Tanh Adam 0.001 Adaptive	(100,) 1000 Tanh Adam 0.05 adaptive	(50,) 500 Tanh Adam 0.001 Adaptive	(100,) 1000 Tanh Adam 0.05 adaptive

Table 4. Evaluation time for N_BaIoT dataset

Classes	Device A		Device B		Device C		Device D		Device E	
	RF	MLP	RF	MLP	RF	MLP	RF	MLP	RF	MLP
Hypertuning	492.07	501.48	508.01	546.98	502.9	530.5	465.78	495.8	560.6	580.4
Training	245.6	280.4	258.4	290.5	238.4	295.6	215.6	250.1	260.8	305.5
Testing	102.5	115.7	112.8	128.4	108.4	120.6	97.5	101.6	122.4	134.9

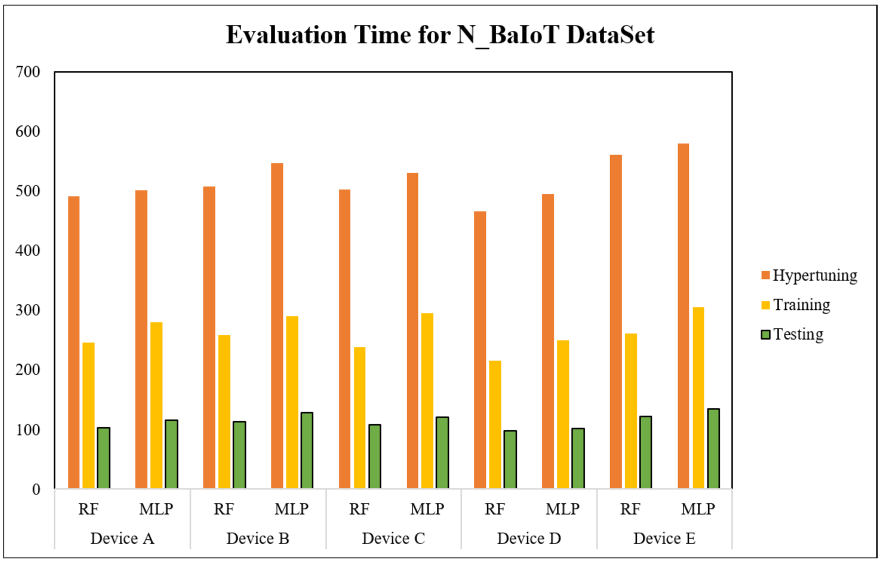


Fig.13. Evaluation time for N_BaIoT dataset

Table 5. Performance of device a: 'Danmini_Doorbell'

Metrics	Dev A					
	RF			MLP		
	Class A	Class B	Class C	Class A	Class B	Class C
Accuracy	1	1	1	0.99	0.98	0.98
Precision	1	1	1	0.99	0.98	0.98
Recall	1	1	1	1	0.99	0.99
F-score	1	1	1	0.99	0.98	0.98

Where **Class A** – Benign, **Class B** – Bashlite, **Class C** – Mirai

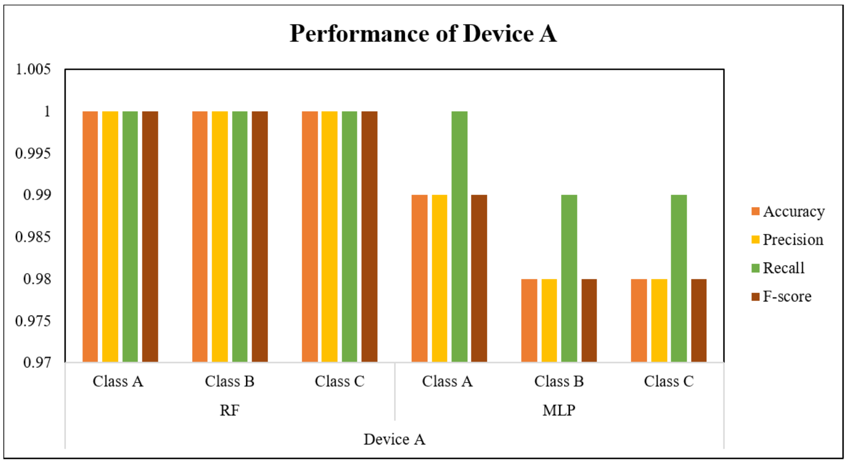


Fig.14. Performance of device a: 'Danmini_Doorbell

Table 6. Performance of device b: 'Ecobee_Thermostat'

Metrics	Dev B					
	RF			MLP		
	Class A	Class B	Class C	Class A	Class B	Class C
Accuracy	1	1	1	0.99	0.97	0.98
Precision	1	1	1	0.99	0.96	0.98
Recall	1	1	1	1	0.97	0.99
F-score	1	1	1	0.99	0.96	0.98

Where **Class A** – Benign, **Class B** – Bashlite, **Class C** – Mirai

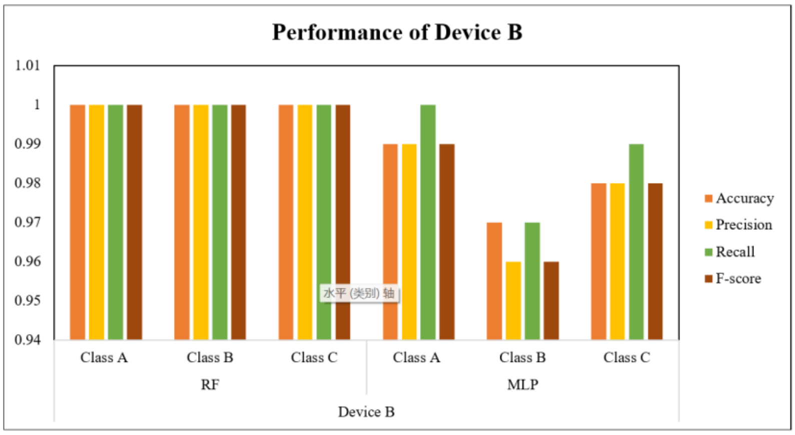


Fig.15. Performance of device b: 'Ecobee_Thermostat'

Table 7. Performance of device c: 'Philips_B120N10_Baby_Monitor'

Metrics	**Dev C**					
	RF			**MLP**		
	Class A	**Class B**	**Class C**	**Class A**	**Class B**	**Class C**
Accuracy	1	1	1	0.99	0.98	0.98
Precision	1	1	1	0.99	0.98	0.98
Recall	1	1	1	1	0.99	0.99
F-score	1	1	1	0.99	0.98	0.98

Where **Class A** – Benign, **Class B** – Bashlite, **Class C** – Mirai

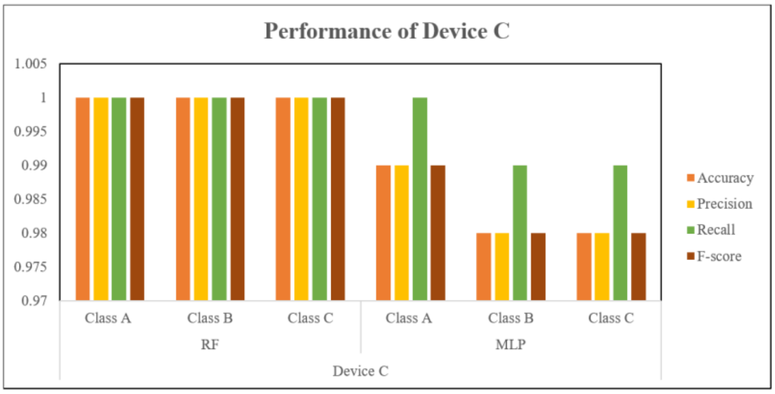


Fig.16. Performance of device c: 'Philips_B120N10_Baby_Monitor'

Table 8. Performance of device d: 'Samsung_SNH_1011_N_Webcam'

Metrics	**Dev D**			
	RF		**MLP**	
	Class A	Class B	Class A	Class B
Accuracy	1	1	0.99	0.98
Precision	1	1	0.99	0.98
Recall	1	1	1	0.99
F-score	1	1	0.99	0.98

Where **Class A** – Benign, **Class B** – Bashlite

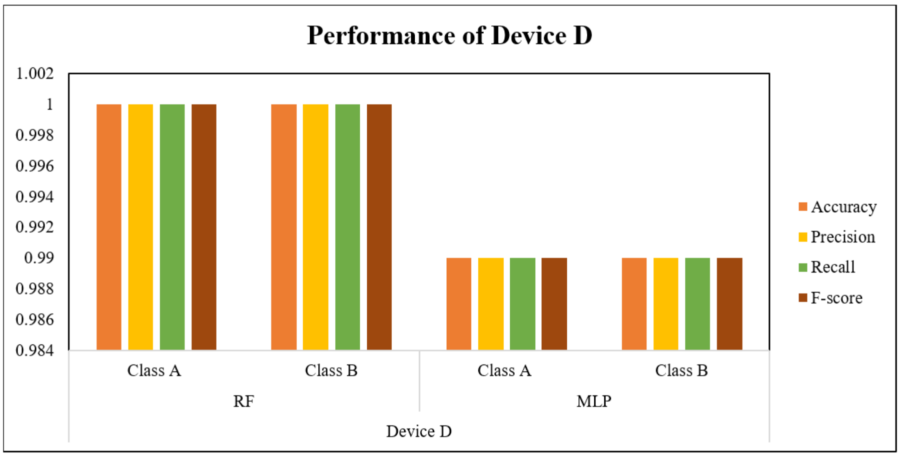


Fig.17. Performance of device d: 'Samsung_SNH_1011_N_Webcam'

Table 9. Performance of device e: 'SimpleHome_XCS7_1002_WHT_Security_Camera'

Metrics	Dev E					
	RF			MLP		
	Class A	Class B	Class C	Class A	Class B	Class C
Accuracy	1	1	1	0.99	0.97	0.98
Precision	1	1	1	0.99	0.96	0.98
Recall	1	1	1	1	0.97	0.99
F-score	1	1	1	0.99	0.96	0.98

Where **Class A** – Benign, **Class B** – Bashlite, **Class C** – Mirai

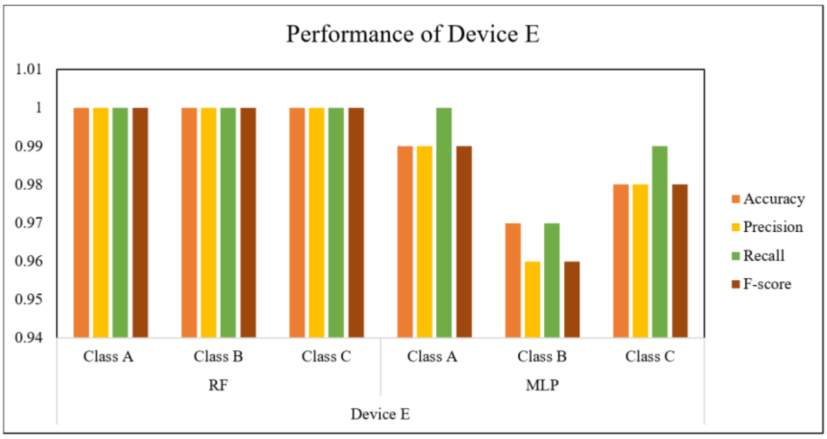


Fig.18. Performance of device e: 'SimpleHome_XCS7_1002_WHT_Security_Camera'

We can see from the findings that the RF classifier performs best, followed by the MLP-NN classifier. We may say that the bashlite botnet has a somewhat lower categorization rate than mirai and benign. It's hard to identify which attack had the best overall classification rate because the classification outcomes were all excellent. Another interesting finding from this data set is an impressive attack detection rate of more than 98% in most circumstances and as high as 100% when utilising the RF method alone. A very high precision and F1 score (one or very near to one) were achieved by both algorithms in almost all of the attacks.

5. Conclusions

Several experiments were conducted on the most extensively used IoT Intrusion IDS dataset using two supervised machine learning classifiers. An effective random forest classifier with more hyper-parameter optimization improves IoT network anomaly detection performance. The number of class labels was built and evaluated based on the number of trees within the ensemble, and two data sets namely UNSW-NB15 and N_BaIoT were included in the experiment. Random forest is proved to be statistically important when compared to other machine learning classifiers such as multi-layer perceptron neural networks, according to our research. Furthermore, with respect to accuracy as well as FAR metric, the suggested model outperformed other techniques discovered in the state-of-the-art method. By emloying a 10-fold cross validation method, it has produced an amazing outcome so far. Future study should focus on the robustness of ensemble classifiers in the face of a variety of incursions. Because ML algorithms have intrinsic flaws, attackers learn to develop intrusions which are more diverse and have the ability of disguising, evading, and even circumventing IDS capabilities alongwith additional network security measures as their study progresses. Protecting against an increase in assault frequency remains a serious difficulty and an elusive research goal. The goal of upcoming research is to create variants of the ML models that address all relevant security concerns.

References

[1] R. Primartha and B. A. Tama, "Anomaly detection using random forest: A performance revisited," Proc. 2017 Int. Conf. Data Softw. Eng. ICoDSE 2017, vol. 2018-Janua, pp. 1–6, 2018, doi: 10.1109/ICODSE.2017.8285847.

[2] M. Al-Zewairi, S. Almajali, and A. Awajan, "Experimental evaluation of a multi-layer feed forward artificial neural network classifier for network intrusion detection system," Proc. - 2017 Int. Conf. New Trends Comput. Sci. ICTCS 2017, vol. 2018-Janua, no. October, pp. 167–172, 2017, doi: 10.1109/ICTCS.2017.29.

[3] H. M. Anwer, M. Farouk, and A. Abdel-Hamid, "A framework for efficient network anomaly intrusion detection with features selection," 2018 9th Int. Conf. Inf. Commun. Syst. ICICS 2018, vol. 2018-Janua, pp. 157–162, 2018, doi: 10.1109/IACS.2018.8355459.

[4] S. Meftah, T. Rachidi, and N. Assem, "Network based intrusion detection using the UNSW-NB15 dataset," Int. J. Comput. Digit. Syst., vol. 8, no. 5, pp. 477–487, 2019, doi: 10.12785/ijcds/080505.

[5] M. Idhammad, K. Afdel, and M. Belouch, "DoS Detection Method based on Artificial Neural Networks," Int. J. Adv. Comput. Sci. Appl., vol. 8, no. 4, 2017, doi: 10.14569/ijacsa.2017.080461.

[6] Hajisalem and S. Babaie, "A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection," Comput. Networks, vol. 136, pp. 37–50, 2018, doi: 10.1016/j.comnet.2018.02.028.

[7] Y. Meidan et al., "N-BaIoT-Network-based detection of IoT botnet attacks using deep autoencoders," IEEE Pervasive Comput., vol. 17, no. 3, pp. 12–22, 2018, doi: 10.1109/MPRV.2018.03367731.

[8] J. O. Mebawondu, O. D. Alowolodu, J. O. Mebawondu, and A. O. Adetunmbi, "Network intrusion detection system using supervised learning paradigm," Sci. African, vol. 9, p. e00497, 2020, doi: 10.1016/j.sciaf.2020.e00497.

[9] V. Timčenko and S. Gajin, "Ensemble classifiers for supervised anomaly based network intrusion detection," Proc. - 2017 IEEE 13th Int. Conf. Intell. Comput. Commun. Process. ICCP 2017, pp. 13–19, 2017, doi: 10.1109/ICCP.2017.8116977.

[10] M. H. Kamarudin, C. Maple, T. Watson, and N. S. Safa, "A LogitBoost-Based Algorithm for Detecting Known and Unknown Web Attacks," IEEE Access, vol. 5, pp. 26190–26200, 2017, doi: 10.1109/ACCESS.2017.2766844.

[11] M. A. M. Aravind and V. K. G. Kalaiselvi, "Design of an intrusion detection system based on distance feature using ensemble classifier," 2017 4th Int. Conf. Signal Process. Commun. Networking, ICSCN 2017, pp. 16–21, 2017, doi: 10.1109/ICSCN.2017.8085661.

[12] N. Moustafa, B. Turnbull, and 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., vol. 6, no. 3, pp. 4815–4830, 2019, doi: 10.1109/JIOT.2018.2871719.

[13] J. Alsamiri and K. Alsubhi, "Internet of things cyber attacks detection using machine learning," Int. J. Adv. Comput. Sci. Appl., vol. 10, no. 12, pp. 627–634, 2019, doi: 10.14569/ijacsa.2019.0101280.

[14] Pandey, S. Thaseen, C. Aswani Kumar, and G. Li, "Identification of botnet attacks using hybrid machine learning models," Adv. Intell. Syst. Comput., vol. 1179 AISC, pp. 249–257, 2021, doi: 10.1007/978-3-030-49336-3_25.

[15] M. Alqahtani, H. Mathkour, and M. M. Ben Ismail, "IoT botnet attack detection based on optimized extreme gradient boosting and feature selection," Sensors (Switzerland), vol. 20, no. 21, pp. 1–21, 2020, doi: 10.3390/s20216336.

[16] Kanimozhi and T. P. Jacob, "Artificial Intelligence outflanks all other machine learning classifiers in Network Intrusion Detection System on the realistic cyber dataset CSE-CIC-IDS2018 using cloud computing," ICT Express, no. xxxx, 2020, doi: 10.1016/j.icte.2020.12.004.

[17] E. Özer, M. İskefiyeli, and J. Azimjonov, "Toward lightweight intrusion detection systems using the optimal and efficient feature pairs of the Bot-IoT 2018 dataset," Int. J. Distrib. Sens. Networks, vol. 17, no. 10, p. 155014772110522, 2021, doi: 10.1177/15501477211052202.

[18] Rathod, Nishit et al. "Model Comparison and Multiclass Implementation Analysis on the UNSW NB15 Dataset." *2021 International Conference on Computational Performance Evaluation (ComPE)*, 2021, doi:10.1109/ComPE53109.2021.9751832

[19] https://research.unsw.edu.au/projects/unsw-nb15-dataset

[20] Akash, Nazmus Sakib et al. "Botnet Detection in IoT Devices Using Random Forest Classifier with Independent Component Analysis." *Journal of Information and Communication Technology*, 2022, doi: 10.32890/jict2022.21.2.3

[21] https://archive.ics.uci.edu/ml/datasets/detection_of_IoT_botnet_attacks_N_BaIoT

[22] S. Joshi and E. Abdelfattah, "Efficiency of Different Machine Learning Algorithms on the Multivariate Classification of IoT Botnet Attacks," 2020 11th IEEE Annu. Ubiquitous Comput. Electron. Mob. Commun. Conf. UEMCON 2020, pp. 0517–0521, 2020, doi: 10.1109/UEMCON51285.2020.9298095.

[23] Y. Yin et al., "IGRF-RFE: A Hybrid Feature Selection Method for MLP-based Network Intrusion Detection on UNSW-NB15 Dataset," 2022.

[24] Q. Zhou, R. Li, X. Lei, H. Zhu, and W. Liu, "An Assessment of Intrusion Detection using Machine Learning on Traffic Statistical Data," vol. 14, no. 8, pp. 1–10, 2018.

[25] M. G. Desai, Y. Shi, and K. Suo, "IoT Bonet and Network Intrusion Detection using Dimensionality Reduction and Supervised Machine Learning," 2020 11th IEEE Annu. Ubiquitous Comput. Electron. Mob. Commun. Conf. UEMCON 2020, pp. 0316–0322, 2020, doi: 10.1109/UEMCON51285.2020.9298146.

[26] Bagui, Sikha et al. "Machine Learning Based Intrusion Detection for IoT Botnet." International Journal of Machine Learning and Computing, 2021, doi: 10.18178/ijmlc.2021.11.6.1068

Authors' Profiles

Priya Maidamwar is an Assistant Professor at G H Raisoni College of Engineering, Nagpur. She is working towards her doctoral degree in the area of network security. She has published more than 15 research papers in reputable journals and at national and international conferences. Her research focuses on machine learning, wireless sensor networks and network security.

Dr. Prasad P. Lokulwar is employed as an Associate Professor at G H Raisoni College of Engineering, Nagpur, Maharashtra His PhD in Computer Science & Engineering was granted by SGBAU Amravati University in Amravati. Additionally, he is responsible for more than 25 research papers that have been published in major national and international journals and conferences. His interests lie in the fields of network security and the internet of things.

Dr. Kailash Kumar is employed by Saudi Electronic University in Riyadh, the Kingdom of Saudi Arabia, as an Assistant Professor in the College of Computing and Informatics. In Jaipur, Rajasthan, India, at Suresh Gyan Vihar University, he earned a PhD in computer science and engineering. He has more than 100 research articles that have been published in international conferences and journals. The Internet of Things, cloud computing, machine learning, and image processing are some of Dr. Kailash Kumar's areas of interest in study.

How to cite this paper: Priya R. Maidamwar, Prasad P. Lokulwar, Kailash Kumar, "Ensemble Learning Approach for Classification of Network Intrusion Detection in IoT Environment", International Journal of Computer Network and Information Security(IJCNIS), Vol.15, No.3, pp.30-46, 2023. DOI:10.5815/ijcnis.2023.03.03