International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Ensembled Gradient Boosting Technique with Decision Tree for Intrusion Detection System

V. S. Stency[1], **Dr. N. Mohanasundaram**[2], **Dr. R. Santhosh**[3]

Submitted: 03/02/2024 **Revised**: 11/03/2024 **Accepted**: 17/03/2024

Abstract: There is a rising need for network attack analysis in today's world as cyber threats and assaults multiply. Cloud computing is chosen by companies all over the world because of the scalability and versatility of its internet-based computer capabilities. Scientists are increasingly concentrating on the security of cloud data, and one of their key priorities is safeguarding hosts, businesses, and data against more sophisticated digital attacks. Numerous approaches have been developed as a consequence of researchers' experiments with Intrusion Detection (ID) architecture during the past few decades. But eventually, the intrusion detection framework won't be able to use these techniques. The goal of this study is to classify whether or not a framework interruption has happened using an ensemble model of an effective gradient-boosting decision tree (EGDT-boost). Using a Gradient Boosting classifier and a Decision tree, this model produced an ensemble classifier. The Decision Tree classifier performs better thanks to the gradient boosting techniques since fewer mistakes are recognised. The suggested classifier is examined in this article together with several other well-established classification methods. In comparison to previous approaches, the suggested model yields better outcomes in terms of Precision, Recall, F-Measure, and Accuracy.

Keywords: *Support Vector Machine, Naïve Biase, Adaboost, Convolutional Neural Network, Recurrent Neural Network., Hybrid Classifier, Ensemble Classifier, Ensemble Neural Network*

1. Introduction

Cybersecurity is also named computer or information security. The essential part of work safeguards the attacker's network and computer information [1]. The term is applied to various security aspects like network security, application security, information security, etc. These security factors are tangled to protect against the attacker with security principles of authentication, integrity, confidentiality, non-reputation, availability, etc [2-8]. Each year, more digital information penetrates the global cyber threat landscape, which continues to evolve rapidly. In the first nine months of 2019, information breaches exposed an astounding 7.9 billion data, according to research from Risk Based Security. Compared to the number of records found over the same period in 2018, this figure is more than twice [9-12].

Health administrations, merchants, and public entities saw many data breaches. In some of these regions, hackers are more prevalent because they collect financial and medical data. Yet, all businesses, even utilization organizations, might concentrate on client data, hidden company actions, or client attacks [13-17]. According to the International Data Corporation, the cost of network security solutions will skyrocket to $133.7 billion by 2022 as the scale of the digital threat is expected to grow. In response to the expanding digital danger, governments worldwide have issued guidelines to assist enterprises in developing appropriate network security processes [18].

In the United States, a digital security system has been established by the National Institute of Standards and Technology (NIST). The design suggests continual monitoring of all electronic resources to counteract the propagation of malicious code and help in early identification. The National Cyber Security Centre of the British government reiterates the relevance of framework checking throughout its "10 steps to network safety" guide. The Australian Cyber Security Centre (ACSC) frequently disseminates guidelines in Australia on how organizations may fight the most current cyber-security threats [20].

2. Literature Review

Shen et al. The main classifier of the ensemble-based ID System. suggested by 81 was the Extreme Learning Machine (ELM). classifier. During the ensemble pruning stage of the recommended approach, a BAT optimization technique is applied to optimize it further. Several datasets were used to evaluate the model, including the KDD Cup'99, NSL-KDD, and Kyoto datasets. Experiments indicated that many E.L.M.s working together in an ensemble outperform a single E.L.M. in terms of performance.

[1]*Research Scholar, Department of CSE, Karpagam Academy of Higher Education,*
Coimbatore - 641021, Tamil Nadu, India.
[2]*Professor, Department of CSE, Faculty of Engineering, Karpagam Academy of Higher*
Education, Coimbatore - 641021, Tamil Nadu, India.
[3]*Professor, Department of CSE, Faculty of Engineering, Karpagam Academy of Higher*
Education, Coimbatore - 641021, Tamil Nadu, India.
E-mail: stenz.denz@gmail.com

According to Gao et al. 82, an adaptive ensemble model containing various basic classifiers such as D.T., R.F., K.N.N., and Deep Neural Network (DNN) was created. An adaptive voting process was used to choose the top classifier. The suggested technique was verified using the NSL-KDD dataset in a series of experiments. By comparing the results of the trials to those of other models, it was proved that the performance efficiency was high. It was determined that the suggested technique did not provide adequate outcomes for the most susceptible attack classes.

T.T.K. (TalaTalaeiKhoei) The authors examine three types of ensemble learning: bagging, boosting, and stacking. They compared their results to those obtained using three common machine learning methods: The naive Bayes algorithm, decision tree, and K-nearest neighbour. The recommended methodologies will undergo training, evaluation, and rigorous testing. We used the CICDos 2019 benchmark for our analysis, which includes multiple DDoS attacks. We use a pair of feature-selection strategies to zero down on the most crucial characteristics. The detection probability, false alarm probability, missed detection probability, and detection accuracy are all factors in a performance assessment. Computer simulations show that stacking-based ensemble learning strategies beat competing algorithms on all four metrics...

Tama, BayuAdhi, and SunghoonLim(2021) A novel classifier ensemble strategy for anomaly-based intrusion detection systems: report and analysis of an empirical assessment (SoE). SoE is a parallel ensemble classifier that combines the strengths of three individual ensemble learners in a unified framework. Random forests, boosting machines, and even more powerful boosting machines are called "ensemble learners.". Statistical significance of classification algorithms may be determined using a variety of metrics, including area under the receiver operating characteristic curve, true positive rate, false-positive rate, and Matthews correlation coefficients. Our work fills a need in the literature by presenting an up-to-date systematic mapping examination and a comprehensive empirical assessment of the most current improvements in ensemble learning approaches applied to I.D.S.s.

Abhishek Divekar et al. (2018) assessed and compared the performance of KDD'99 possibilities using classification techniques like Naive Bayes, K-means, Neural Network, R-F, SVM, and D.T. UNSW-NB15 was considered to be a superior and more contemporary alternative to KDD'99. Regarding the f1-score, the experiment's findings indicated that the trained classifiers outperformed those trained using KDD'99 and NSL-KDD.

Using the association rule mining concept, the authors of (B. B. DipaliGangadhar Mogal, 2017) proposed a technique for selecting the most optimum attributes, which they termed "association rule mining.". The central point approach was utilized to determine attribute values, and the Apriori algorithm was used to narrow the scope of the study. Logistic regression and naive Bayes methods were used for the evaluation. Based on the results, the central point method coupled with the apriori algorithm is optimal concerning accuracy and computational cost. The study was conducted on the NSL-KDD and UNSW-NB15 datasets.

They attempted to analyze the performance and efficiency of NIDS according to the authors (Srivastava, 2018). They applied two feature reduction strategies to obtain these results: LDA and C.C.A. The researchers utilized seven classifiers, each with parameters and measurement metrics, F.P.R., Training length, precision, accuracy, and receiver operating characteristic curve area (R.O.C.). Random, naive Bayes, rep tree, R.F., random committee, bagging randomizable, and filtered sampling are just a few options. The study indicates that the UNSW-NB15 findings achieved using LDA and random trees were superior.

The authors (H.M. Anwer, 2018) employed wrapper and filter feature selection techniques in their case study to discover the lowest feasible range of effective features while maintaining the highest level of accuracy. The UNSW-NB15 dataset was categorized using the J48 and Naive Bayes machine learning algorithms, merged with feature selection techniques from the S.U., R.F., I.G., OR, and C.S. to serve as statistical filters. Using the J48 algorithm to predict 18 variables from the G.R. method, the researchers achieved an accuracy of 88% while accelerating factor 2 by a factor of 2. In the end, this was decided to be the best technique for this study.

To accomplish their conclusions, The authors (M.K. Hooshmand, 2020) employed feature selection processes and set theory's quorum and union combination techniques to combine the results of many approaches. Many machine learning algorithms, including R.F., have been compared in terms of their performance while employing optimum feature sets, and the findings have been made available...

(Performance assessment of intrusion detection with Apache Spark and machine learning, 2018; M. Belouch.) The authors performed experimental tests on the Apache Spark environment for large data. They evaluated the effectiveness of frequently used categorization machine learning approaches, including neural network (N.B.), support vector machine (SVM), deep learning (D.T.), and random forest (R.F.). They timed how long it took network intrusion detection systems to detect an intrusion, how long it took to construct a defence, and how long it took to predict an intrusion. Using the UNSW-NB15 data set to test performance, they found that the R.F. technique outperformed the other four algorithms in terms of specificity, accuracy, sensitivity, and execution time.

Researchers in (M. Idhammad 2017) used artificial neural networks to examine a victim-end-based dos-detection approach (ANN). This work detected dos using back-propagation and feed-forward learning techniques. They chose a set of features regarded as more valuable using an unsupervised correlation-based technique. The studies were conducted in three stages: The first step was to gather information about the incoming network traffic, the second was to perform feature reduction for D.O.S. detection, and the third was to divide the network traffic into two classes: benign traffic and malicious attack traffic. Results were comparable to state-of-the-art dos detection methods when tested on both the NSW-KDD and the UNSW-NB15 datasets...

In (Slay N.M. 2015), The authors offered a hybrid strategy to feature reduction based on the C.P. of attribute values followed by an A.R.M. and reported their findings. To save processing time, the dataset was first separated into equal-sized divisions. The CP approach's result was then utilized as input to the A.R.M. strategy to minimize the number of features. The first step was to gather information about the incoming network traffic, the second was to perform feature reduction for D.O.S. detection, and the third was to divide the network traffic into two classes: benign traffic and malicious attack traffic. Results were comparable to state-of-the-art dos detection methods when tested on both the NSW-KDD and the UNSW-NB15 datasets... According to them, the model increased accuracy while decreasing false alarm rates and processing time. The used data sets were NSL-KDD and UNSW-NB15.

Employing the reptree algorithm with protocol subset slicing, the authors of (M. Belouch, A two-stage classifier technique for network intrusion detection using the reptree algorithm, 2017) developed a two-phase classification technique for network intrusion detection. In the first phase, data were segregated for various protocols such as TCP, U.D.P., and others. Feature selectors such as ANN, NB, and D.T. were employed in the second stage to reduce the number of features from 40 to 20 in the third step. They discovered that two-stage classifiers beat single-stage classifiers in terms of the ratio of speed to accuracy.

Apoorva Deshpande and R. Sharma (2018) suggested a model for network intrusion detection based on normalized features, a multilevel ensemble classifier, and a multilevel ensemble classifier. After normalizing the data in the first phase, it was chosen to employ multilevel ensemble classifiers in the second step.

Researchers use various methods to choose which features to focus on (et al., 2020). Combine the methods of recursive feature reduction, variance, chi-square analysis, and the variance threshold. To achieve a final result, the outcomes of separate feature selection strategies were integrated with the help of an intersection method. When compared to the effect of all characteristicsR.F. and Adaboost ensemble algorithms were used to investigate the influence of chosen characteristics on the FAR and accuracy detection rate measures. During testing, it was discovered that the suggested feature selection strategy considerably influenced the performance of classifiers.

Hossein Gharaee and co-workers have presented an anomaly-based intrusion detection system (Hosseinvand, 2016). SVM and G.A. were utilized as machine learning algorithms, with a new feature reduction methodology tossed in for good measure. As a feature selection technique, a genetic algorithm with an extra innovation fitness function is combined with an additional innovation fitness function. The outcome demonstrates that the data dimension has decreased, the accuracy has increased, and the FAR has dropped.

In (Le, Thi-Thu-Huong,2022), explanations of machine learning (ML) model predictions were used in conjunction with big IoT-based IDS datasets to improve attack detection performance. This was carried out to help people comprehend how assaults are identified. The decision tree (DT) and random forest (RF) classifiers used in the ML-based IDS technique are based on the ensemble trees methodology, which doesn't need a lot of processing power to train the model. The trials also made use of the IoTDS20 dataset. In addition, the eXplainable AI (XAI) approach was used to apply the Shapley additive explanations (SHAP) technique to understand and explain the classification choices made by the DT and RF models. This methodology is useful for understanding the ensemble tree approach's ultimate judgement, but it also helps cybersecurity specialists improve and assess the precision of their conclusions based on the explanations of the findings.

It was recommended that the Support Vector Machine (SVM) and the Chaos Game Optimisation (CGO) algorithm be integrated to manage the complexity of big data and diverse security data ensembles (Ponmalar, 2022). In addition to increasing intrusion classification accuracy, the suggested technique also recognises nine different types of assaults in the UNSW-NB15 dataset.

When compared to other techniques, the Ensemble Support Vector Machine (SVM) that is combined with Chaos Game Optimisation (CGO) shows excellent performance in terms of precision, recall, F1-score, accuracy, and ROC curve. The suggested technique outperforms the chi-SVM in terms of accuracy (96.29%), which is an increase of 6.47 percentage points over the chi-SVM (89.12%).

Gain-ratio, chi-squared, and information gain are three of the most effective feature selection methods used in this proposed model, which uses improved weighted majority voting to provide a qualifying result and four of the top classifiers (SVM, LR, NB, and DT). Furthermore, an experimental method built on the ground-breaking Honeypot dataset was created (Krishnaveni, 2022). In each trial, the tools Honeypots, Kyoto, and NSL: KDD were applied. With an accuracy of 98.29%, a false alarm rate of 0.012%, a detection rate of 97.9%, and an area under the curve (AUC) of 0.9921, the recommended intrusion detection strategy based on the Honeypot dataset is a better and more efficient than existing approaches.

3. Ensemble classifiers

An ensemble machine learning model is created when two or more machine learning models work together to make predictions (E.M.L.). Members of the ensemble could be the same or various models, and they could have been trained on the same or separate training data[1, 2]. Ensemble members refer to the models who contribute to the ensemble. Statistical methods such as the mode or mean can be used to aggregate the predictions made by the ensemble members or more advanced algorithms that learn how much to trust each member and under what conditions[2,3] and then combine those forecasts. Articles on some of the most popular and commonly used methods, such as core bagging and boosting approaches [4,], were published in huge volumes when the research of ensemble methods truly took off in the 1990s. [5] When using ensemble methods, the cost and complexity of the calculation are significantly enhanced. This increase is due to the higher level of skill and time necessary to train and maintain many models vs. a single model, resulting in a higher overall cost of ownership. [6,7] As previously stated, there are two major reasons why an ensemble model is preferable to a single model; both of these factors are intertwined, as follows: When comparing the performance of an ensemble of models to a single contributing model, the ensemble produces more accurate predictions and results. An ensemble minimizes the spread or dispersion of forecasts while reducing the number of predictions to improve model performance[8,9]. [10] Rather than using a single predictive model, ensembles of predictive models can be used to improve prediction performance on a predictive modelling task. According to the research[11][12], this is achieved by reducing the variance of the prediction error while increasing the bias of the prediction process (i.e., in the context of the bias-variance trade-off). The improvement in resilience or reliability in a model's average performance is another key and underappreciated consequence of ensemble techniques[13][14]. These are both crucial elements to bear in mind while we work on a machine learning project, and we may choose to emphasize one or both of these traits in our model[15].

4. ENSEMBLE MODEL OF GRADIENT BOOSTING WITH DECISION TREE (EGDT-BOOST)

4.1Preprocessing

In this work, data standardization has been used for Pre-processing, standardization's modification of data. Centring the data involves subtracting the mean from each feature, and scaling it involves dividing (non-constant) values by the standard deviation of the features, as seen in the following example. After data standardization, the mean and standard deviation will be one, and the standard deviation will be zero. Model performance may be dramatically enhanced by standardizing their design and construction. The RBF kernel of SVMs and the l1 and l2 regularizers of L.M.s assume that all features are centred around zero and have variance in the same order as the input features. The variation of a single quality may dominate the objective function if it is several orders of magnitude larger than the variance of the other qualities, preventing the estimator from acquiring knowledge about the other qualities in the normal fashion.

Following a particular notion of standardization, the StandardScaler, the principal scaler in learning, standardizes the data. When the following formula is used for the data, it becomes purely centred: where u represents the mean, s represents the standard deviation, and x is scaled by the fraction (x-u)/s, where x is the original value of x.

With the MinMaxScaler, Scaling each feature to a certain value range within a specified range of values allows you to modify the look of a feature. The feature range option (which has a default value of (0,1)) can be used to specify the range for the feature. Since this scaler is more precise when the distribution is not Gaussian or when the standard deviation is very small, it is used more often under these conditions. It is robust to outliers yet sensitive to them, and thus if your data contains any outliers, you should consider using a different scaler.

In the case of max(x) – min(X), the formula is:

$$(\max(x)–\min(x) = x \text{ scaled}.$$

It is doubtful that using your data's mean and standard deviation to scale will result in good results if your data has many anomalies. In these instances, the RobustScaler can be used to achieve the desired results. Upon removal of the median value, the data is scaled following the quantile range of the data.

4.2 Feature Extraction

By industry standards, approaches for choosing characteristics often fall into two categories. It relies on the conclusion of the feature selector: whether it returns a subset of relevant features or a ranked list of all the relevant characteristics; it also depends on the output of the feature selector (known as feature ranking). [17] In the latter circumstance, it is required to define a threshold to lower the complexity of the problem, which is a difficult question to answer. Based on the interaction between a feature selection algorithm and the inductive learning methodology used to infer a model [18], feature selection techniques are often categorized into three basic approaches. Existing filters include those that rely on generic data characteristics and are independent of the induction process; wrappers that utilize the prediction supplied by a classifier to assess subsets of features; and filters that rely only on the output of the induction algorithm. and embedded techniques that execute F.S. during training are tailored to various learning machines. [19].

This work extracts the most valuable features from a given dataset using the choose best feature selection techniques and the SelectKBest class. The features are chosen using the SelectKBest approach depending on which feature has the greatest score out of k. The "scornful" argument's value can be changed to allow us to perform a classification and regression data analysis. Selecting the best features is a crucial stage in the process of getting ready a large dataset for training. It enables us to eliminate the data's less important components and reduces the suggested intrusion detection system's training time, both of which are advantageous.

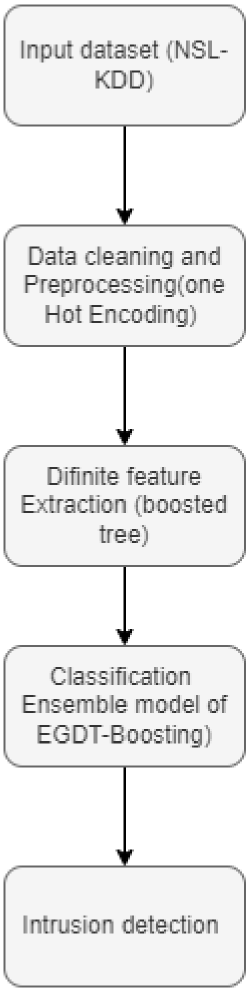

Figure 1: Architecture of EGDT-Boosting

4.3 EGDT-BOOST classifier

Boosting with gradients is an ensemble machine-learning technique extensively used in data science to address classification and regression problems. It's easy to use and works with various data, including small and heterogeneous data. P pooling their efforts efficiently generates a strong learner from a collection of many weak learners. Gradient-boosted decision trees are employed in the proposed enhanced ensemble model of gradient boosting with a decision tree (EGDT-boost). A succession of decision trees is built progressively during training. Each successive tree is built with a lower loss level than the previous ones. The initial parameters dictate the number of trees that are planted. Gradient boosting is implemented in an additive method, which generates a series of approximations greedily iteratively (EGDT-boost). The EGDT-boost algorithm is used to improve this gradient-boosting technique.

The greedy method averages the goal for each category group and applies it to all categories. The issue is that the target value is utilized to build a representation for the group variables, which is subsequently used for prediction, resulting in target leakage. The Holdout technique addresses this problem by splitting the training data set. However, as a result, the training data's effectiveness is considerably diminished. Leaving one out of the sample excludes the target population, but it's ineffective. Online Learning methods, which sequentially send training samples over time, impact ordered target statistics. It inserts false time as a sigma random permutation, a random permutation of the training instances. It will depend only on training examples from the past to prevent target leakage (samples that occurred before that particular sample in the fake time).

Definite features are common in datasets, and there are several ways to deal with them in boosted trees. The suggested model automatically handles definite features in the input data, unlike current gradient boosting algorithms (which require numeric data). One-hot encoding is one of the most used methods for dealing with definite data. However, it gets challenging when a large number of characteristics are involved. To address this, characteristics are classified and determined by goal statistics. When I think of goal statistics, I think of phrases like greedy, hold out, leave one out, and order.

Algorithm for proposed EGDT-Boosting

Step 1. Import the libraries and modules that are required.

Step 2. Import NSL-KDD data

Step 3. Cleanse and Pre-Process data using standardization.

Step 4. Separate the training and testing phases.

Step 5. Create a list of column indices that contain the specific data as part of step 5(During training, the model will be given this list to work from definite Feature Extraction).

Step 6. Write a function using this information that takes a data frame and produces a list containing the indexes of all non-numeric columns.

Step 7. Convert all definite columns to the group data type the proposed model requires.

Step 9. Divide our data into two datasets: one for training and one for testing.

Step 10. Assess the model's performance.

5. Implementation Result and analysis

This phase aims to present the experimental results of the proposed ENSEMBLE MODEL OF GRADIENT BOOSTING WITH DECISION TREE (EGDT-BOOST) strategies in conjunction with five magnificence category methodologies (starting with Dos and progressing through Probe and r21) that were used to detect network intrusions using the NSL-KDD Cup intrusion detection datasets. To assess the viability of our intrusion detection model. This work compared ten efficient classification models from the domains of Decision Tree(DT), Random Forest (RF), KN.N., Support Vector Machine (SVM), Linear Regression (LR), Stochastic Gradient Descent (SGD), Adaboost, XGboost, Voting, and Lightgbm. In this proposed work, the length of the Train dataset has been considered as 5290866 and the length of the Test dataset 946848. Figure 2 describes the Data instance after Pre-processing of Standardisation.

Attack Class	Data Instance
U2r	5000
Probe	7000
R21	10500
DOS	50000
Benign	60900

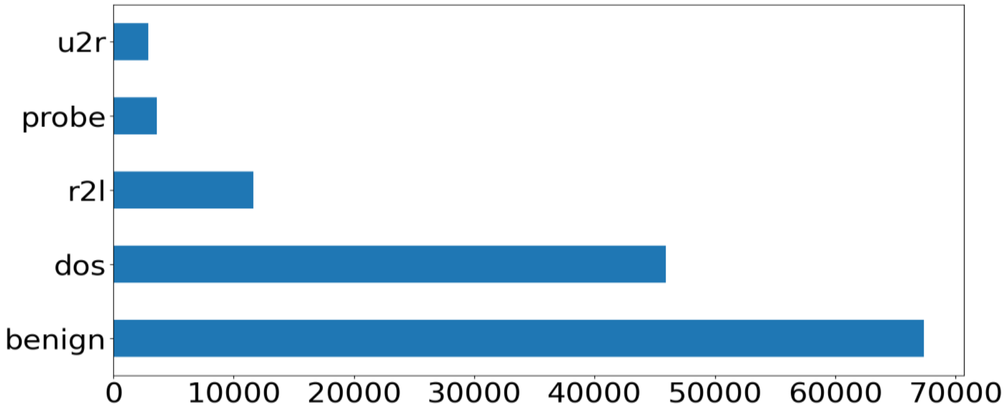

Figure 2 Data instance After Pre-processing of Standardisation

Table 1 describes the total number of samples that belong to each of the five classes of the training dataset attack category. Figure 3 describes the classification report of the proposed EGDT-BOOST with the confusion matrix.

Table 1 Total Number Of Instance And The Class

Attacks	Samples
benign	067343
dos	045927
probe	011656
r2l	0995
U2r	052

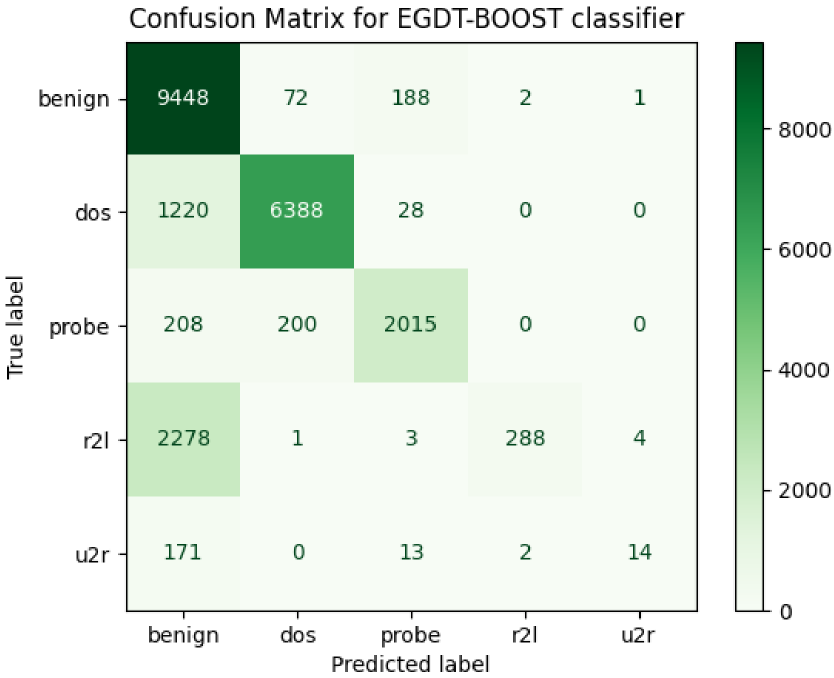

Figure 3 Confusion matrix of Proposed EGDT-BOOST classifier

Table 2 represents a distinct determination of the metrics for each class in this model as if each class were to have its classifier. Here are the Macro averaging per-class stats and the weighted average for convenience.

Table 2 Performance analysis of EGDT-BOOST classifier

Class	Precision	Recall	F-measure
benign	71	97	82
Dos	96	84	89
Probe	90	83	86
R21	99	11	20
U2r	74	07	13

Instead of having many F1 ratings for each class, it would be ideal to average them all together to get a single score that represents overall performance. Macro averaging is one of the many methods, possibly the most straightforward.

Table 3 Performance analysis of weighted avg and macro avg

Performance metrics	Weighted avg(%)	Macro avg(%)
Precision	85	86
Recall	81	56
f-measure	58	77

Table 3 describes the Performance analysis of the weighted avg and macro avg; the arithmetic mean of all the F1 scores for each class in Intrusion Detection is used to compute the macro-averaged F1 score for the classification. In this work, the macro average of Intrusion Detection uptrained 86,56 and 77 for precision, recall, and f-measure. This function treats all classes identically, regardless of their support settings.

The weighted-average F1 score is calculated by averaging all per-class F1 scores while accounting for the amount of help each class receives. Refer to the number of times the class has occurred in the data collection while discussing support. The term "weight" refers to the proportion of support given to each class as a percentage of overall support given to all classes. With weighted averaging, the output average would have considered each class's contribution, which would have been weighted by the number of cases in the dataset for that class. Table 4 describes the performance analysis of the proposed work with various other existing algorithms.

Table 4 Performance Analysis Of Various Classifiers

Methods	Attack class	Precision	Recall	F1-score
Decision Tree	benign	67	96	81
	Dos	96	83	90
	Probe	87	63	71
	R21	98	10	18
	U2r	67	01	03
Random forest	benign	66	97	79
	Dos	96	82	89
	Probe	87	61	72
	R21	96	04	0
	U2r	50	01	03
KNN	benign	67	97	79

	Dos	96	78	86
	Probe	79	68	73
	R21	97	07	12
	U2r	75	03	06
SVM	benign	66	94	77
	Dos	92	80	86
	Probe	90	65	75
	R21	96	10	19
	U2r	83	03	05
LR	benign	66	98	66
	Dos	93	80	79
	Probe	95	92	98
	R21	98	09	22
	U2r	86	02	21
SGD	benign	96	83	90
	Dos	87	63	71
	Probe	98	10	18
	R21	1.00	00	00
	U2r	00	00	00
Adaboost	benign	66	95	81
	Dos	96	82	89
	Probe	87	61	72
	R21	00	00	00
	U2r	00	00	00
XGboost	benign	61	92	79
	Dos	98	62	89
	Probe	87	61	72
	R21	96	05	08
	U2r	50	11	03
Voting	benign	67	97	79
	Dos	96	78	86
	Probe	86	74	80
	R21	96	04	08
	U2r	25	01	01
Lightgbm	benign	67	97	79
	Dos	96	80	87
	Probe	83	65	73
	R21	99	09	17
	U2r	76	08	14
Proposed model	**benign**	**71**	**97**	**82**
	Dos	**96**	**84**	**89**
	Probe	**90**	**83**	**86**
	R21	**99**	**11**	**20**
	U2r	**74**	**07**	**13**

Table 5Performance Analysis in Accuracy of Various Classifiers

Methods	**Accuracy**
DT	72.1111

RF	76.7699
KNN	76.3174
SVM	75.7452
LR	75.1419
SDG	74.6627
Adaboost	67.0999
XGboost	77.1114
Voting	76.8675
Lightgbm	77.8675
Proposed	**80.5225**

In this work, the weighted average of Intrusion Detection uptrained 0.85,0.81 and 0.58 for precision, recall, and f-measure. Both analyses show that the proposed model achieves high efficiency also in single classes analysis. Table 5 describes the performance analysis of the accuracy of the proposed work with various other existing algorithms. Figure 5 shows that the proposed work archives a high accuracy of 80.5225% over other work.

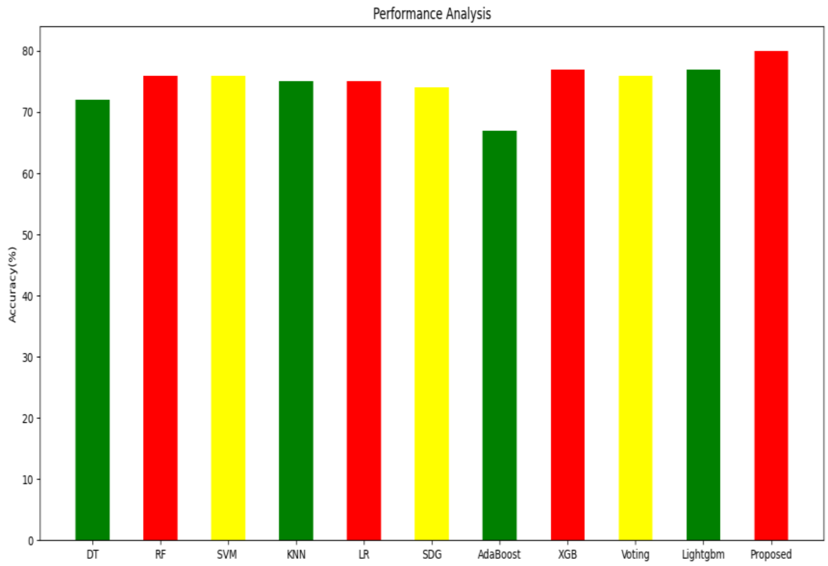

Figure 4 Performance Analysis in Accuracy of Various Classifiers

6. Conclusion

Network security with attack detection using the Ensemble Classifier of EGDT-BOOST has been proposed as an alternate structure for network security. In today's world, most routed communications are not just used for beneficial purposes. Still, cybercriminals are also making use of routing systems to conduct port scanning, data exfiltration, and other types of online fraud. The security of networks is very vulnerable to these types of cybercrime activities. Ensemble Classifier is a machine learning approach for prediction and classification that has shown to be incredibly successful in recent years. A prominent technique for improving learning models' efficiency is combining numerous models. This work has tested this strategy, and the results are provided as an Ensemble Classifier-based classification model. It is possible to develop the Ensemble Classifier model by combining two different learning models. This work compared ten efficient classification models from the domains of Decision Tree(DT), Random Forest (RF), KN.N., Support Vector Machine (SVM), Linear Regression (LR), Stochastic Gradient Descent (SGD), Adaboost, XGboost, Voting, and Lightgbm. The models were selected from the following domains: The outcome demonstrates that the proposed EGDT-BOOST achieved an efficient accuracy of 80 per cent in attack detection when tested.

7. References

[1] Bilge, Leyla, and Tudor Dumitraş. "Before we knew it: an empirical study of zero-day attacks in the real world." In Proceedings of the 2012 A.C.M.

conference on Computer and communications security, pp. 833-844. 2012.

[2] Holm, Hannes. "Signature-based intrusion detection for zero-day attacks:(not) a closed chapter?." In 2014 47th Hawaii international conference on system sciences, pp. 4895-4904. IEEE, 2014.

[3] Lamba, Anil, Satinderjeet Singh, and Singh Balvinder. "Mitigating zero-day attacks in IoT using a strategic framework." International Journal for Technological Research in Engineering 4, no. 1 (2016).

[4] Zhang, Mengyuan, Lingyu Wang, SushilJajodia, Anoop Singhal, and Massimiliano Albanese. "Network diversity: a security metric for evaluating the resilience of networks against zero-day attacks." IEEE Transactions on Information Forensics and Security 11, no. 5 (2016): 1071-1086.

[5] Portokalidis, Georgios, Asia Slowinska, and Herbert Bos. "Argos: an emulator for fingerprinting zero-day attacks for advertised honeypots with automatic signature generation." ACM SIGOPS Operating Systems Review 40, no. 4 (2006): 15-27.

[6] Boetto, Erik, Maria Pia Fantini, Aldo Gangemi, DavideGolinelli, Manfredi Greco, Andrea Giovanni Nuzzolese, Valentina Presutti, and Flavia Rallo. "Using altmetrics for detecting impactful research in quasi-zero-day time-windows: the case of COVID-19." Scientometrics (2021): 1-27.

[7] Sohi, Soroush M., Jean-Pierre Seifert, and FatemehGanji. "RNNIDS: Enhancing network intrusion detection systems through deep learning." Computers & Security 102 (2021): 102151.

[8] Kamati, Toivo Herman, Dharm Singh Jat, and SaurabhChamotra. "Design and Development of System for Post-infection Attack Behavioral Analysis." In Proceedings of Fifth International Congress on Information and Communication Technology, pp. 554-565. Springer, Singapore, 2021.

[9] Garcia, Norberto, Tomas Alcaniz, Aurora González-Vidal, Jorge Bernal Bernabe, Diego Rivera, and Antonio Skarmeta. "Distributed real-time slowdowns attacks detection over encrypted traffic using artificial intelligence." Journal of Network and Computer Applications 173 (2021): 102871.

[10] Bokka, Raveendranadh, and TamilselvanSadasivam. "Deep Learning Model for Detection of Attacks in the Internet of Things Based Smart Home Environment." In Proceedings of International Conference on Recent Trends in Machine Learning, IoT, Smart Cities, and Applications, pp. 725-735. Springer, Singapore, 2021.

[11] Singh, Geeta, and NeeluKhare. "A survey of intrusion detection from the perspective of intrusion datasets and machine learning techniques." International Journal of Computers and Applications (2021): 1-11.

[12] Aksoy, Muhammet, OrhanOzdemir, GuneyGuner, BarisBaspinar, and EmreKoyuncu. "Flight Trajectory Pattern Generalization and Abnormal Flight Detection with Generative Adversarial Network." In AIAA Scitech 2021 Forum, p. 0775. 2021.

[13] AdtiyaNurCahyo, RisanuriHidayat, and Dani Adhipta, "Performance Comparison of Intrusion Detection System based Anomaly Detection using Artificial Neural Network and Support Vector Machine", Advances of Science and technology for Society,978-0-7354-1413-6,doi-10.10631/1.4958506,2016.

[14] Salima Omar, AsriNgadi, and Hamid H. Jebur , "Machine Learning Techniques for Anomaly detection: An Overview", Internation Journal of Computer Application,ISSN: 0975-8887, Volume 79-No.2 October, 2013.

[15] Sergay Andropov, Alexei Guirik, Mikhail Budko and Marina Budko, "Network Anomaly Detection using Artificial Neural Network", Open Innovation Association(FRUCT) 20th Conference,2017, ISSN NO:2305-7254,IEEE 2017.

[16] Mrutyunjaya Panda and Manas Ranjan Patra, "Network Intrusion Detection Using Naïve Bayes", International Journal of Computer science and Network Security, Vol. 7, No. 12, December 2007.

[17] Manjiri V. Kotpalliwar and RakhiWajgi, "Classification of Attacks Using Support Vector Machine (SVM) on KDDCUP'99 IDS Database", Fifth International Conference on Communication Systems and Network Technologies, 978-1-4799-1797-6, pp: 987-990, April

[18] Khoei, TalaTalaei, GhilasAissou, When Chen Hu, and Naima Kaabouch. "Ensemble learning methods for anomaly intrusion detection system in smart grid." In 2021 IEEE International Conference on Electro Information Technology (E.I.T.), pp. 129-135. IEEE, 2021.

[19] Tama, BayuAdhi, and Sunghoon Lim. "Ensemble learning for intrusion detection systems: A systematic mapping study and cross-benchmark evaluation." Computer Science Review 39 (2021): 100357.

[20] Deshpande, A., & Sharma, R. (2018). Multilevel Ensemble Classifier using Normalized Feature based Intrusion Detection System. International Journal of Advanced Trends in Computer Science and Engineering, 7(5), 72-76. https://doi.org/10.30534/ijatcse/2018/02752015

[21] Divekar, A., Parekh, M., Savla, V., Mishra, R., &Shirole, M. (2018). Benchmarking datasets for anomaly-based network intrusion detection: KDD CUP 99 alternatives. In IEEE 3rd International Conference on Computing, Communication and Security (ICCCS), 1-8.

[22] Hooshmand, M.K. (2020). Using Ensemble Learning Approach To Identify Rare Cyber-Attacks In Network Traffic Data. In International Conference on Advanced Computer Science and Information Systems (ICACSIS), 141-146.

[23] Mogal, D.G., Ghungrad, S.R., &Bhusare, B.B. (2017). NIDS using machine learning classifiers on UNSW-NB15 and KDDCUP99 datasets. International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), 6(4), 533-537.

[24] Ullah, F., & Babar, M.A. (2018). Architectural tactics for big data cybersecurity analytics systems: a review. Journal of Systems and Software, 151, 81-118.

[25] Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. Ieee Access, 5, 21954-21961.

[26] Dua, S., & Du, X. (2016). Data mining and machine learning in cybersecurity. C.R.C. press. Buczak, A.L., &Guven, E. (2015). A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications surveys & tutorials, 18(2), 1153-1176.

[27] Anwer, H.M., Farouk, M., & Abdel-Hamid, A. (2018). A framework for efficient network anomaly intrusion detection with features selection. In 9th International Conference on Information and Communication Systems (ICICS), 157-162.

[28] Gharaee, H., &Hosseinvand, H. (2016). A new feature selection I.D.S. based on genetic algorithm and SVM. In 8th International Symposium on Telecommunications (I.S.T.), 139-144.

[29] Belouch, M., El Hadaj, S., &Idhammad, M. (2017). A two-stage classifier approach using reptree algorithm for network intrusion detection. International Journal of Advanced Computer Science and Applications, 8(6), 389-394.

[30] Belouch, M., El Hadaj, S., &Idhammad, M. (2018). Performance evaluation of intrusion detection based on machine learning using Apache Spark. Procedia Computer Science, 127, 1-6.

[31] Idhammad, M., Afdel, K., &Belouch, M. (2017). Dos detection method based on artificial neural networks. International Journal of Advanced Computer Science and Applications, 8(4), 465-471.

[32] Hooshmand, M.K., & Gad, I. (2020). Feature selection approach using ensemble learning for network anomaly detection. CAAI Transactions on Intelligence Technology, 5(4), 283-293.

[33] Sheikhan, M., Jadidi, Z., &Farrokhi, A. (2012). Intrusion detection using reduced-size R.N.N. based on feature grouping. Neural Computing and Applications, 21(6), 1185-1190.

[34] Alom, M.Z., Bontupalli, V., & Taha, T.M. (2015). Intrusion detection using deep belief networks. In National Aerospace and Electronics Conference (NAECON), 339-344.

[35] Moustafa, N. (2017). Designing an online and reliable statistical anomaly detection framework for dealing with large high-speed network traffic (Doctoral dissertation, University of New South Wales, Canberra, Australia).

[36] Moustafa, N., & Slay, J. (2015). A hybird feature selection for network intrusion detection systems: central points and association rules. In Australian Information Warfare Conference, 5-13.

[37] Moustafa, N., & Slay, J. (2016). The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. Information Security Journal: A Global Perspective, 25(1-3), 18-31.

[38] Dahiya, P., & Srivastava, D.K. (2018). Network intrusion detection in big dataset using spark. Procedia computer science, 132, 253-262.

[39] Heshmati, B., Hashempour, L., Saberi, M.K., Fattahi, A., &Sahebi, S. (2020). Global research trends of public libraries from 1968 to 2017: A

bibliometric and visualization analysis. Webology, 17(1), 140-157.

[40] Le, Thi-Thu-Huong, et al. "Classification and Explanation for Intrusion Detection System Based on Ensemble Trees and SHAP Method." Sensors 22.3 (2022): 1154.

[41] Ponmalar, A., and V. Dhanakoti. "An intrusion detection approach using ensemble Support Vector Machine based Chaos Game Optimization algorithm in big data platform." Applied Soft Computing 116 (2022): 108295.

[42] Krishnaveni, Sivamohan, et al. "Network intrusion detection based on ensemble classification and feature selection method for cloud computing." Concurrency and Computation: Practice and Experience 34.11 (2022): e6838.

Authors

Ms. Stency V S received BTech degree in Computer Science and Engineering from University of Calicut, Kerala, India in 2008 and ME degree in Computer Science and Engineering from Anna University, Chennai, Tamil Nadu, India in 2014. She is currently pursuing PhD at the Department of Computer Science and Engineering, Karpagam Academy of Higher Education, Coimbatore, India in Intrusion Detection System using Ensemble Deep Learning Techniques.

E-mail: stenz.denz@gmail.com

[1]. **Dr. N. Mohana Sundaram** holds B.E degree in Electrical and Electronics Engg. from Madras University in 1979. Holds M.E degree in Computer Science and Engg from Bharathiar University in 1991 and Ph.D. in Computer Science and Engineering from Karpagam University. He has a total Teaching Experience of about 43 years in India and Abroad. Primary Research areas are Neural Networks, Data Mining and Machine Learning.

E-mail: itismemohan@gmail.com

Dr. R. Santhosh, is the Professor and Head of Computer Science and Engineering Department at Karpagam Academy of Higher Education, Coimbatore, Tamil Nadu. He holds BTech degree in IT in 2006 and M.E degree in Software Engineering in 2009 from Anna University, Chennai, Tamil Nadu, India. He also holds MBA from Alagappa University in 2011 and Ph.D. in Computer Science and Engineering from Karpagam University in 2016. He has a Teaching Experience of about 13 years.

E-mail: santhoshrd@gmail.com