Bulletin of Electrical Engineering and Informatics
Vol. 13, No. 3, June 2024, pp. 1741~1751
ISSN: 2302-9285, DOI: 10.11591/eei.v13i3.6843

An extreme gradient boost based classification and regression tree for network intrusion detection in IoT

Silpa Chalichalamala[1], Niranjana Govindan[2], Ramani Kasarapu[3]
[1]Department of Computer Science and Engineering, SRM Institute of Science and Technology, Kattankulathur, Chennai, India
[2]Department of Computing Technologies, SRM Institute of Science and Technology, Kattankulathur, Chennai, India
[3]School of Computing, Mohan Babu University (Erstwhile Sree Vidyanikethan Engineering College), Tirupati, India

Article Info

Article history:

Received May 30, 2023
Revised Sep 11, 2023
Accepted Sep 28, 2023

Keywords:

Classification and regression tree
Cyber security
Extreme gradient boost
Internet of things
Network intrusion detection

ABSTRACT

Nowadays, modern technology includes various devices, networks, and apps from the internet of things (IoT), which consist of both positive and negative impacts on social, economic, and industrial effects. To address these issues, IoT applications and networks require lightweight, quick, and adaptable security solutions. In this sense, solutions based on artificial intelligence and big data analytics can yield positive outcomes in the realm of cyber security. This study presents a method called extreme gradient boost (XGBoost) based classification and regression tree to identify network intrusions in the IoT. This model is ideally suited for application in IoT networks with restricted resource availability because of its distributed structure and built-in higher generalization capabilities. This approach is thoroughly tested using botnet internet of things (BoT-IoT) new-generation IoT security datasets. All trials are conducted in a range of different settings, and a number of performance indicators are used to evaluate the effectiveness of the proposed method. The suggested study's findings provide recommendations and insights for situations involving binary classes and numerous classes. The suggested XGBoost model achieved 99.53% of accuracy in attack detection and 99.51% in precision for binary class and multiclass classifications, respectively.

Corresponding Author:

Niranjana Govindan
Department of Computing Technologies, SRM Institute of Science and Technology
Kattankulathur, Chennai, India
Email: niranjag@srmist.edu.in

1. INTRODUCTION

An intrusion detection system (IDS) consists of software and hardware components that monitor the operations and undesirable activities in security systems. Network intrusion detection systems (NIDS) and host intrusion detection systems (HIDS) are the two most popular IDS available in the field [1]. This research provides a NIDS to enhance the performance of the attack detection rate using ensemble machine learning [2]. The application of classification algorithms can be used to solve the important decision-making issue of network intrusion detection. In this subject intrusion detection systems, a number of machine learning techniques have been used which include neural networks, fuzzy logic, support vector machines, nave bayes, K nearest neighbors, and decision trees [3]. The goal of network intrusion detection systems is to identify assaults by analyzing network traffic. While these systems have typically operated using hard-coded rules, more research is being done to examine the use of machine learning (ML) [4]. IDS plays a major role in the internet of things (IoT) ecosystem for intrusion detection which gives alarms when an unexpected activity occurs [5]. The constant requirement for current definitions of various attacks is one of the most difficult

tasks in the fields of virus and intrusion detection. IDS used signature-based method to separate the abnormal traffic and ordinary traffic in botnet internet of things (BoT-IoT) [6] by comparing patterns found in the data flow under study with those recorded in a database of known attacks, NIDS can identify assaults.

An anomaly-based NIDS often finds anomalies by creating a model of the monitored system's typical behavior and identifying behavior that deviates from the model as abnormal or suspicious [7]. Systems that use saved signatures to identify intrusions operate by comparing them to incoming patterns. On the other side, anomaly-based IDS create typical pro-files and identify any deviations. Typically, a normal profile is created carefully by observing the ongoing actions of individuals, networks, and applications for a predetermined amount of time [8]. IDS uses sensors which are used to find the harmful behavior of crucial parts of NIDS [9]. The most well-known context of ML applications in the industry is network IDSs (NIDSs), which specifically detect harmful activities in networks. Anomaly-based IDSs can locate or detect unknown attacks, whereas signature-based IDSs are only capable of detecting well-known assaults with great accuracy [10]. Additionally, ML models aim to increase the efficiency of NIDS by decreasing the zero-error rate [11]. The accuracy, confusion matrix, recall, and precision measures are employed in this study to validate and examine the performance of the models that are developed [12]. The capacity to detect anomalies is significantly enhanced by the automatically learned essential elements that can more accurately represent traffic behavior. Combining the predictions of numerous base estimators has the advantage of increasing generalizability and robustness compared to using just one estimator [13]. This study employs a machine learning algorithm (MLA) based on NIDS to enhance detection performance. NIDSs are often used in external networks as a crucial security system element to spot hostile assaults that can get past firewalls and authentication procedures [14]. Even though many earlier efforts have had some success with IDS, intrusion detection is still a difficult subject because of the large amount of network traffic data [15]. NIDs protects networks by actively monitoring, analyzing network traffic to swiftly detect and respond to any malicious activities. It acts as a vigilant security system, safeguarding against cyber threats and ensuring network integrity and safety [16]. However, current techniques for detecting attacks in network systems, particularly those that depend on periodic models and extensive training data, resulting in challenges in achieving effective intrusion detection. Then, need to develop a system that can accurately classify real-time data and assess its effectiveness in achieving consistently high accuracy rates [17], [18]. The study of network intrusion detection is essential to support cybersecurity defenses. By understanding the evolving strategies of cyber threats and developing advanced detection mechanisms, it can safeguard critical information and infrastructure. This research will empower organizations to proactively counter cyber-attacks, ensuring data confidentiality, integrity, and availability, in an interconnected digital world [19], [20]. The objectives of this study are to explore the limitations of existing techniques in detecting attacks in network systems, specifically those reliant on periodic models and extensive training data by using classification methods. Then, to accurately classifying real-time data and evaluating its effectiveness in achieving high accuracy rates an NIDs based stream learning was used.

Alkadi *et al.* [21] identifiy the cyber-attacks a collaborative intrusion detection system based deep block chain framework (DBF). DBF was suggested to identify cyber-attacks by utilizing IDS-based on bidirectional long short-term memory (BiLSTM) techniques that can acquire at any range in time to protect private data using smart contract and privacy preservation-based blockchain. Hence, the DBF used to protect additional privacy assurances and security during the live relocation of virtual machines (VMs) in the cloud. However, DBF has disadvantages such as communication difficulty, which reflects the data cost of propagating a new block to all parties in a structure in each round. Shitharth *et al.* [22] developed an innovative clustering based classification methodology to precisely detect intrusions from the different types of IDS datasets such as NSL-KDD, CICIDS, and BoT-IoT. The intension of this research was to solve complex problems such as inefficiency in handling large dimensional datasets, high computational complexity, false detection, and more time consumption for training the models. The data separation was applied by forming the clusters by using an intelligent anticipated distance-based clustering (ADC) incorporated with the density-based spatial clustering of applications with noise (DBScan) algorithm. the most suitable optimal parameters are selected using the perpetual pigeon galvanized optimization (PPGO) technique. The likelihood naive bayes (LNB) classification approach is implemented to accurately predict the classified label as to whether normal or attack. However, the tendency of convergence to a local optimum is a drawback of this work.

Zeeshan *et al.* [23] identifies the malicious traffic attacks in IoT using a protocol based deep intrusion detection (PB-DID). PB-DID structure uses all of the information from the UNSW-NB15 and BoT-IoT data-sets by combined them to generate a train the LSTM based and single customized data-set for un-supervised deep techniques using 26 features. Hence, the PB-DID technique addresses the data imbalance issues and also minimises over fitting the training and testing datasets. However, due to security loopholes, attack detection might be challenging to identify the network flaws in PB-DID. Moghanian *et al.* [24]

discovered grasshopper optimization algorithm multi-layer perception (GOAMLP) enables more precise ANN learning to lower the error rate of intrusion detection. The GOAMLP techniques chooses applicable parameters like bias and weight to reduce the intrusion detection error in the neural network. GOAMLP achieves better than MLP and other IDS methods. To reduce intrusion detection inaccuracy, the GOAML algorithm has improved weight and bias. The GOAMLP allows for the consideration of this topic as an optimization problem with a minimization strategy. Kunhare *et al.* [25] implemented to identify unusual traffic, Natural language processing and ensemble-based machine learning (NLPIDS) are used. Using NLPIDS, a text corpus will be used to create vector spaces, and those vector spaces will be used to train machine learning models to find anomalies. However, this technique has classification issues and complicates finding a minor class occurrence in NLPIDS. The limitations found from the literature survey are communication difficulty, tendency of convergence to a local optimum, security loopholes, classification issues.

To address the limitations of existing methods such as communication difficulties, data cost, coverage tendencies, security loopholes, and optimization issues, the study proposes the utilization of extreme gradient boost (XGBoost) based classification and regression tree. This approach aims to enhance intrusion detection capabilities, offering improved accuracy and efficiency in handling real-time network data.

The contributions of the research are as:

- In this research, an XGBoost based classification and regression tree was suggested to detect the network intrusions in IoT.
- The classification accuracies of classifiers for batch and stream learning are accepted and assessed over time. This study demonstrates the existing techniques that are incapable of detecting attacks in network systems. Because it requires a periodic model and a large amount of training data.
- A novel intrusion detection-based stream learning was suggested to classify the accuracy time and it is used to achieve the accuracy rates in updating techniques.

The rest of the paper is organized as: section 2 discusses the explanation about the proposed method and its block diagram for cyber security in IoT. The proposed XGBoost and CART algorithm, the corresponding NIDS and pseudocode are described in section 3. The experimental evaluation results are presented in section 4, and section 5 concludes the paper.

2. PROPOSED METHOD

This research proposed classification and regression tree (CART) and XGBoost algorithm which contains the following stages such as pre-processing, feature selection and classification. This section gives a general overview of the machine learning and natural language processing-based strategy for detecting network intrusion. The suggested method's process is shown in Figure 1. To train an ensemble machine learning framework, these vector spaces are processed. The learned models are then used to identify abnormalities in the data that indicate network intrusion [26].

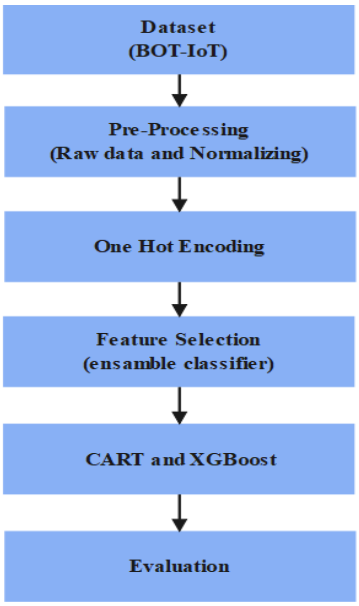

Figure 1. Block diagram for the proposed work

2.1. Datasets

This research first uses datasets to improve the detection effects, which vary in terms of the number of characteristics and instances. The BoT-IoT dataset are used to test the features of the proposed model.

BoT-IoT: the BoT-IoT datasets contain DDoS, DoS, OS and service scans, keylogging, and data filtration assaults, with the DDoS, and DoS attacks for protocol utilization. BoT-IoT is a term used to describe a collection of hacked computers, smart appliances, and internet-connected gadgets that have been commandeered for illegal uses [24].

2.2. Pre-processing

After collecting the data from BoT-IoT datasets, the preprocessing is used for altering the raw data and normalizing that are mentioned.

Normalization: in this study, four distinct attribute normalization approaches are introduced as a preprocessing step for data anomaly intrusion detection. Next, three techniques are used to compare the detection results on the normalized data. Security normalization is a process that identifies and gathers complete information associated with security. The process uses the information aggregated from a consumer's account and compares it against proprietary reference data was shown in (1):

$$X_n = (X - Xminimum)/(Xmaximum - Xminimum) \quad (1)$$

X_n=*Value of Normalization*

X_{max}=*Maximum value of a feature*

X_{min} = *Minimum value of a feature*

2.3. One hot encoding

One hot encoding [27] is the crucial process of transforming the variables in categorical data that will be fed into machine and deep learning algorithms, improving predictions and model classification accuracy. To convert symbolic features into numerical features, the datasets are handled using a single hot encoding technique. It was better able to calculate a probability for output values when using numeric values that are shown in (2):

$$v \in \{0,1\} \sum_{i=1}^{m} v_i = 1 \quad (2)$$

2.4. Feature selection for ensemble classifier

The ensemble classifier receives a sizable amount of the chosen features for network attack detection [28]. By enhancing regression feature selection, feature selection enhances network intrusion detection performance. It has the most powerful feature selection and is employed in a variety of circumstances. In feature selection, the lasso regression is also a popular technique for reducing the dimensionality of data, and errors in quantitative analysis and have gained increasing attention. Removing extraneous elements lowers the complexity of the data, which is very important to IDS. To shrink the dimensions of network data, feature selection techniques remove superfluous data an intrusion detection system's overall efficacy is considerably increased when the number of pertinent traffic attributes may be decreased without having a detrimental impact on classification accuracy showed in (3):

$$\gamma(\sigma, \alpha) = \int v\left(y, f\left((x * \sigma), \alpha\right)\right) dp\ (x, y) \quad (3)$$

On the other hand, a search through the collection of feature subsets using an induction algorithm's estimated accuracy as a yardstick for the usefulness of a given feature subset which is described as the alternative approach which is shown in (4):

$$T_{\omega rap}(\sigma, \alpha) = min_{\sigma} T_{alg}(\sigma) \quad (4)$$

3. CLASSIFICATION AND REGRESSION TREE (CART)

This study used the CART algorithm [29], which combines the benefits of many data detection algorithms to produce the best outcomes. The procedure of binary recursive partitioning is iterative and dividing the data into partitions and branches was used to construct regression trees. The CART algorithm

contains various branches and divisions that are further divided into smaller groups for classification and regression methods. The CART method does this by using the g index criterion to find the sub-nodes' best homogeneity. By taking into account the best attribute and threshold value, the root node is used as the training set and divided into two as shown in (5):

$$obj(\theta) = \sum_i^n \left(\left(y_{i,}, y_j \right) \right) + \sum_{k=1}^k \Omega \left(f_k \right) \quad (5)$$

where $obj(\theta)$ is the objective function, Ω is the regularization term, k is the number of trees, y_i is the prediction of instance i.
In CART, Gini index is a metric used for classification as given in (6):

$$Gini\ Index = 1 - \sum_{i=1}^c P_i \quad (6)$$

where c is the number of classes and P_i is the probability of each class in the dataset.

A decision tree algorithm always tries to maximize the value of information gain, and a node/attribute having the highest information gain is split first. It can be calculated using (7):

$$Information\ Gain = Entropy(S) - [(WeightedAvg) * Entropy\ (eachfeature)] \quad 7)$$

Theentropy is a metric to measure the impurity in a given attribute. It specifies randomness in data. Entropy can be calculated as shown in (8):

$$Entropy(s) = -P(yes)\ log\ 2\ P(yes) - P\ (no)\ log\ 2\ P(no) \quad (8)$$

3.1. Extreme gradient boost

XGBoost was proposed for NIDs due to its exceptional ability to handle complex, high-dimensional data and effectively address the challenges faced in intrusion detection, such as communication difficulties, data cost, coverage tendencies, security loopholes, and optimization problems. Its ensemble learning technique, leveraging decision trees, offers superior classification performance, making it a robust and efficient choice for detecting and mitigating cyber threats in real-time network environments. Furthermore, XGBoost has high-dimensional data handling, superior classification performance, robust against overfitting, efficient handling of missing data, and support for parallel processing, making it a reliable and effective tool for cyber threat detection. The gradient-boosted trees approach is implemented using the open-source software known as XGBoost, which stands for extreme gradient boosting [30]. Due to its accuracy and simplicity, it has been one of the most used machine-learning techniques in kaggle tournaments. It's a supervised learning method that may be used for classification or regression issues. Gradient boosting makes simple to grasp XGBoost, because this method employs decision trees as a “weak” predictor. In addition, its implementation was specially designed for the best speed and performance. For structured and tabular data, XGBoost performs well in neural networks that are a typically better choice for working with unstructured data, such as images that are expressed in (9):

$$P_{i(t)} = [\ P_{i(t)}^{eta}, P_{i(t)}^{max\ \_depth}, P_{i(t)}^{max\ \_child\_weight}, P_{i(t)}^{gamma}, P_{i(t)}^{subsample}, P_{i(t)}^{colsample\_bytree}\ ] \quad (9)$$

For usage in logistic regression, the logistic loss is another frequently utilized loss function shown in (10):

$$V_{i(t)} = [\ V_{i(t)}^{eta}, V_{i(t)}^{max\ \_depth}, V_{i(t)}^{max\_child\_weight}, V_{i(t)}^{gamma}, V_{i(t)}^{subsample}, V_{i(t)}^{colsample\_bytree}\ ] \quad (10)$$

The XGboost model can be expressed mathematically as shown in (11):

$$F_{i(t)} = (P_{i(t)} \rightarrow Xgboost(trainingset)_{[metric=R_{curve}} \quad (11)$$

The personal best and global best are mathematically shown in (12) and (13):

$$pbest_{i(t)} = max\left(F_{i(j)}\right), 0 \leq j \leq t \quad (12)$$

$$Gbest_{i(t)} = max\left(pbest_{K(t)}\right), 1 \leq K \leq m \quad (13)$$

$f_n = fibonacci\ sequence$

$f_k = functional\ constant$

3.2. XGBoost–CART hybridization

The CART algorithm was proposed for NIDs due to its ability to handle both classification and regression tasks effectively. It offers a clear and interpretable decision tree structure, aiding in identifying important features for attack detection. Its adaptability to different types of data and simplicity make it a valuable choice for network intrusion detection applications. Furthermore, the advantages of CART for NIDs has interpretable decision trees facilitate understanding of attack patterns, suitability for both classification and regression tasks, ability to handle large datasets, and adaptability to diverse data types, enhancing overall intrusion detection effectiveness. The proposed hybrid XGBoost and CART possess the advantage of better learning, and improved classification accuracy based on the split criteria. It has the objective of information gain and classifying less occurred data with a gradient weighted model with a lesser estimator size. The constructor of the XgBoost classifier will call the CART tree function to create its estimator with its logistic objective model that has a great influence on the network model to improve the accuracy.

3.3. Pseudo code

In the pseudo code, the combinational algorithm was discovered by using a proposed model that are showed in below pseudo code.

Pseudo Code

```
Input: Features
Output: Intrusion Attacks
For each feature f in Data D
        Perform data normalization
        Do Computer w // Lasso Regression
        while
              For j=1,2,….N do
                    Update the value of w
        Do (check converged)
Initialize training data instance space S
For t=1,2,,…T do
        Train a weak learner h :X --> R using the distribution D // XG Boost
        Determine the weight a of h
        Update the distribution over the training of data
End for
Compute the final score for the instances
Create Node N based on the final score
If samples s in S are all of the same class C Then
        Return N with the label C
End if
If A attribute is not null || the value of attributes is the same as another
instance S
        Label the class as the majority of S instances
End if
Find the best splitting attribute a in A using the attribute selection method
        For a' in a:
              Label node N with splitting criterion with S' equals S which
              represents a' equals a
              If S' is null
                    Attach a leaf with majority of leaf class in S to node N
              Else
                    Attach the node returned by tree T to N
              End if
        End for
Evaluate the performance metrics
```

Explanation:

All of the features are being read by the data, and each feature is being normalised. After conducting min max normalisation, lasso regression is used to calculate the weight value for each piece of data. In a while loop, we are updating the lasso convergence process for each weight value. The chosen features from lasso are then used as a training dataset depending on the convergence. The decision tree model is given the training data, and the distribution of the data in XGBoost is used to identify the weak learners. This distribution is being used to calculate the weights for each prediction A. The XGBoost method will be used

to update the prediction distribution. Therefore, all of the weak learners will be strengthened for better data training. This is how the overall score of the cases used to create the prediction score is determined. For this instance, all test samples for each attribute are calculated. That specific property is chosen for that instance based on which classes have the majority of instances. The majority classes of leaves are chosen as the class value of the test data based on this. To do this, performance metrics evaluation is calculated.

4. RESULT AND DISCUSSION

The features of the suggested network model is evaluated using the metrics like accuracy, precision, F1-score, recall and false alarm rate (FAR) as shown in Table 1, where TP, TN, FP, and FN stand for true positives, true negatives, false positives, and false negatives respectively. The first step in binary classification is to distinguish between communications are harmful. The performance of the proposed work is showed in the Table 1 and the performance metrics for binary and multi-class classification is shown in Tables 2, 3, and 4 respectively.

Table 1. Performance metrics of the proposed work

Metric(%)	Definition
Accuracy	$\frac{TP+TN}{TP+FP+TN+FN}$
Precision	$\frac{TP}{TP+FP}$
Recall	$\frac{TP}{TP+FN}$
FAR	$\frac{FPR+FNR}{2}$
F1-score	$\frac{2\times Recall\times Precision}{TP+TN}$

Table 2. The performance measures obtained for the proposed technique for binary classes in terms of precision, recall, F1 measure, accuracy, and AUC

Model type class	Class	Precision (%)	Recall (%)	F1 measure (%)	Accuracy (%)	AUC (%)
Random forest	Attack	99.28	99.34	99.35	99.34	97.35
	Normal	99.26	36.51	52.42	99.34	97.35
Naive bayes	Attack	99.44	99.82	99.36	99.63	70.35
	Normal	03.62	90.25	06.54	99.63	70.35
Decision tree	Attack	99.43	99.56	99.61	99.71	91.86
	Normal	95.68	36.93	54.43	99.71	91.86
Proposed XGBoost–CART	Attack	99.65	99.84	99.54	99.89	100
	Normal	99.85	93.42	97.69	99.81	100

Table 3. Results evaluation for the proposed technique for multi classes in terms of category

Model type class	Class	Precision (%)	Recall (%)	F1 measure (%)	Accuracy (%)	AUC (%)
Random forest	Normal	98.56	97.41	97.17	97.31	97.35
	DDos	96.33	98.34	97.15	97.55	97.35
	DoS	99.29	23.45	38.46	97.96	97.35
	Reconnaissance	99.24	97.26	98.24	97.24	97.35
	Theft	0	0	0	0	97.35
Naive bayes	Normal	66.46	96.45	78.25	71.51	70.35
	DDos	89.24	44.86	59.56	71.36	70.35
	DoS	03.42	90.12	06.37	71.43	70.35
	Reconnaissance	89.23	21.54	33.82	71.84	70.35
	Theft	99.34	57.48	73.46	71.72	91.86
Decision tree	Normal	97.36	88.46	92.34	91.14	91.86
	DDos	86.42	96.73	91.55	91.22	91.86
	DoS	94.25	31.54	46.92	91.43	91.86
	Reconnaissance	94.54	57.22	71.44	91.68	91.86
	Theft	0	0	0	91.83	92.45
Proposed XGBoost – CART	Normal	99.63	99.45	99.83	99.51	100
	DDos	99.55	99.63	99.45	99.35	100
	DoS	99.34	93.44	97.25	99.44	100
	Reconnaissance	99.49	99.18	99.32	99.73	100
	Theft	99.56	99.71	97.15	99.64	100

Table 4. Comparison table for BoT-IoT dataset

BoT-IoT dataset	Accuracy (%)	Precision (%)	F1-score (%)	Recall (%)	FAR (%)	AUC (%)
Random forest	78.012	78.683	66.202	63.292	3.504	97.356
Decision tree	91.464	74.515	60.455	54.794	2.092	91.972
Naïvebayes	71.571	69.532	50.293	62.095	0.611	74.652
Proposed XGBoost–CART	99.998	99.512	98.608	98.282	0.352	100

The performance and the graphical representations for binary classification of BoT-IoT dataset in terms of accuracy, precision, recall, AUC and F1-score was shown in Table 2 and Figure 2. The results from the Figure 2 showed that CART and XGBoost for binary classification perform better results in all the performance metrics when compare with other classifiers in attack and normal classes. The performance and the graphical representations for mult-class classification of BoT-IoT dataset in terms of accuracy, precision, recall, AUC and F1-score was shown in Table 3 and Figure 3.

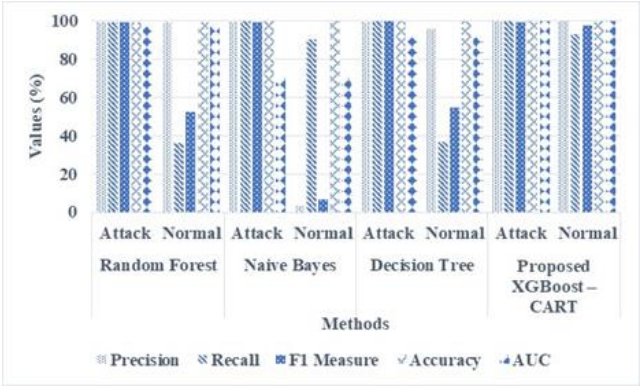

Figure 2. Graphical representations for binary classes in terms of precision, recall, F1 measure, accuracy, and AUC

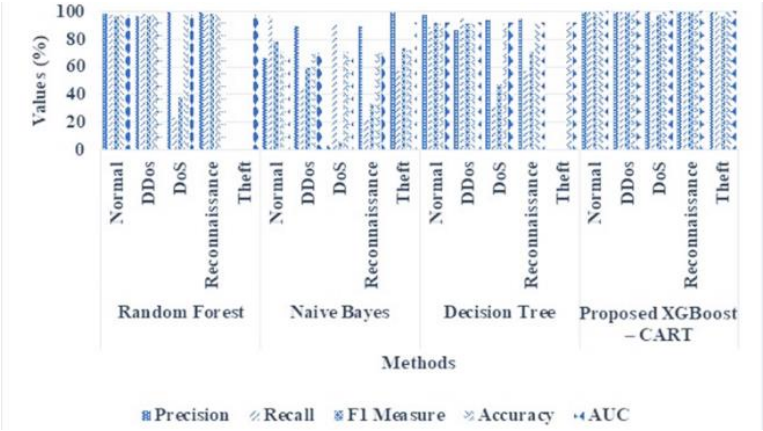

Figure 3. Graphical representations for multi classes in terms of category

The results from Figure 3 showed that CART and XGBoost for multi-class category classification perform better results in all the performance metrics when compare with other classifiers in terms of normal, DDoS, DoS, reconnaissance, and theft classes. Table 4 shows the comparative analysis of BoT-IoT dataset. The results showed that CART and XGBoost for multi class classification perform better results in all performance metrics when compare with other classifiers mentioned in Figure 4. Table 5 showed the suggested XGBoost model achieved 99.998% accuracy in attack detection, and 99.512% of precision, and gives better performance when compared to all the methods.

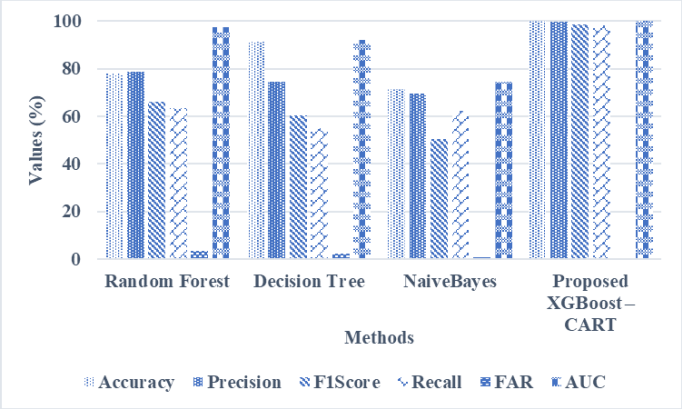

Figure 4. Performance of BoT-IoT dataset

Table 5. Comparison table for proposed work

Models	Dataset	Accuracy (%)	Precision (%)
Alkadi *et al.* [21]	BoT-IoT	98.910	-
Shitharth *et al.* [22]		99.995	99.245
Zeeshan *et al.* [23]		96.310	-
Proposed method		99.998	99.512

4.1. Discussion

According to the results, the existing methods like deep block chain framework [21], perpetual pigeon galvanized optimization–likelihood naïve bayes (LNB) [22] and protocol based deep intrusion detection [23] was compared with proposed method in terms of accuracy and precision. The XGBoost based classification and regression Tree was suggested to identify network intrusions in the IoT. XGBoost is an ensemble learning method for NIDs that combines multiple decision trees, boosting their performance through gradient boosting techniques, to achieve high accuracy and efficiency. CART algorithm creates a binary decision tree for NIDs, recursively splitting data based on the most significant features, resulting in a tree-based model for intrusion detection. From the observations of Table 5, Alkadi *et al*. [21] has achieved 98.910% accuracy in attack detection using DBF method. However, the DBF method has disadvantages such as communication difficulty, which reflects the data cost of propagating a new block to all parties in a structure in each round. Shitharth *et al*. [22] has achieved 99.995% accuracy and 99.245% precision in attack detection by using PPGO algorithm. However, the tendency of convergence to a local optimum is a drawback of this work. Zeeshan *et al*. [23] has achieved 96.310% of accuracy. However, due to security loophole the attach detection might be challenging to identify the network flaws in PB-DID. The proposed method has achieved 99.998% accuracy and 99.512% precision with efficient communication, low convergence to local optimum and security measures.

5. CONCLUSION

In this paper, a unique technique for network intrusion detection-based cyber security using XGBoost and CART is suggested. This method is used to capture attack flow detection by using a network flow graph edge properties and topological structure. This research concentrates on the usage of XGBoost and CART for the identification of dangerous network traffic in IoT networks. Here, four IoT NIDS benchmark datasets are used for experimental testing, and the results show that XGBoost and CART-based NIDS performs well when compared to state-of-the-art ML-based classifiers. The discovered intrusion detection has sustainable findings for the XGBoost model achieved 99.998% accuracy in attack detection and 99.512% of precision for binary class and multiclass classifications, respectively.

However, to enhance predictive accuracy, the suggested XGBoost and CART algorithms requires fine-tuning learning rates, optimizing tree depth, and addressing class imbalances. Future improvements can be achieved by experimenting with diverse techniques, aiming to maximize accuracy by refining hyper parameters and incorporating with innovative methods.

REFERENCES

[1] R. Qaddoura, A. M. Al-Zoubi, H. Faris, and I. Almomani, "A multi-layer classification approach for intrusion detection in iot networks based on deep learning," *Sensors*, vol. 21, no. 9, p. 2987, Apr. 2021, doi: 10.3390/s21092987.

[2] S. Sandosh, V. Govindasamy, and G. Akila, "Enhanced intrusion detection system via agent clustering and classification based on outlier detection," *Peer-to-Peer Networking and Applications*, vol. 13, no. 3, pp. 1038–1045, May 2020, doi: 10.1007/s12083-019-00822-3.

[3] V. 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*, vol. 7, no. 3, pp. 366–370, Sep. 2021, doi: 10.1016/j.icte.2020.12.004.

[4] A. Rosay, K. Riou, F. Carlier, and P. Leroux, "Multi-layer perceptron for network intrusion detection: From a study on two recent data sets to deployment on automotive processor," *Annales des Telecommunications/Annals of Telecommunications*, vol. 77, no. 5–6, pp. 371–394, Jun. 2022, doi: 10.1007/s12243-021-00852-0.

[5] Z. Hu, L. Wang, L. Qi, Y. Li, and W. Yang, "A Novel Wireless Network Intrusion Detection Method Based on Adaptive Synthetic Sampling and an Improved Convolutional Neural Network," *IEEE Access*, vol. 8, pp. 195741–195751, 2020, doi: 10.1109/ACCESS.2020.3034015.

[6] S. Rajagopal, P. P. Kundapur, and K. S. Hareesha, "A Stacking Ensemble for Network Intrusion Detection Using Heterogeneous Datasets," *Security and Communication Networks*, vol. 2020, pp. 1–9, Jan. 2020, doi: 10.1155/2020/4586875.

[7] S. I. Pérez, S. Moral-Rubio, and R. Criado, "A new approach to combine multiplex networks and time series attributes: Building intrusion detection systems (IDS) in cybersecurity," *Chaos, Solitons and Fractals*, vol. 150, p. 111143, Sep. 2021, doi: 10.1016/j.chaos.2021.111143.

[8] C. Yin, Y. Zhu, S. Liu, J. Fei, and H. Zhang, "Enhancing network intrusion detection classifiers using supervised adversarial training," *Journal of Supercomputing*, vol. 76, no. 9, pp. 6690–6719, Sep. 2020, doi: 10.1007/s11227-019-03092-1.

[9] Y. K. Saheed, A. I. Abiodun, S. Misra, M. K. Holone, and R. Colomo-Palacios, "A machine learning-based intrusion detection for detecting internet of things network attacks," *Alexandria Engineering Journal*, vol. 61, no. 12, pp. 9395–9409, Dec. 2022, doi: 10.1016/j.aej.2022.02.063.

[10] J. Ribeiro, F. B. Saghezchi, G. Mantas, J. Rodriguez, and R. A. Abd-Alhameed, "HIDROID: Prototyping a Behavioral Host-Based Intrusion Detection and Prevention System for Android," *IEEE Access*, vol. 8, pp. 23154–23168, 2020, doi: 10.1109/ACCESS.2020.2969626.

[11] M. Abdel-Basset, H. Hawash, R. K. Chakrabortty, and M. J. Ryan, "Semi-Supervised Spatiotemporal Deep Learning for Intrusions Detection in IoT Networks," *IEEE Internet of Things Journal*, vol. 8, no. 15, pp. 12251–12265, Aug. 2021, doi: 10.1109/JIOT.2021.3060878.

[12] C. F. T. Pontes, M. M. C. De Souza, J. J. C. Gondim, M. Bishop, and M. A. Marotta, "A New Method for Flow-Based Network Intrusion Detection Using the Inverse Potts Model," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1125–1136, Jun. 2021, doi: 10.1109/TNSM.2021.3075503.

[13] M. A. Rahman, A. T. Asyhari, L. S. Leong, G. B. Satrya, M. H. Tao, and M. F. Zolkipli, "Scalable machine learning-based intrusion detection system for IoT-enabled smart cities," *Sustainable Cities and Society*, vol. 61, p. 102324, Oct. 2020, doi: 10.1016/j.scs.2020.102324.

[14] L. Yang, A. Moubayed, and A. Shami, "MTH-IDS: A Multitiered Hybrid Intrusion Detection System for Internet of Vehicles," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 616–632, Jan. 2022, doi: 10.1109/JIOT.2021.3084796.

[15] P. Kumar, G. P. Gupta, and R. Tripathi, "Design of Anomaly-Based Intrusion Detection System Using Fog Computing for IoT Network," *Automatic Control and Computer Sciences*, vol. 55, no. 2, pp. 137–147, Mar. 2021, doi: 10.3103/S0146411621020085.

[16] A. Boukhalfa, A. Abdellaoui, N. Hmina, and H. Chaoui, "LSTM deep learning method for network intrusion detection system," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 3, pp. 3315–3322, Jun. 2020, doi: 10.11591/ijece.v10i3.pp3315-3322.

[17] A. J. Mohammed, M. H. Arif, and A. A. Ali, "A multilayer perceptron artificial neural network approach for improving the accuracy of intrusion detection systems," *IAES International Journal of Artificial Intelligence*, vol. 9, no. 4, pp. 609–615, Dec. 2020, doi: 10.11591/ijai.v9.i4.pp609-615.

[18] F. A. Al-Ibraheemi, S. Al-Ibraheemi, and H. Amintoosi, "A hybrid method of genetic algorithm and support vector machine for DNS tunneling detection," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 2, pp. 1666–1674, 2021, doi: 10.11591/ijece.v11i2.pp1666-1674.

[19] T. B. Seong, V. Ponnusamy, N. Z. Jhanjhi, R. Annur, and M. N. Talib, "A comparative analysis on traditional wired datasets and the need for wireless datasets for IoT wireless intrusion detection," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 22, no. 2, pp. 1165–1176, May 2021, doi: 10.11591/IJEECS.V22.I2.PP1165-1176.

[20] I. Wahidah, Y. Purwanto, and A. Kurniawan, "Collaborative intrusion detection networks with multi-hop clustering for internet of things," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 4, pp. 3255–3266, Aug. 2021, doi: 10.11591/ijece.v11i4.pp3255-3266.

[21] O. Alkadi, N. Moustafa, B. Turnbull, and K. K. R. Choo, "A Deep Blockchain Framework-Enabled Collaborative Intrusion Detection for Protecting IoT and Cloud Networks," *IEEE Internet of Things Journal*, vol. 8, no. 12, pp. 9463–9472, Jun. 2021, doi: 10.1109/JIOT.2020.2996590.

[22] S. Shitharth, P. R. Kshirsagar, P. K. Balachandran, K. H. Alyoubi, and A. O. Khadidos, "An Innovative Perceptual Pigeon Galvanized Optimization (PPGO) Based Likelihood Naïve Bayes (LNB) Classification Approach for Network Intrusion Detection System," *IEEE Access*, vol. 10, pp. 46424–46441, 2022, doi: 10.1109/ACCESS.2022.3171660.

[23] M. Zeeshan *et al.*, "Protocol-Based Deep Intrusion Detection for DoS and DDoS Attacks Using UNSW-NB15 and Bot-IoT Data-Sets," *IEEE Access*, vol. 10, pp. 2269–2283, 2022, doi: 10.1109/ACCESS.2021.3137201.

[24] S. Moghanian, F. B. Saravi, G. Javidi, and E. O. Sheybani, "GOAMLP: Network Intrusion Detection with Multilayer Perceptron and Grasshopper Optimization Algorithm," *IEEE Access*, vol. 8, pp. 215202–215213, 2020, doi: 10.1109/ACCESS.2020.3040740.

[25] N. Kunhare, R. Tiwari, and J. Dhar, "Particle swarm optimization and feature selection for intrusion detection system," *Sadhana - Academy Proceedings in Engineering Sciences*, vol. 45, no. 1, p. 109, Dec. 2020, doi: 10.1007/s12046-020-1308-5.

[26] M. G. Karthik and M. B. M. Krishnan, "Hybrid random forest and synthetic minority over sampling technique for detecting internet of things attacks," *Journal of Ambient Intelligence and Humanized Computing*, Mar. 2021, doi: 10.1007/s12652-021-03082-3.

[27] T. Al-Shehari and R. A. Alsowail, "An insider data leakage detection using one-hot encoding, synthetic minority oversampling

and machine learning techniques," *Entropy*, vol. 23, no. 10, p. 1258, Sep. 2021, doi: 10.3390/e23101258.
[28] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, "Building an efficient intrusion detection system based on feature selection and ensemble classifier," *Computer Networks*, vol. 174, p. 107247, Jun. 2020, doi: 10.1016/j.comnet.2020.107247.
[29] D. H. Lee, S. H. Kim, and K. J. Kim, "Multistage MR-CART: Multiresponse optimization in a multistage process using a classification and regression tree method," *Computers and Industrial Engineering*, vol. 159, p. 107513, Sep. 2021, doi: 10.1016/j.cie.2021.107513.
[30] O. Sagi and L. Rokach, "Approximating XGBoost with an interpretable decision tree," *Information Sciences*, vol. 572, pp. 522–542, Sep. 2021, doi: 10.1016/j.ins.2021.05.055.

BIOGRAPHIES OF AUTHORS

Silpa Chalichalamala completed B.Tech in Information Technology and M. Tech in Software Engineering from Sree Vidyanikethan Engineering College, Tirupati, India. She is a research Scholar at SRMIST, Chennai, India and working as Assistant Professor at Mohan Babu University Tirupati, India. Her research interests include internet of thing, block chain technology, and machine learning. She can be contacted at email: silpa.c8@gmail.com.

Niranjana Govindan received the B.E. degree from Madras University M.Tech. and Ph.D. degrees in Computer Science and engineering from SRM University. She is having nearly 20 years of teaching experience. She is currently Professor with the Department of CSE, SRM Institute of Science and Technology. She has published around 20 Scopus indexed articles. Her research interests include networking, machine learning, deep learning, and image processing. She has received the Young Investigator Award in 2012 and the IET Women Engineer Award in 2018. She can be contacted at email: niranjag@srmist.edu.in.

Kasarapu Ramani is currently working as Professor and Program head (Data Science) in Mohan Babu University (Erstwhile Sree Vidyanikethan Engineering College), Tirupati. She received her B.Tech.(ECE) degree from S.V. University, Tirupati, M.Tech.(CS) from JNTU University, Hyderabad and Ph.D. (CSE) from JNTUK University, Kakinada. She published 28 papers in National and International Journals. Also, she presented 23 papers in National and International Conferences. She authored 03 books and 04 book chapters. Her research interests include image processing, data science and cloud computing. She can be contacted at email: head-ds@mbu.asia.