ISSN: 2584-0495 Vol. 4, Issue 1, pp. 1828-1833

International Journal of Microsystems and IoT

ISSN: (Online) Journal homepage: https://www.ijmit.org

An Approach to Detect DDoS Attacks in Application Layer using Machine Learning

Amardeep Kumar, Chandrashekhar Azad, Danish Ali khan

Cite as: Kumar, A., Azad, C., & khan, D. A. (2026). An Approach to Detect DDoS Attacks in Application Layer using Machine Learning. International Journal of Microsystems and IoT, 4(1), 1828–1833. https://doi.org/10.5281/zenodo.18269425

Published online: 15 January 2026

https://doi.org/10.5281/zenodo.18269425

International Journal of Microsystems and IoT
Vol.4, Issue 1, pp.1828-1833; DOI: https://doi.org/10.5281/zenodo.18269425

An Approach to Detect DDoS Attacks in Application Layer using Machine Learning

Amardeep Kumar, Chandrashekhar Azad, Danish Ali khan
Department of Computer Applications, National Institute of Technology Jamshedpur, India
2019pgcais10@nitjsr.ac.in, csazad.ca@nitjsr.ac.in, Dakhan.cse@nitjsr.ac.in

ABSTRACT

A common machine learning paradigm is ensemble learning that has shown obvious benefits in a variety of applications. In the sense of machine learning, an ensemble is a machine learning system that is built with a grouping of different models that operate in parallel and whose outputs are combined with a decision fusion strategy to generate a single response for a given problem. DDoS (Distributed Denial of Service) attacks on the application layer have boosted the effectiveness of traditional flooding-based DDoS attacks, posing a growing threat to the Internet-based web services are available. These attacks can inflict comparable damage as their lower-layer counterparts while using a smaller number of attacking assets. HTTP, as the most widely used protocol on the Internet, is a popular target of getting flooding attacks, which are used in a variety of application-layer DDoS attacks. We have suggested an alternative in this paper with an ensemble approach and compared the results by adopting machine learning base classifiers namely KNearest Neighbors, Logistic Regression, Support Vector Machine, GB Naïve Bayes, and SOM to detect the DDoS accuracy and after that, we have used different ensemble learning method namely Bagging, Random forest, Extra Tree classifier, Voting Ensemble technique, Stochastic Gradient Boosting and Boosting to detect the DDoS accuracy. According to our findings, the ensemble approach in machine learning providing better results in terms of accuracy.

Internet of Things (IoT), HTTP, Network Security, DDoS, Machine Learning

I. INTRODUCTION

Due to its simple operation and high efficiency, a Distributed denial-of-service of service attack has become a popular and important network security attack.[1] The simple definition of DDoS is, it is a type of attack where an attacker attacks a targeted server by disrupting the normal traffic signal so that the legitimate user cannot get access to that server.[2] The attacker uses bots where a bot is a piece of malicious code which comes from different locations and with the help of flooding they jam the signal and take control over the targeted server.[3] DDoS is a cyber-warfare technique that can isolate a nation from the internet. The Computer Incident advisory capability (CIAC), An organization that reported the first occurrence of DDoS attack, the previous version of this attack is known as the DoS (Denial of Service) attack in which includes a single source.[4]The first attack was reported in 1999 and till now DDoS attacks are happening day by day. Among all DDoS attacks, the largest attack happened in the year 2017 and its speed was around 60 Gbps.

The attack is classified into types DoS (Denial of service) attack and DDoS where The difference between these two types of attacks is the number of sources used in DoS attack is one whereas in DDoS, tools are available such as Trinity, Low orbit ion cannon, Trinoo, tribal flood network, and Midstream. To devise a scientific categorization of DDoS attacks, we notice the methods used to plan and play out the attack (enroll, misuse, and contaminate stages), the attributes of the attack itself (use phase), and the effect it has on the person who is associated with that organization. When a DDoS attack occurs, it is very difficult to find the origin of DDoS attacks, from where the malicious packets are coming because they are coming from different locations[5]. DDoS attackers mainly use viruses like trojan to inject a particular system. To carry out a DDoS attack, an assailant must first obtain the strength of an organization of online frameworks in a cycle. When the frameworks or different machines get contaminated with malware everyone reflects into a Bot, at that point the assailant can undoubtedly get access over the PCs through controllers. In

2018 the web has seen probably the biggest DDoS attack ever. [3] Surprisingly, as the internet grows, so does the number of DDoS attacks and malicious programmers. Fighting off new threats has almost become a daily occurrence for many larger organizations, as they must be vigilant at all times.In contrast to a few years ago, now is more critical than ever to have a DDoS insurance plan in place before anything like this happens. If your website goes down, first it gives a bad impression to potential customers and can have a variety of consequences, including lost revenue, customer loyalty, your company's overall reputation, and even worker confidence.

This paper Includes, accuracy detection of DDoS attack based on machine learning. To increase the accuracy of our predictions, we will use the Ensemble learning methodology. We used a dataset to detect DDoS attacks. [4]Application Layer DDoS Dataset)[6]and one form CAIDA (Center for Applied Internet Data Analysis) [7]. The following are our contributions to this paper

- DDoS attack detection and analyzing the accuracy of detection by using ensemble learning techniques.
- Creating a model that uses a single classifier to determine whether a packet is malicious or not.
- Combining single classifiers into a model where more than single classifiers will be there that what the meaning of the ensemble method.

The rest of the papers contains a various section whichis, Section 2 contains the associated works to DDoS identification and detection. Section 3 contains Base machine learning classifiers which we have used for comparisons with the algorithm pseudo-code. Section 4 contains a brief about the application layer and its type. Section 5 contains proposed work and section results and comparison.

II. RELATED WORKS

AU – Hosseini, [1] et al. Throughout this paper, we develop a general hybrid method for detecting DDoS attacks with progressive learning based on a continuous data framework. Kim,[4] et al. On the proxy hand, we use nave Bayes, random forest, decision tree, multilayer perceptron (MLP), and k-nearest neighbors (K-NN) to improve performance.decision tree and then created abnormality identification models (1-class SVM). They discovered that their model worked well for obscure attacks, and they also used deep learning techniques to detect attacks of DDoS in the SDN environment. They had collected traffic from a situation involving the Home Remote Organization (HWN). Also, they were successful to got 96.65% accuracy. Meng wang,[8] et al. In this paper choose a multilayer perceptron to combine sequential features During the training process, MLP was used to pick the best features, and When a significant detection error is detected dynamically, design input to reconstruct the detector. SNanda[9]used a Bayesian approach and got a 91.68 percent accuracy, which means that out of 278,597 attacks, their model could accurately predict 254,833 of them.In the SDN network, Niyaz[10] used deep learning techniques to detect the DDoS attack. They've gathered data from the HWN situation (home remote organization). Furthermore, they achieved a precision of 96.5 percent.Thilagam[11]et at. used machine learning methods such as Naive Bayes, K-nearest neighbors, K-Means, and K-medoids to detect the DDoS attack. They discovered that when they have done a comparison with different algorithms with the highest precision, the Naive Bayes model performs well.

We analyzed accuracy detection with Ensemble Learning from the literature review that it is a method in which to solve a problem, various learners are trained or planned. After that, the information is analyzed to the classification methods, who classify it before detecting the intruder data prediction. Ensemble models can provide excellent accuracy results and probability of false alarm.

III. MACHINE LEARNING CLASSIFIERS

1. Naïve Bayes

It is classification method based on Bayes' theorem with the "naive" assumption of conditional probability between any pair of features given the value of the target class.Bayes' theorem states the following relationship given a class variable y and a dependent feature vector $x1$ through xn.[12]et al. class variables are indicated in equation (1)

$$P(y|x1,\dots,xn) = \frac{P(y|x1,\dots,xn)}{P(x1,\dots,xn)} \qquad (1)$$

In Gaussian Naive Bayes, continuous values associated with each function are believed to be distributed according to a Gaussian distribution.A Gaussian distribution is another name for a normal distribution. When plotted, it results in a bell-shaped curve that is symmetric around the mean of the feature values, as shown below:

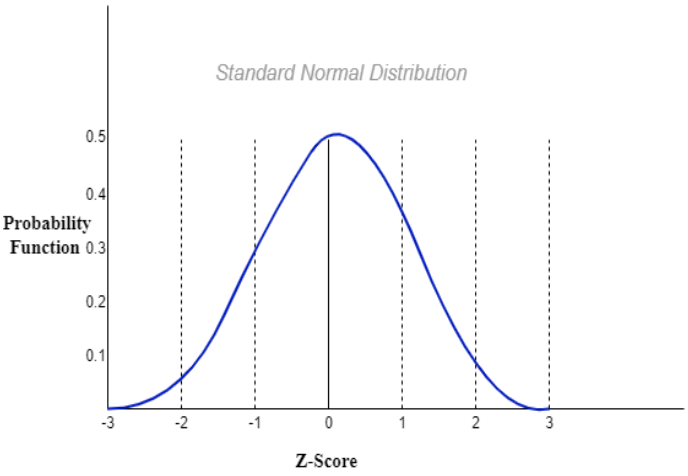


Figure 1: Gaussian Naïve Bayes Classifier

Figure 1 shows a graph representation of Gaussian distribution which is also known as standard normal distribution. In figure 1 $x-axis$ contains $z-score$ value which is also knows as standard score that represents a raw score that lies below or above the mean. The calculated value by mean of the $z-score$ is always 0 and calculated value of standard deviation is always increment of 1.$y-axis$ represents a probability function which is of bell-curved shape.

Pseudo Code for Naïve bays

Step1 : Load/import new/existing Dataset
X = Dataset.data
Y = Dataset.target
Step 2 : Store the feature matrix (X) and response vector (Y)
X_test, X_train, Y_test, Y_train = train_test_split (X,Y, test_size = 40%)
Step3: Spliting X and Y into training and testing.
Gaussian_naive_bayes = GaussianNB()
Gaussian_naive_bayes.fit(X_train,Y_train)
Step 4 : Train the model and then fit the model into required Classifier
Step 5 : Making predictions on testing dataset
Step 6 : Predict, generate confusion matrix and print the result

2. K – Nearest Neighbors

K-nearest neighbors is a non-parametric, lazy learning algorithm. Its goal is to predict the classification of a new sample point using a database with data points divided into multiple classes. For identifying the new point, it calculates the Euclidean distance for that it uses

$$D(p,q) = D(q,p) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} \qquad (2)$$

Pseudo Code for K-Nearest Neighbor

Step1 : Load/import new/existing Dataset
X = Dataset.data
Y = Dataset.target
Step 2 : Store the feature matrix (X) and response vector (Y)
X_test, X_train, Y_test, Y_train = train_test_split (X,Y, test_size = 40%)
Step 3 : Fit the KNN classifier form the training datasetE
Step 4 : Find the K-Neighbor of a point
Step 5 : Predict the probability label for the data you've been given
Step 6 : Revert the mean accuracy/result, generate the confusion matrix

3. Decision Tree

DecisionTree (DTs) are non-parametric supervised classifiers for classification or regression. The aim is to create a model that predicts the value of a dependent variable by learning basic decision rules from feature values. A tree is an approximation to a piecewise constant [13]. The workflow of decision trees is shown in Figure 2. The piecewise constant is approximated by a tree.

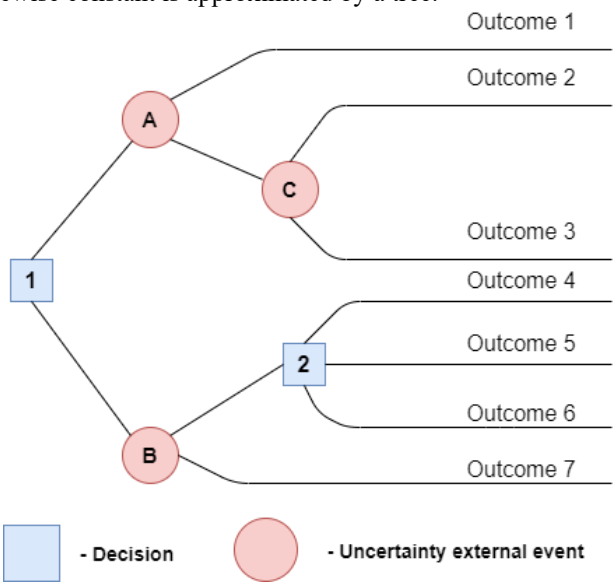


Figure 2 : Workflow of Decision Tree

4. Support Vector Machine

The aim of the support vector technique is used to find a Hyper hat divides between datasets in N-dimensional space (N — the number of features). Equation (3) is used to calculate the Weight w to each x of support vectors as follows –

$$X = \sum_{i=1}^{n} y^i * w^i * sim\,(X, V^i) \qquad (3)$$

Pseudo code for support vector machine

Step1 : Load/import new/existing Dataset
X = Dataset.data
Y = Dataset.target
Step 2 : Store the feature matrix (X) and response vector (Y)
X_test, X_train, Y_test, Y_train = train_test_split (X,Y, test_size = 40%)
Step 3 : Import Support vector machine
Step 4 : Fit train & test value into model
Step 5 : Predict the value
Step 6 : Plot values/ create confusion matric/ get accuracy.

IV. DDOS ATTACK ON APPLICATION LAYER

This section contains some common DDoS attack from which we will understand what types of attacks are there and where it is occurring[14]. Because of how quickly network bandwidth can be drained, a significant number of DDoS attacks target it [15]. Attackers simply bombard the server with a large packet over the network, completely draining the network frequency range. This is accomplished using network-layer procedure, for example UDP (User Datagram Protocol) and ICMP (Internet Control Message Protocol). Although framework layer was the prioritybecause the experiment, we have done is on Application Layer Dataset. Here are some common DDoS attacks –

- SYN flood- The TCP (Transmission Control Protocol) connection Sequence is exploited by a synchronized flood and is also known as a three-way handshake
- HTTP Flood: To use the maximum server resource of an end-user HTTP (Hypertext Transfer Protocol) floods send artificial GET or POST requests.
- UDP Floods: The target of a user datagram protocol attack is to attack random ports on a computer or network with the help of UDP (User datagram protocol) packets.
- Smurf Attack: A malware program called Smurf is used to exploit the Internet protocol and Internet Control Message Protocol.
- Fraggle Attack: In Fraggle Attack uses a large number of UDP (user datagram protocol) traffic to exploit the router's network of broadcasting, it is quite related to the Smurf attack and it uses UDP rather than ICMP (internet control message protocol).
- Shrew Attack: This attack uses the same link to target TCP using short synchronized traffic.
- Ping of Death: for a targeted system continuously pining with malicious packets affects the Internet Protocol.
- Application Layer Attack: If there is any specific weakness in the application that will lead to an attack on the entire server.
- NTP Amplification: With the help of an amplified attack it exploits the Network Time Protocol server.

Why DDoS Attacks at the Application Layer are Dangerous

Cybercriminals are still updating their toolkits and searching for new application layer attack techniques. A multi-vector cyber attack on a computer is a digital attack upon which hacker uses several entry points that involve recognizable patterns, a determined attacker can track the effects of his attack and change it to the qualified and determined defender, which makes application-layer DDoS attacks the riskiest. Since active attackers are known to change payload patterns regularly to prevent simple DDoS mitigation, keeping an up-to-date list of known attack patterns rapidly becomes inefficient due to size and the pace at which it must be modified. Maintaining a long-lived collection of payload patterns is also important because payload patterns carry a high risk of causing collateral damage

V. PROPOSED WORK

The ensemble approach utilizes multiple classifier systems and integrates many machine learning algorithms that take advantage of model efficiency to achieve higher accuracy than individual models [16]et al.

In this section, we have used ensemble learning to increase accuracy. We have taken a dataset from Kaggle 'Application layer DDoS dataset' [5] with that first we use feature selection to select the optimal features then we have applied ensemble methods, where we have used averaging, Bagging, Random Forest, Boosting, AdaBoost, etc., to get the accuracy.

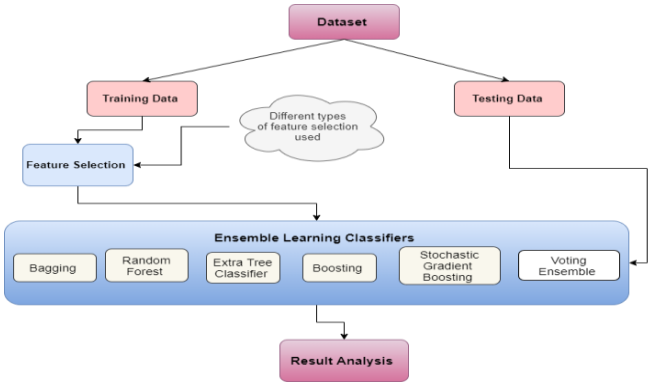


Figure 3 :Proposed Architecture

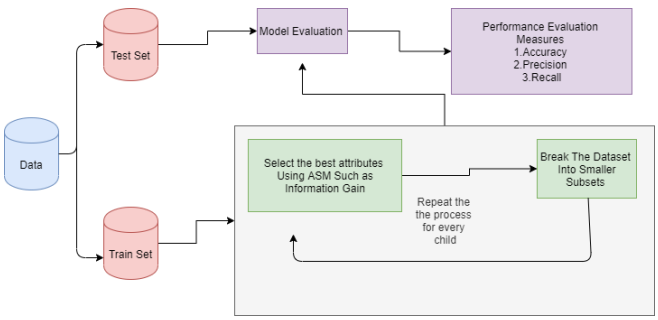


Figure 4 :General Process of DDoS detection

We have used of ensemble learning with the expectation of greater precision, false alarm rate, detection rate. For Averaging in ensemble learning we have used this formula

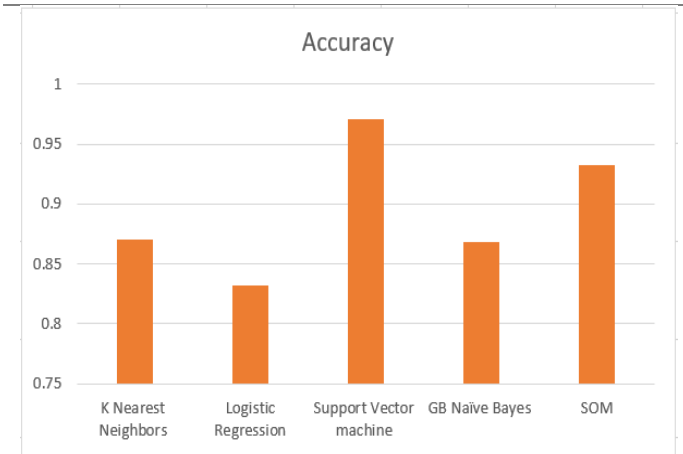
$$p = \frac{p1+p2+p3}{3}$$

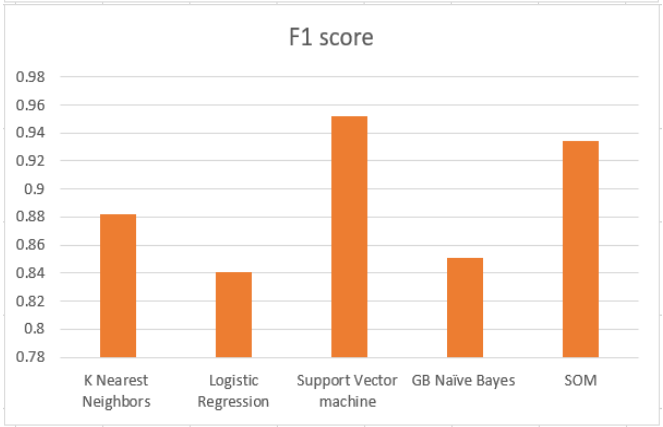
VI. RESULTS AND DISCUSSIONS

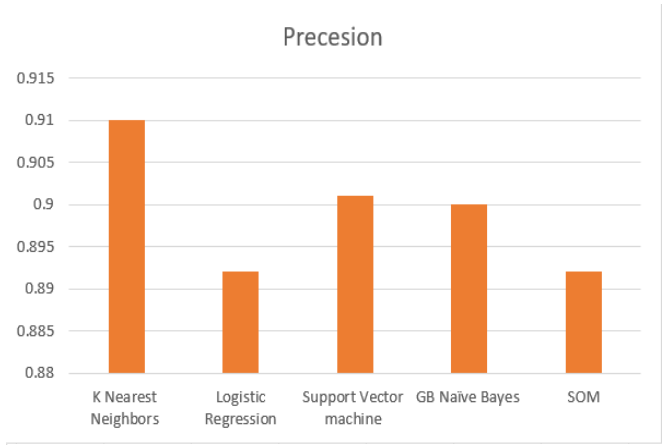
We have evaluated the results of the existing base classifier of machine learning, in which we have taken five base classifiers to find the results which include accuracy, F1 which is a function of Precision and Recall, Precession, and Recall. All the results are shown in Table 1. Whatever result we have got that all we calculated without using ensemble learning techniques.

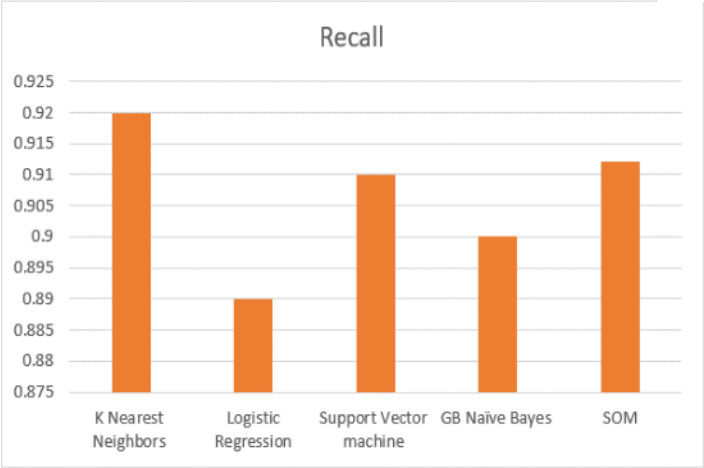
Table 1: machine learning base classifiers without ensemble learning

	Accuracy	*f1 score*	Precision	Recall
KNearest Neighbors	0.870	0.882	0.910	0.92
Logistic Regression	0.832	0.841	0.892	0.89
Support VectorMachine	0.971	0.952	0.901	0.91
GB Naïve Bayes	0.868	0.851	0.90	0.90
SOM	0.932	0.934	0.892	0.912









Now in table 2, we have calculated the different parameters like Accuracy, f1 score, precession, and recall by using ensemble learning techniques with the same dataset with their different models which are Averaging, Bagging, Boosting, AdaBoost, and Gradient Boosting*Table 2 : Accuracy with Ensemble Learning Methods*

Ensemble Method	Accuracy	Model / Classifiers used
Bagging	0.937	KFoldDecision Tree
Random Forest Classifier	0.940	KFold
Extra Tree Classifier	0.942	KFold Cross Validation
Boosting Classifier	0.935	KFold Cross Validation
Stochastic Gradient Boosting	0.946	KFoldCross Validation
Voting Ensemble	0.941	KFold, Logistic, Decision Tree, Support Vector Machine

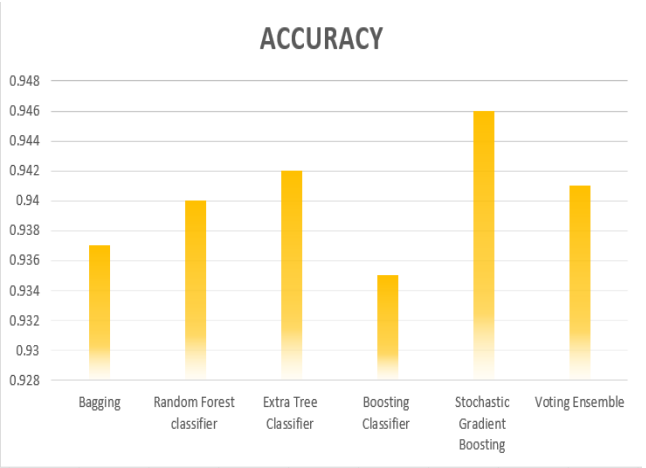


Table 2shows the result of Accuracy with the help of ensemble learning techniques, in which different types of ensemble learning methods are used which are Bagging, Random Forest Classifier, Extra Tree Classifier, Boosting, Stochastic Gradient Boosting, and Voting Ensemble. Above all methods, we have applied the same dataset which we used for base machine learning classifiers and got a result. By analyzing all result we found that the accuracy with help of base classifiers are KNN 87%, Logistic Regression 83.1%, SVM 97.1%, Naïve Bayes 86.8%, and SOM 92.3 after that we analyze the results which came from ensemble learning methods where with different ensemble methods we got the different results like from Bagging we have the accuracy rate is up to 93.7%, Random forest Classifier 94%, Extra Tree Classifier 94.2%, Boosting Classifier 93.5%, Stochastic Gradient Classifier 94.6% and with Voting ensemble we have got an accuracy of 94.1%.

VII. CONCLUSION

We have used ML classification algorithms in this paper. where we have used base classifiers as KNN, Logistic Regression, SVM, GB Naïve Bayes and SOM, and Ensemble learning methods which are Bagging, Random Forest, Extra Tree Classifier, Boosting, Stochastic Gradient Boosting and Voting Ensemble to identify a distributed denial-of-service (DDoS) attack in Application Layer. We evaluated the proposed work based on factors such as accuracy, f1 score, Precision, and Recall where we mainly focused on accuracy. We have analyzed that accuracy is more by using ensemble learning methods in which we have achieved that the Stochastic Gradient Boosting is giving the more accuracy.In the future, we'd like to work with a new set of data to implement proper classification system selection and evaluate outcomes.

REFERENCES

[1] S. ,. -. A. M. ,. AU - Hosseini, "The hybrid technique for DDoS detection with supervised learning algorithms," 2019.

[2] T. D. Khundrakpam Johnson Singh, "MLP-GA based algorithm to detect application layer DDoS attack,," in *https://doi.org/10.1016/j.jisa.2017.09.004.*, 2017.

[3] V. M. R. V. S. H. M. C. Aanshi Bhardwaj, "Distributed denial of service attacks in cloud: State-of-the-art of scientific and commercial solutions,," in *https://doi.org/10.1016/j.cosrev.2020.100332.*, 2021.

[4] G. &. L. S. &. K. S. Kim, "A novel hybrid intrusion detection method integrating anomaly detection with misuse detection.," in *Expert Systems with Applications. 41. 1690–1700. 10.1016/j.eswa.2013.08.066.* , 2014.

[5] P. R. I. D. M. M. M. F. Vinícius de Miranda Rios, "Detection of reduction-of-quality DDoS attacks using Fuzzy Logic and machine learning algorithms,," in *https://doi.org/10.1016/j.comnet.2020.107792.*, 2021.

[6] kaggle, Application layer dataset, https://www.kaggle.com/wardac/applicationlayer-ddos-dataset.

[7] CAIDA, The CAIDA UCSD DDoS Attack 2007 Dataset, https://www.caida.org/passive/ddos-20070804_dataset.xml, 2007.

[8] Y. L. J. Q. Meng Wang, "A dynamic MLP-based DDoS attack detection method using feature selection and feedback,," in *(https://www.sciencedirect.com/science/article/pii/S0167404819301890)*, 2020.

[9] F. Z. C. D. E. W. a. B. Y. S. Nanda, "Predicting network attack patterns in SDN using machine learning approach," in 2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), Palo Alto, CA, USA, 2016, pp. 167-172, doi: 10.1109/NFV-SDN.2016.7919493., 2016.

[10] Q. a. S. W. a. J. A. Y. Niyaz, "A Deep Learning Based DDoS Detection System in Software-Defined Networking (SDN)," in *European Alliance for Innovation n.o.*, 2017.

[11] A. P. a. P. S. Thilagam, "DDoS Attacks at the Application Layer: Challenges and Research Perspectives for Safeguarding Web Applications," in *IEEE Communications Surveys & Tutorials, vol. 21, no. 1, pp. 661-685, Firstquarter 2019, doi: 10.1109/COMST.2018.2870658.*, 2019.

[12] M. A. Soodeh Hosseini, "The hybrid technique for DDoS detection with supervised learning algorithms," 2019.

[13] E. A. -. M. K. TY - JOUR AU - Fenil, "Survey on DDoS defense mechanisms," 2020.

[14] A. P. a. P. S. Thilagam, "DDoS Attacks at the Application Layer: Challenges and Research Perspectives for Safeguarding Web Applications," in *IEEE Communications Surveys & Tutorials, vol. 21, no. 1, pp. 661-685, Firstquarter 2019, doi: 10.1109/COMST.2018.2870658*, 2018.

[15] A. S. N. M. D. G. N. a. M. M. M. L. Barki, "Detection of distributed denial of service attacks in software defined networks," in *International Conference on Advances in Computing, Communications and Informatics (ICACCI), Jaipur, India, 2016, pp. 2576-2581, doi: 10.1109/ICACCI.2016.7732445.*, 2016.

[16] G. V. A. G. V. M. B. T. O. G. M. B. P. P. R. W. V. D. J. V. A. P. D. C. M. B. M. P. Fabian Pedregosa, "Scikit-learn: Machine Learning in Python," in *https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html*, 2011.

[17] N. S.-M. A. A.-B. V. Bolón-Canedo, "An ensemble of filters and classifiers for microarray data classification," in *https://doi.org/10.1016/j.patcog.2011.06.006.*, 2012.

[18] P. S. K. K. Karanpreet Singh, "Application layer HTTP-GET flood DDoS attacks: Research landscape and challenges," in *https://doi.org/10.1016/j.cose.2016.10.005.*, 2017.

[19] K. M. S. a. P. D. V. Deepa, "Detection of DDoS Attack on SDN Control plane using Hybrid Machine Learning Techniques," in *2018 International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 2018, pp. 299-303, doi: 10.1109/ICSSIT.2018.8748836.*, 2018.

[20] S. K. V. G. K. N. F. a. S. B. K. C. K. Aridas, "Uncertainty Based Under-Sampling for Learning Naive Bayes Classifiers Under Imbalanced Data Sets," in *IEEE Access, vol. 8, pp. 2122-2133, 2020, doi: 10.1109/ACCESS.2019.2961784.*, 2020.