JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 3, 2022, p. 355-366
https://publishoa.com
ISSN: 1309-3452

Homogenous Ensemble Learning for Denial of Service Attack Detection

Nazanin N. Abdulla [1]* ; [2]**Rajaa K. Hasoun**

[1]*Informatics Institute for Postgraduate Studies Iraqi Commission for Computers & Informatics*

[2]*University of Information Technology and Communication*

* Corresponding author :nazanin.shafi95@gmail.com

Received 2022 April 02**; Revised** 2022 May 20**; Accepted** 2022 June 18**.**

Abstract: Network hacking has become more resource-intensive in recent years, particularly in firms and organizations that rely on the internet, such as Amazon, because the hacker's goal is to prohibit the user from using the network's resources whether the target is offensive or pecuniary. As a result, several approaches for detecting intrusion and network penetration have arisen, utilizing various techniques such as artificial intelligence processes, machine learning, and deep learning to discriminate between authorized and illegitimate users. This paper discusses the diverse methods of ensemble learning as a part of machine-learning techniques and Ensemble-learning techniques used in different datasets to identify denial of service attacks. Demonstrates a variety of machine-learning algorithms, including Random Forest, Bagging, and Boosting, these are used to discover various sorts of attacks in the NSL_KDD dataset Bagging algorithms had the maximum detection accuracy of 99 %, while AdaBoost methods had the lowest accuracy of 93 %. As a result, the same approaches were used on the same dataset, but exclusively to detect DoS attacks. Bagging, RF, and eXtreme gradient boost (XGB) algorithms had the maximum detection accuracy of 99 %, while AdaBoost methods had the lowest accuracy of 37 %.

Keywords: Machine Learning Ensemble, NSL-KDD, Denial of Services Attack, Performance Evaluation.

1. Introduction

Many attackers organize the transmission of a large amount of meaningless data to try to overload the target's computing resources or the close network links in a volumetric DOS attack. DOS is the most dangerous attack that causes a problem with the network traffic [3]. Since the Computer Incident Warning Service announced the first attack event in 1999, DoS attacks have been one of the most persistent network security risks. Despite the reality that many defence mechanisms have been suggested in industry and academia, DOS attacks remain a major threat that is increasing year after year. DoS attacks are an essential security issue in any network architecture. The full network can be damaged by overlapping a bandwidth that DOS attacks do it. DOS exploits any fault that occurs in the table of software defines networks (SDN) to crush the controller by using the packet in messages. The enemy can use a variety of methods to fake packet fields and send huge traffic tables, as well as regular traffic, to SDN architecture, leading bandwidth, and other resources to be depleted. The controller is flooded with faked flows without an efficient security system, forcing the controller to fail by establishing new flow rules and actions [2]. This paper aims to use Ensemble learning that can be used to identify attacks, especially the attacks that prevent access to network traffic like denial of service attacks by using homogenous ensemble learning bagging and boosting methods.

2. Related work

S. Aljawarneh et al. in 2019 [4] presented an upgraded J48 algorithm that improves the J48 algorithm and was used to improve the accuracy and performance detection of the new IDS method. This improved J48 method is thought to aid in the detection of potential assaults that could jeopardize network confidentiality. The researchers used a variety of datasets and combined several methodologies such as the Naïve Bayes, Random Tree, J48, and NB-Tree to achieve their goal. Throughout all of the trials, an NSL_KDD incursion dataset was used.

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 3, 2022, p. 355-366
https://publishoa.com
ISSN: 1309-3452

G. Kushwah et al. in 2020 [6] proposed a method for detecting DDOS traffic that uses net flow feature selection and machine learning. They utilized Random Forest to create a detector, which they tested on a network trace in a research facility that included both benign traffic and simulated DDoS traffic generated by typical DDoS tools of various types. Their system has a false-positive rate of less than 1% and a 99 percent accuracy rate, according to test data.

C. Chukwuemeka et al. in 2020 [7] proposed a deep learning and Long Short Term Memory-based DDoS detection system (LSTM). The proposed model was tested on the UNSW-NB15 and NSL_KDD intrusion datasets, with Singular Value Decomposition extracting twenty-three (23) and twenty (20) attack features from UNSW-NB15 and NSL-KDD, respectively, from UNSW-NB15 and NSL-KDD, respectively (SVD).

R. Dong et al. in 2020 [8] analysed the problem is that the intrusion detection mechanism in wireless sensor networks (WSN) is exceedingly sophisticated, resulting in high computing complexity and low intrusion detection performance. Based on the information gain ratio and bagging approach, the ensemble learning algorithm was utilized to construct an intrusion detection model for WSN. The computational complexity of the intrusion detection approach is decreased by lowering the dimension of the acquired WSN traffic data using a feature selection method based on the information gain ratio.

N. Singh et al. in 2020 [9] employed three distinct datasets to find the accuracy of five of the most popular machine learning algorithms: Decision Tree, K-Nearest Neighbour, Naïve Bayes, Random Forest, and Support Vector Machine on KDDCup99 dataset, NSL_KDD dataset, and WSN-DS dataset. The goal of this study is to see if a new dataset called WSN-DS provides better accuracy than existing datasets using the same machine learning methods. The results show that the WSN-DS dataset surpasses the NSL_KDD dataset and the KDD-Cup99 dataset, which have an accuracy of 99.46% and 99.07%, respectively.

3. NSL_KDD Dataset

The NSL_KDD dataset, which is an improvement to the KDD'99 dataset, the NSL_KDD data set was proposed as a solution to some of the KDDCUP'99 data set's fundamental issues. The most extensively used data set for anomaly detection is KDDCUP'99[10]. Hundreds of thousands of connection records are included in the collection, each of which defines a normal or abnormal status as shown in (Table 1) the features in the dataset.

Table.1 NSL_KDD features

No.	Feature	No.	Feature
1	Duration	22	Is-guest-login
2	Protocol type	23	Count
3	Service	24	Srv-count
4	Flag	25	Serror-rate
5	Src-bytes	26	Srv-serror-rate
6	Dst-bytes	27	Rerror-rate
7	Land	28	Srv-rerror-rate
8	Wrong-fragment	29	Same-srv-rate
9	Urgent	30	Diff-ser-rate
10	Hot	31	Srv-diff-host-rate
11	Num-failed-logins	32	Dst-host-count
12	Logged-in	33	Dst-host-srv-count
13	Num-compromised	34	Dst-host-same-srv-rate
14	Root-shell	35	Dst-host-diff
15	Su-attempted	36	Dst-host-same-srv-port-rate
16	Num-root	37	Dst-host- srv-diff-

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 3, 2022, p. 355-366
https://publishoa.com
ISSN: 1309-3452

			host-rate
17	Num-file-creations	38	Dst-host-serror-rate
18	Num-shells	39	Dst-host-srv-serror-rate
19	Num-access-files	40	Dst-host-serror-rate
20	Num-outbound-cmds-files	41	Dst-host-srv-serror-rate
21	Is-host-login	42	Class label

Feature encoding and data normalization are used to pre-process network data. The feature conversion process required converting non-numeric feature values to numeric values, and the network feature normalization process involved scaling network feature values into a comparable value range using the min-max normalization approach. The dataset includes 41 features as shown in (Table 1), and 125973 records. It is containing 5 classes that are 1 normal and 4 type of attacks (Denial of Service (DoS), Probe, User to Root (U2R), Remote to Local (R2L)), the DoS only has 7458 records [7].

Denial of Service Attack (DoS) is an attack in which the attacker makes some computer or memory resource is too busy or full to answer genuine requests, and the machine refuses legitimate users access. A probing attack is an attempt to obtain knowledge about a network of computers in order to obfuscate security mechanisms. The User to Root Attack (U2R) is a type of hack in which the attacker gains access to a system's normal user account (perhaps through password sniffing, a dictionary attack, or social engineering) and then exploits a vulnerability to get root access. An attacker who has the capacity to send packets via a network but does not have access to that system commits a Remote to Local Attack (R2L) [10].

4. The Proposed System

This study provides a classification system based on feature selection for detecting DoS assaults as shown in Fig. 1 how used ensemble algorithms for the detection of Attacks in the NSL_KDD dataset. The suggested framework in (Fig. 1) is partitioned into five stages:

- Load Dataset
- Divided dataset into two-part train and test
- Pre-processing train and test part
- Apply algorithms on data set to detect all attacks type
- Evaluation result

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 3, 2022, p. 355-366
https://publishoa.com
ISSN: 1309-3452

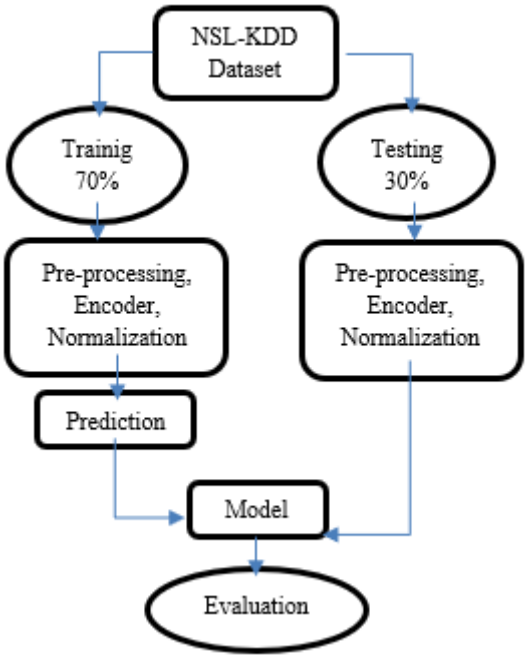


Figure1. The architecture of the proposed system

4.1. Dataset loading stage

Load Dataset first step in the proposed system is to load the NSL_KDD dataset.

4.2. Dataset partition stage

The partition dataset divided the dataset into two-part 70% for training and 30% for testing before pre-processing and applying algorithms.

4.3. Pre-processing stage

Pre-processing the proposed framework's third stage, pre-processing, deals with data encoding and min-max normalization, By supplying uniform and smooth data, this stage makes it easier for classification algorithms to produce effective results in the shortest amount of time. The characters' values are replaced with numeric values throughout the encoding process. The normalizing procedure, on the other hand, handles noisy inputs and keeps feature values within a set range (-1 and 1).

4.4. Detection stage

Techniques of detection in the implementation of intrusion detection systems, and machine learning techniques are currently being used extensively. A machine learning algorithm has the advantage of being able to extract useful information from a dataset. By mixing different models, ensemble learning increases machine learning performance. In terms of prediction accuracy, this technique outperforms a single model. Ensemble learning can be divided into two categories: When a multi-classifier system is made up of multiple types of learners, it is referred to as a homogeneous ensemble like bagging and boosting or a heterogeneous ensemble such as stacking, this paper explained how to probe, U2R, R2L and DoS attacks was detected by using a homogenous type of ensemble learning algorithms. First step applying six ensemble learning (RF, Bagging, AdaBoost, Gradient Boost, Extreme GB, Light GB, and Cat Boost on a dataset to detect the four types of attacks and evaluate their performance. After that apply the same techniques to the dataset but to detect only DoS attacks and measure the performance of the detection process.

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 3, 2022, p. 355-366
https://publishoa.com
ISSN: 1309-3452

4.4.1. Random forest classifier

The random forest and random subspace approaches are the most extensively utilized ensemble methods. Because of their simplicity and ability to predict outcomes, Random forest employs a large number of unpruned, self-contained decision trees[11]. The RF Algorithm is defined by Eq. (1) as follow [12]:

$$p(v|g) = \frac{1}{S} \sum_{s=1}^{S} p_s(v|g) \qquad (1)$$

Where P (v| g) represents the approximate density of labels of the class

Bagging classifier

Bagging, also known as bootstrap aggregation is a technique for improving the accuracy and consistency of a machine learning system. It can be used for regression as well as classification. Bagging also eliminates variance. The Bagging Algorithm is defined by Eq. (2), if there are n data instances represented by [13]:

D = {o1, o2 ... on}

(2) Then, at random, produce a sample of the same size n and with the same probability (1/n for each observation).

4.4.2. Boosting classifier

It is a sequential procedure in which each successive model seeks to rectify the prior model's mistakes. The models that follow are reliant on the prior model. Boosting works in the steps below.

1. From the original dataset, a subset is produced.
2. All data points are given equal weights at the start.
3. This subset is used to build a foundation model.
4. This model is applied to the entire dataset to create predictions.

- **Gradient Boost classifier**

One of the most reliable learning methods is the Boosting algorithm. The GBT method has been used by several researchers in classification applications. When compared to many other classifiers, one of the main benefits of the GBT approach is that it provides a more precise classification. The GB Algorithm is defined by Eq. (3) [14]:

$$\sum_{i=1}^{N} \log(1 + \exp(-2y_i f(x_i)) \qquad (3)$$

Where the number of instances is denoted by N, the label of instance i is y_i, features of instance i is represented by x_i, and $f(x_i)$ is the model's predicted label, d depth of decision trees, K is the number of iterations, The fraction of data η used at each iterative phase, with α learning rate values ranging from 0 to 1. If α is supplied, the GBT is referred to as the Gradient Boosting Machine (GBM). In practice, the more repetitions a prediction tree has, the more accurate it becomes.

- **AdaBoost classifier**

Freund and Schapire presented "adaptive boosting," or "AdaBoost," in 1997, they developed a broader version of the original boosting approach. AdaBoost generates a collection of hypotheses, then utilizes weighted majority voting to combine decisions among the classes specified by the hypotheses. A weak classifier is taught to produce hypotheses by picking instances from a continually refreshed distribution of the training data [13]. The Ada Boost Algorithm is defined by Eq. (4) [15]:

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 3, 2022, p. 355-366
https://publishoa.com
ISSN: 1309-3452

$$z_t = \sum_{i=1}^{m} D_t(i)\exp(-\alpha_t y_i h_t(x_i)) \qquad (4)$$

Where D_t is the weight distribution over all training samples; αt is a parameter that is used to update D_t (i); $y_i \in \{-1, +1\}$ (+1 represents merge while −1 represents split); h_t is the weak hypothesis.

- **Extreme Gradient Boost classifier**

It's a supervised machine learning approach that's a better version of the gradient boosting algorithm. It is based on the ensemble approach and employs the ensemble methodology. By integrating the predictions of weak learners, the XGBoost algorithm creates a powerful learning model. The XGBoost classifier, in addition to its speed and great performance, addresses the overfitting problem and makes optimal use of computational resources. These benefits derive from the ability to mix regularization and predictive words in the training phase, as well as the ability to execute many tasks at the same time. The XGB Algorithm is defined by Eq. (5) [16]:

$$f_i^{(t)} = \sum_{n-1}^{t} f_n(x_i) = f_i^{(t-1)} + f_t(x_i) \qquad (5)$$

Where $f_t(x_i)$ is the prediction at step t, $f_i^{(t)}$ and $f_i^{(t-1)}$ are the Predictions at step t and (t-1), respectively and x_i is the input value. To avoid the problem of overfitting while keeping the model's computational performance

- **Light Gradient Boost classifier**

Microsoft developed Light GBM, an open-source GBDT algorithm. To maximize parallel learning, the parallel voting DT technique uses a histogram-based strategy to speed up the training process, decrease memory utilization, and integrate advanced network connections. Light GBM also grows trees leaf by leaf, dividing the leaf with the greatest variance gain. The LGB Algorithm is defined by Eq. (6) [17]:

$$V_j^*(d) = \frac{1}{n}\left(\frac{\left(\sum_{x_i \in A_l} g_i + \frac{1-a}{b}\sum_{x_i \in B_l} g_i\right)^2}{n_r^j(d)} + \frac{\left(\sum_{x_i \in A_r} g_i + \frac{1-a}{b}\sum_{x_i \in B_r} g_i\right)^2}{n_r^j(d)}\right) \qquad (4)$$

Where

$$A_l = \{x_i \in A : x_{ij} \le d\}. A_r = \{x_i \in A : x_{ij} > d\}$$

$$B_l = \{x_i \in B : x_{ij} \le d\}. B_r = \{x_i \in B : x_{ij} > d\}$$

The coefficient 1-a /b is used as the data point at which the split is computed to find the best gain invariance, and is the data point at which the sum of the gradients across B is normalized to the size of A. Because of its gain of variance technique, which integrates weak learners in the algorithms and tree-growing methods, Light GBM outperforms traditional GBDT models in terms of classification and prediction [17].

- **CatBoost classifier**

It is a categorical-featured unbiased gradient boosting technique. It has categorical properties as well as a revolutionary order-boosting approach that does not predict shift. It offers a variety of categorized features and solutions. Its approach has been optimized and is now used in tree splitting rather than pre-processing. Because the characteristics have a limited number of classes, the classifier converts categorical features to numeric features with a large number of occurrences using one-hot encoding. The classes for composite features are switched with the average target. The Cat Boost Algorithm is defined by Eq. (7) [18].

$$x_{\sigma i.k} = \frac{\sum_{j=1}^{i-1}[x_{\sigma i.k} = x_{\sigma j.k}]y_{\sigma j} + a*p}{\sum_{j=1}^{i-1}[x_{\sigma i.k} = x_{\sigma j.k}]y_{\sigma j} + a} \qquad (7)$$

Where xσi, k = xσj, k will use value 1 when the circumstance is fulfilled; The prior value is denoted by p, while the weights of the prior value are denoted by a. To complete the regression task and compute the prior probability, the average

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 3, 2022, p. 355-366
https://publishoa.com
ISSN: 1309-3452

of the entire dataset, P, is employed.

5. Performance evaluation stage

Some metrics were used to evaluate the performance such as Accuracy, Precision, Recall, and f1-score. After that applying the same algorithms but only to detect DoS attack in the dataset, then evaluated performance. The performance of the recommended classifiers model may be measured using a variety of criteria. The following are the metrics [19]:

5.1. Accuracy

It is calculated from TP and TN and represents how well the model predicts the classes. It's calculated as Eq. (8):

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \qquad (8)$$

5.2. Precision

The percentage of all samples expected to be from class I that are really from class I is calculated as Eq. (9):

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \qquad (9)$$

When the amount of data by class is unbalanced, accuracy alone isn't always sufficient to assess the model's effectiveness. If the model predicts everything as class 0, and there are 99 cases of class 0 and 1 example of class 1, the accuracy is 99%. but when precision is taken into consideration, the model performs badly. The precision of class 0 will be zero in this case.

5.3. Recall

Sensitivity is another name for it. The proportion of all samples predicted to be class I that is really class I This is how it's defined in Eq. (10):

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \qquad (10)$$

As a result, the prior example's class 0 will also have zero recall. The goal of our model is to maximize both precision and recall.

5.4. F-score

It's a mix of recollection and accuracy. The harmonic mean is what it's called. This is how it is defined in Eq. (11):

$$\text{F}_1 = 2 * \frac{\text{precision}*\text{recall}}{\text{precision+recall}} \qquad (11)$$

Where,
TP = true positives: the number of positive examples anticipated that are actually positive.
FP = false positives: the number of examples predicted positively but turned out to be negative.
TN = true negatives: the number of expected negative examples that are truly negative.
FN = false negatives: the number of examples that were expected to be negative but turned out to be positive.

6. Results and discussion

We present the experimental findings of the suggested approach in terms of classification accuracy, model construction time, and false positives in this section. We also compare it to various machine learning approaches that are currently available. For evaluation purposes, the evaluation of a classifier on a test dataset and Stratified Cross-Validation is seen as more relevant. The training and testing datasets provided by NSL KDD are used to assess the classification accuracy.

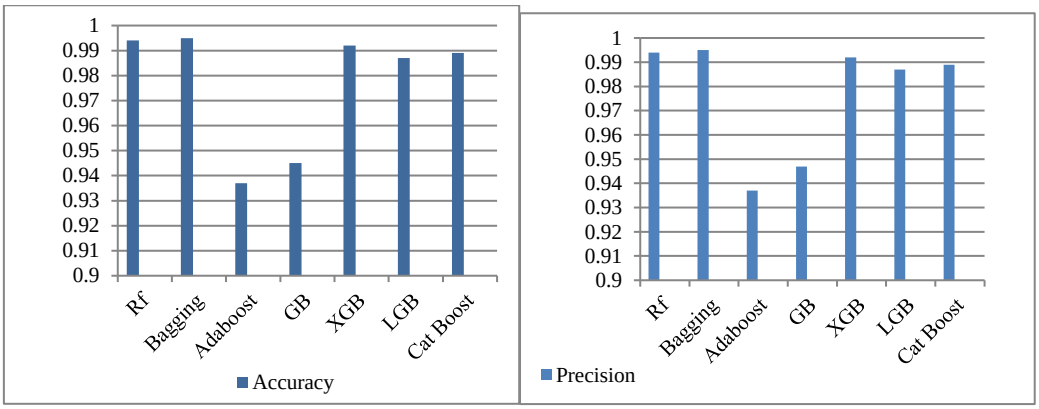
JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 3, 2022, p. 355-366
https://publishoa.com
ISSN: 1309-3452

The proposed method is compared to other common machine learning algorithms. The findings of the suggested framework are discussed in this section. Various accuracy measurements derived from the confusion matrix are used to assess performance. As shown in (Table 2) the result of measurement when applying ensemble algorithms on the dataset to discover all sorts of attacks in dataset.

Table 2. Performance detect all types of attack in the NSL_KDD (DoS, U2R, R2L, Probe)

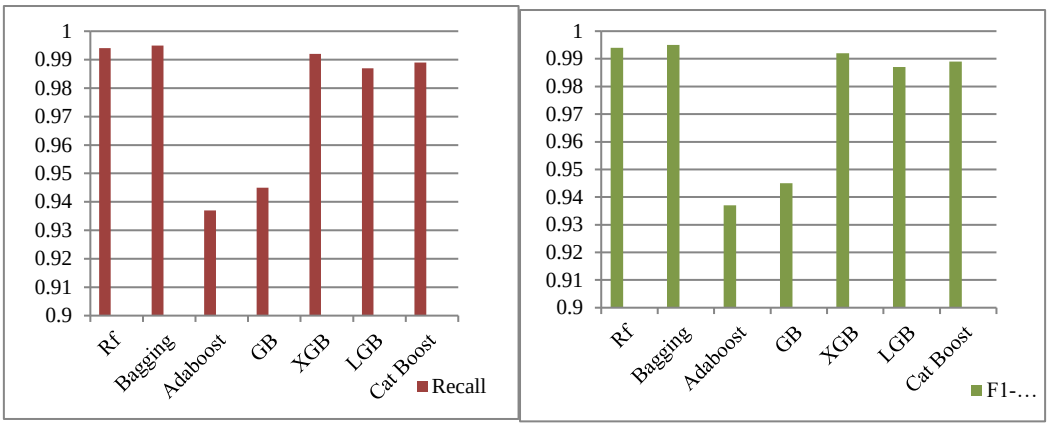
Classifier	**Accuracy**	**Precision**	**Recall**	**F1-score**
RF	0.994	0.994	0.994	0.994
Bagging	0.995	0.995	0.995	0.995
AdaBoost	0.937	0.937	0.937	0.937
Gradient Boost	0.945	0.947	0.945	0.945
Extreme Gradient Boost	0.992	0.992	0.992	0.992
Light Gradient Boost	0.987	0.987	0.987	0.987
Cat Boost	0.989	0.989	0.989	0.989

As indicated in (Table 2), bagging techniques have higher accuracy of 99.5% when using ensemble learning algorithms, however, AdaBoost has the lowest result of 93.7% for detecting four types of attacks in the dataset.
As shown in (fig. 2) the Accuracy, Precision, Recall, and F1-score were compared between the algorithms in a detect full dataset with four type's attacks such as (U2R, DoS, R2L, Probe) and found the highest and lowest accuracy of algorithms.



a. Accuracy b. Precision

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 3, 2022, p. 355-366
https://publishoa.com
ISSN: 1309-3452



c. Recall F-Score

Figure 2 Accuracy, Precision, Recall, F1-score comparisons for detecting an attack

Table 2 shows a comparison of the accuracy between different algorithms in related works on the NSL_KDD dataset.

Researchers & References	**Classifier**	**Feature selection**	**Detection Accuracy**
Rui-Hong Dong et al in 2019[8]	Boosting-C5.0	All dataset	0.80
Shadi Aljawarneh et al in 2019[4]	Enhanced J48 algorithm	All dataset	0.90
Gargi Kadam et al in 2020 [19]	RF	All dataset	0.991
Neha Singh et al in 2020[9]	SVM	All dataset	0.994
Chukwuemeka Christian et al 2021[20]	SVM	All dataset	0.956
Proposed System	Bagging	All dataset	0.995

As shown in (Table 3), when ensemble algorithms are applied to a dataset to detect only DoS attacks in the entire dataset without employing feature selection approaches, the random forest and bagging algorithms have the greatest detection accuracy of 99.9%.

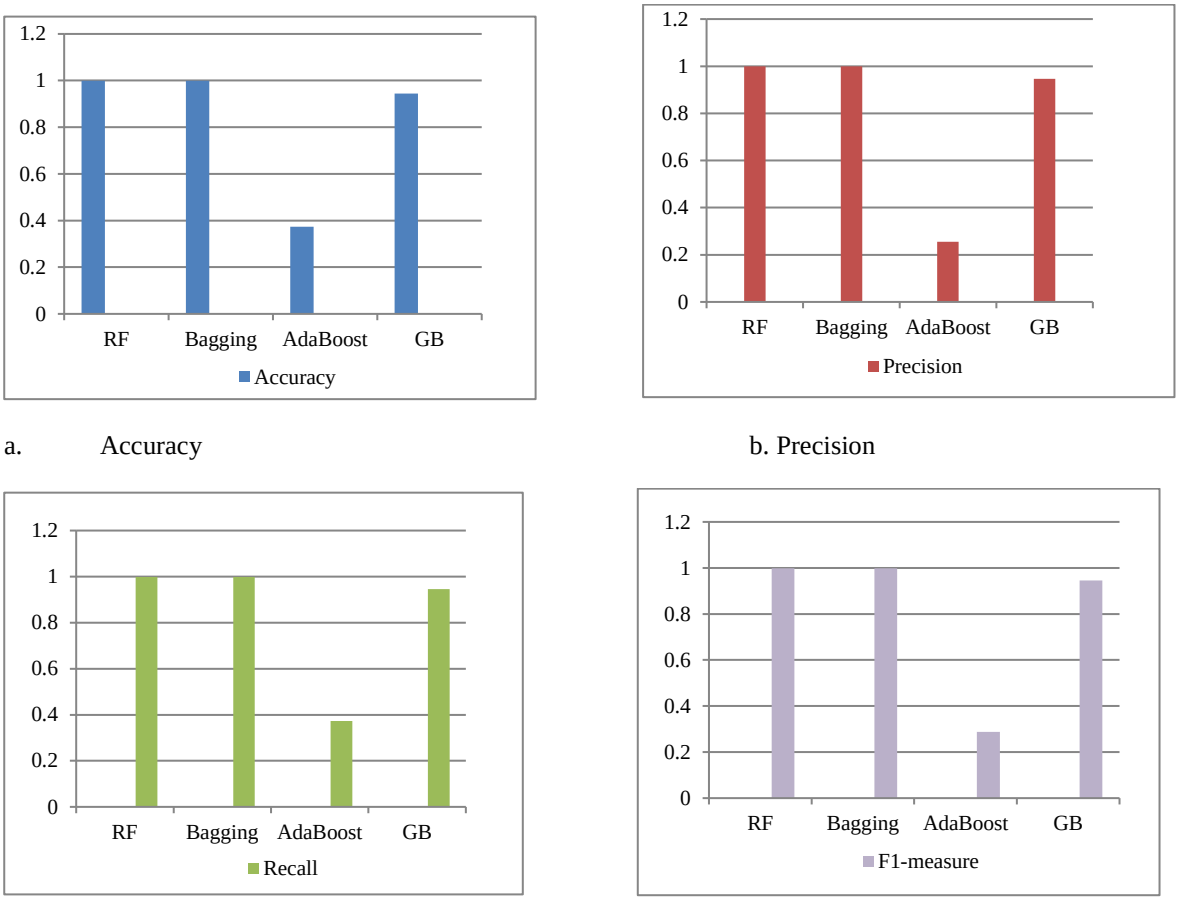
Table 3 Performance Comparison to detect only DoS attack in the NSL_KDD

Classifier	**Accuracy**	**Precision**	**Recall**	**F1-score**
Random Forest	0.999	0.999	0.999	0.999
Bagging	0.999	0.999	0.999	0.999
AdaBoost	0.373	0.255	0.373	0.288
Gradient Boost	0.945	0.947	0.945	0.945
Extreme Gradient Boost	0.999	0.998	0.999	0.998

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 3, 2022, p. 355-366
https://publishoa.com
ISSN: 1309-3452

Light Gradient Boost	0.454	0.416	0.454	0.385
Cat Boost	0.998	0.998	0.998	0.998

As shown in (fig. 3) the Accuracy, Precision, Recall, and F1-score were compared between the algorithms in the detect full dataset but only for detecting Dos attacks find the highest and lowest accuracy of algorithms.



a. Accuracy b. Precision

c. Recall d. f1-measure

Figure 3.Accuracy, precision, Recall, F1-score comparisons for detecting DoS in NSL-KDD

7. Conclusion

Network intrusion detection is one of the most recent fields of network security research. Intrusion detection systems strive to figure out whether a user's network activity is normal or suspicious, and then respond appropriately. This research used machine learning and homogeneous ensemble learning to create a categorization system. In the NSL_KDD dataset, the suggested framework was designed to detect network denial of service (DoS) attacks. Random Forest (RF), Bagging, and Boosting are examples of classification algorithms. In this experiment, The NSL_KDD dataset is used, and accuracy metrics like as Precision, Recall, F-measure, and Accuracy are used to assess performance. The suggested architecture surpassed all other classifiers in every accuracy measure, according to the findings. This research used these algorithms in two steps first to detect four types of attack in the NSL_KDD dataset, second to detect only DoS attacks and compare

the proposed system and other studies to find the highest accuracy for detection, as a result, the suggested framework's findings on the NSL_KDD dataset compassion between these steps, bagging algorithms have the highest accuracy in recognizing the four types of assault 99.5%, however, when the same techniques are used to just detect DoD types in the dataset, RF and Bagging have the maximum detection accuracy 99.9%.

References

[1] U. Kumari and U. Soni, "A review of intrusion detection using anomaly based detection," Proc. 2nd Int. Conf. Commun. Electron. Syst. ICCES 2017, vol. 2018-Janua, no. Icces, pp. 824–826, 2018, doi: 10.1109/CESYS.2017.8321199.

[2] A. A. Aburomman and M. B. I. Reaz, "A survey of intrusion detection systems based on ensemble and hybrid classifiers," Comput. Secur., vol. 65, pp. 135–152, 2017, doi: 10.1016/j.cose.2016.11.004.

[3] N. Bindra and M. Sood, "Detecting DDoS Attacks Using Machine Learning Techniques and Contemporary Intrusion Detection Dataset," Autom. Control Comput. Sci., vol. 53, no. 5, pp. 419–428, 2019, doi: 10.3103/S0146411619050043.

[4] S. Aljawarneh, M. B. Yassein, and M. Aljundi, "An enhanced J48 classification algorithm for the anomaly intrusion detection systems," Cluster Comput., vol. 22, pp. 10549–10565, 2019, doi: 10.1007/s10586-017-1109-8.

[5] A. Iqbal, S. Aftab, I. Ullah, M. A. Saeed, and A. Husen, "A Classification Framework to Detect DoS Attacks," Int. J. Comput. Netw. Inf. Secur., vol. 11, no. 9, pp. 40–47, 2019, doi: 10.5815/ijcnis.2019.09.05.

[6] G. S. Kushwah and V. Ranga, "Voting extreme learning machine based distributed denial of service attack detection in cloud computing," J. Inf. Secur. Appl., vol. 53, 2020, doi: 10.1016/j.jisa.2020.102532.

[7] C. C. Ugwu, O. O. Obe, O. S. Popoola, and A. O. Adetunmbi, "A distributed denial of service attack detection system using long short term memory with Singular Value Decomposition," Proc. 2020 IEEE 2nd Int. Conf. Cyberspace, CYBER Niger. 2020, pp. 112–118, 2021, doi: 10.1109/CYBERNIGERIA51635.2021.9428870.

[8] S. Liu, L. Wang, J. Qin, Y. Guo, and H. Zuo, "An intrusion detection model based on IPSO-SVM algorithm in wireless sensor network," J. Internet Technol., vol. 19, no. 7, pp. 2125–2134, 2018, doi: 10.3966/160792642018121907015.

[9] N. Singh and D. Virmani, "Computational method to prove efficacy of datasets," J. Inf. Optim. Sci., vol. 42, no. 1, pp. 211–233, 2021, doi: 10.1080/02522667.2020.1747193.

[10] H. Chae, B. Jo, S. Choi, and T. Park, "Feature Selection for Intrusion Detection using NSL-KDD," Recent Adv. Comput. Sci. 20132, pp. 184–187, 2013.

[11] G. Tuysuzoglu and D. Birant, "Enhanced bagging (eBagging): A novel approach for ensemble learning," Int. Arab J. Inf. Technol., vol. 17, no. 4, pp. 515–528, 2020, doi: 10.34028/iajit/17/4/10.

[12] S. K. Kiangala and Z. Wang, "An effective adaptive customization framework for small manufacturing plants using extreme gradient boosting-XGBoost and random forest ensemble learning algorithms in an Industry 4.0 environment," Mach. Learn. with Appl., vol. 4, p. 100024, Jun. 2021, doi: 10.1016/J.MLWA.2021.100024.

[13] M. Batta, "Machine Learning Algorithms - A Review," Int. J. Sci. Res. (IJ, vol. 9, no. 1, pp. 381-undefined, 2020, doi: 10.21275/ART20203995.

[14] A. Alsirhani, S. Sampalli, and P. Bodorik, "DDoS Detection System: Utilizing Gradient Boosting Algorithm and Apache Spark," in Canadian Conference on Electrical and Computer Engineering, 2018, vol. 2018-May, doi: 10.1109/CCECE.2018.8447671.

[15] L. Hu and R. Zanibbi, "Segmenting handwritten math symbols using adaboost and multi-scale shape context features," Proc. Int. Conf. Doc. Anal. Recognition, ICDAR, pp. 1180–1184, 2013, doi: 10.1109/ICDAR.2013.239.

[16] Kumar, S. (2022). A quest for sustainium (sustainability Premium): review of sustainable bonds. Academy of Accounting and Financial Studies Journal, Vol. 26, no.2, pp. 1-18

[17] Allugunti V.R (2022). A machine learning model for skin disease classification using convolution neural network. International Journal of Computing, Programming and Database Management 3(1), 141-147

[18] Allugunti V.R (2022). Breast cancer detection based on thermographic images using machine learning and deep learning algorithms. International Journal of Engineering in Computer Science 4(1), 49-56

[19] H. A. Alamri and V. Thayananthan, “Bandwidth Control Mechanism and Extreme Gradient Boosting Algorithm for Protecting Software-Defined Networks Against DDoS Attacks,” IEEE Access, vol. 8, pp. 194269–194288, 2020, doi: 10.1109/ACCESS.2020.3033942.

[20] M. R. Machado, S. Karray, and I. T. De Sousa, “LightGBM: An effective decision tree gradient boosting method to predict customer loyalty in the finance industry,” 14th Int. Conf. Comput. Sci. Educ. ICCSE 2019, no. Iccse, pp. 1111–1116, 2019, doi: 10.1109/ICCSE.2019.8845529.

[21] B. Kim, D. E. Lee, G. Hu, Y. Natarajan, S. Preethaa, and A. P. Rathinakumar, “Ensemble Machine Learning-Based Approach for Predicting of FRP–Concrete Interfacial Bonding,” Mathematics, vol. 10, no. 2, 2022, doi: 10.3390/math10020231.

[22] M. Grandini, E. Bagli and G. Visani, "Metrics for Multi-Class Classification: An Overview", arXiv:2008.05756v1 [stat.ML] 13 Aug 2020.