International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 06 | June - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

Classification and Prediction of Distributed Denial of Service Attacks using Random Forest and a Comparative Analysis of Machine Learning Algorithms

Sandhiya M[1], Ramyasri G[2], Tharun surya M[3]

S. Kavishree, M.Tech, Assistant Professor,

[1]Department of Information Technology, Sri Venkateswara College of Engineering
[2]Department of Information Technology, Sri Venkateswara College of Engineering
[3]Department of Information Technology, Sri Venkateswara College of Engineering

---

Abstract - Hackers or cybercriminals utilize DDoS (Distributed Denial of Service) attacks, a type of malicious cyber-attack, to render an online service, network resource, or host system inaccessible to its intended users on the Internet. The proposed system forecasts DDoS attacks using random forest, and we also compared four different classification techniques. The Random Forest, Logistic Regression, Support Vector Classifier, K-Nearest Neighbors, and Decision Tree algorithms are the machine learning models on which this project is based. Python was utilized as a simulator and the DDoS dataset was used to achieve the proposed goal. In order to identify the model performance after applying the machine learning methods, we built a confusion matrix.

Key Words: Random Forest, DDoS, Machine Learning, Classification, attacks.

1.INTRODUCTION

Nowadays with the advent of 4G, 5G networks and economic smart devices there is a massive growth in the usage of the internet that has become a part of daily life. A vast range of services provided over the internet in diverse application areas such as business, entertainment, and education, etc. made it a vital component in framing various business models. This context made security over wireless networks as the most important factor while using the internet from unsecured connections [1]. Different security algorithms and frameworks are developed to enable protection from Internet-based attacks while devising high performance IDS (Intrusion detection systems) which act as a defensive wall while confronting the attacks over internet-based devices. Distributed Architecture based computing environments like cloud computing and IoT are more prone towards DDoS attacks in which multiple devices are coordinated to launch attacks over distributed targets. DDOS attacks are primarily launched in the context of exhausting the connectivity and the processing of the target server resources in which it enables the access constraints to the legitimate users to utilize the services provided by the target server that leads towards the partial unavailability or total unavailability of the services.

There are many types of attack in NIDS. However, this paper focuses on distributed denial of service (DDoS) attacks. DDoS is very similar to denial-of-service attacks. The difference is that the second has a single source of attack, whereas the first has multiple sources of attack. Both types of attack result in inaccessible network resources, due to the complete consumption of the network resources by these attacks. The challenge for an effective NIDS is to have a high accuracy rate with a low false positive rate and a low false negative rate. These are some of the main metrics currently being emphasized for NIDS research.

As an outcome of several research studies, there are several statistical mechanisms to detect the intrusions in the network traffic on analyzing the source and destination IP address, detection based on the port degeneration values, destination decay and wavelet-based analysis, etc [4]. With the massive usage of cloud computing and IoT technologies, the model for DDoS attack has been changing frequently with the frameworks of computing. Design and development of the novel statistical models are time-consuming as it will not be able to sustain rapid and dynamic changes within the network. The major drawback observed while constructing the statistical model is that it is bounded towards a single application scenario and the range of complexity in building and maintain the model. In the context of resolving the problems of the statistical models in detecting and predicting DDoS attacks, the researchers have focused on the deep and machine learning algorithms to develop context-aware prediction models that are bounded to be less complex and high performance-centric. It is evident from various research studies that Machine learning algorithms have demonstrated high performance while adopting towards the dynamic changes within the network and predicting the network traffic along with the intrusions within the network. Machine learning and deep learning algorithms have the ability to identify unconstrained information within massive amounts of data which draws the attention of various researchers to study the application of these strategies. Researchers in [2] have utilized the access patterns of various clients, flow size constrained to the network traffic and chronological behavior while devising machine learning models to classify abnormal network from a normal network in the circumstance of controlling the servers. The major advantage of machine learning models is that data is updated dynamically within the prediction model such that the changes within the network could be easily identified. Few studies evident that still there are few deficiencies while adopting machine and deep learning algorithms because of its substantial computational complexity. DDoS attack patterns vary from different network components. Primarily DDoS attacks involved in devastating the target remote server or network traffic towards the server could be categorized into three categories that include application layer-based attacks, Protocol level attacks, and Network traffic attacks.

The contribution of this article is twofold. Firstly, we perform classification and prediction of DDoS attacks using random

forest. Secondly a comparison of various machine learning algorithms in a distributed denial of service attacks. The rest of the article is organized as follows: Section 2 describes the related work that has been done and reviews the current studies related to machine learning approaches in detecting network attacks; Section 3 presents a methodology and dataset we used for the experiments; Section 4 shows the experiments we performed using ML algorithms; Section 5 discusses the results obtained; finally, in Section 6, the conclusions and future work are presented.

2. RELATED WORK

Zekri et al. [3] focused on how Distributed Denial of Service affects cloud performance by utilizing network resources. The attack techniques implemented and evaluated different ML algorithms in a cloud computing environment. The authors presented a DDoS protection design, and the algorithms they implemented and evaluated in cloud environments were Naive Bayes, Decision Tree (C4.5), and K-Means (KM). The results showed that their accuracy and detection time of algorithms Naive Bayes, Decision Tree (C4.5), K-Means were 91.4% in 1.25 s, 98.8% in 0.58 s, and 95.9% in 1.12 s, respectively. This approach can work on real-time anomaly detection and mitigation techniques and other security challenges. The drawback of this approach is that they evaluated only a few models to detect DoS attacks. Priya et al. [5] proposed an ML-based model for the detection of DDoS attacks. The authors applied three different machine learning models: K-Nearest Neighbors (KNN), Random Forest (RF) and Naive Bayes (NB) classifier. The proposed approach can detect any type of DDoS attack in the network. The results of the proposed approach showed that the model can detect attacks with an average accuracy of 98.5%.

Saravanan [6] presented a classification algorithm that works on network security data for intrusion detection. Multiple classification algorithms were implemented and evaluated using the big data tool Apache Spark and training time and testing time were measured. However, the authors evaluated few classification algorithms. As compared to the existing systems, they found better results with a good false-positive ratio. Better results were found for assessing the classification algorithm in Apache Spark on network security data than in existing systems. The accuracy of algorithms Decision Tree (DT), Logistic Regression (LR), Support Vector Machine (SVM) and SVM with Stochastic Gradient Descent (SGD) were 96.8%, 93.9%, 92.8%, and 91.1%.

Gadze et al. [7] proposed deep learning models to detect and mitigate the risk of DDoS attacks that target the centralized controller in Software Defined Network (SDN) through Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN). The accuracies of the models were lower. LSTM and CNN were 89.63% and 66%, respectively, when data spitted was in 70/30 ratio. However, in the case of LSTM model to detect TCP, UDP and ICMP, DDoS was the most time-consuming among the 10 attempts.

3. MATERIALS AND METHODOLOGY

3.1 Dataset

We used the application layer DDoS dataset available on Kaggle. The dataset belongs to a large dataset category and consists of around 0.9 million records with 77 features columns and a target column. Mainly it consists of three labels: (1) HTTP; (2) UDP; and (3) TCP.

This study was performed using Jupiter notebook and python as simulator installed on a Win7 OS. The hardware specifications of the laptop were Intel i5 CPU, 1.4 GHz 64-bit processor, 8 Gb RAM, and Intel HD Graphics 3000.

3.2 Preprocessing

Raw data normally include many imperfections such as missing value, redundancies and inconsistencies. Therefore, pre-processing is required to produce a clean dataset that will ensure that an ML technique can build and train a model smoothly without any errors [8].

3.2.1 Missing data

Missing data include empty values or values not compatible with the data format. For example, features with numerical formats must consist of numbers only, and cannot include any symbols or alphabetic characters. The simplest approach is to discard or remove such data all together. However, these data points could be important; therefore, we use the maximum likelihood approach [15]. All missing data were replaced with linearly interpolated values.

3.2.2 Class Labels.

Each dataset instance represents a snapshot of the network traffic at a given point in time. These instances are labelled according to the nature of the traffic, that is, whether the traffic is benign or malicious. The labels across the four datasets vary, therefore they are encoded to have homogeneity in the class labelling system. Classification is binary, where benign traffic is labelled as NORMAL, and malicious traffic is labelled as DDOS ATTACK. Table 1 summarizes the classification system.

Table-1: Labelling system for binary classification

Label	Scenario
NORMAL	Traffic is benign
DDOS ATTACK	Traffic is malicious

3.2.3 Feature Selection

All the features that have min, max, mean and std (standard deviation) values were removed except the mean value of the feature because these values refer to the same features but use different calculated values and also features with non-numerical data were also removed to improve the algorithm performance, we removed these features 38 features.

3.3 Machine learning

We perform classification and prediction of DDoS attack using random forest as random forest algorithm is

approximately 100 times faster than other algorithms and best working for classification problems and also perform a comparative analysis of machine learning algorithms named Logistic regression, Support vector classifier, K-Nearest Neighbors and Decision tree. These five algorithms are supervised algorithms that can produce binary classification.

3.3.1 Logistic Regression

This algorithm uses a regression model to find the best-fitting model that describes a dependent variable based on a set of independent variables. The outcomes of the dependent variable consist of only two possible values: true or false. Therefore, it is well suited for binary classifications.

3.3.2 Support Vector Machine (SVM)

This algorithm finds the optimum hyperplane that separates two classes with the maximum distance between the border points of each class. These border points form the support vector. Therefore, it is effective for high-dimensional space problems, and is memory efficient. However, if the feature count is larger than the number of samples, this technique will have only a mediocre performance [9].

3.3.3 Decision Tree

The Decision Tree classification algorithm works as a human thinking ability while making a decision. The classification model is created by the decision tree algorithm, which generates a decision tree. Each branch descending from that node represents one of the possible values, and each node in a decision tree represents a test for a feature. Because the core structure of Decision Trees is unaffected by the values taken by each feature, they can function efficiently with unnormalized datasets.

3.3.4 Random Forest

The basic idea behind random forest algorithm is to create a large number of decision trees, each trained on a random subset of the data and a random subset of the features. The final prediction is then made by taking the majority vote of all the individual tree predictions. When dealing with a dataset with a huge number of features, the decision tree algorithm is prone to overfitting, which complicates the model and learning process. Random Forest (RF) classification algorithm is an ensemble Decision Tree classification algorithm that incorporates several weaker models to build a more accurate one.

3.3.5 K-Nearest Neighbors

K-Nearest Neighbors (KNN) is a non-parametric, the lazy classification algorithm that memorizes class labels rather than learning how to discriminate them. To classify or predict a new data point using the KNN algorithm, we first need to calculate the distance between the new data point and all other datapoints in the dataset. The most commonly used distance metric is Euclidean distance. Once the distances are calculated, we select the K nearest neighbors to the new data point.

3.5 Performance Metrices

Metrices are used to quantify the ML performance. Such metrices can be calculated based on a confusion matrix as shown in Table 2 [10].

3.5.1 Accuracy

This metric determines the accuracy, all correct prediction, of the model. It is the model abilities to predict both positive and negative results correctly.

Accuracy: $$\frac{TP + TN}{TN + TP + FP + FN}$$

3.5.2 True Positive Rate (TPR)

This metric calculates how often the model is able to predict a positive result correctly. Similar to Accuracy, but difference is it only takes positive observation.

TPR: $$\frac{TP}{TP + FN}$$

3.5.3 False Alarm Rate (FAR)

This metric calculates how often the model is predicting a positive result wrongly. It provides indication of possible

error of the model, thus lower value is better.

FAR: $$\frac{FP}{FP + TN}$$

Table-2: Confusion matrix table

		Predicted Class	
		Negative (Normal)	Positive (Attack)
Actual Class	Negative (Normal)	True Negative (TN)	False Positive (FP)
	Positive (Attack)	False Negative (FN)	True Positive (TP)

4 RESULTS

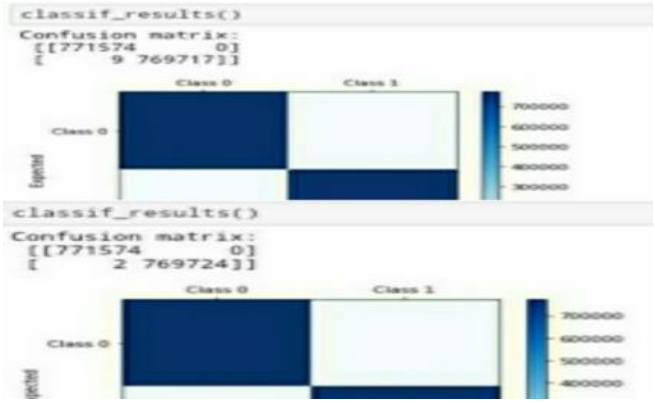
We have made a model for classification and prediction of DDoS attacks using random forest which is shown in Fig 1. We also performed comparative analysis using various ML algorithms and their results are shown in Table 3. After applying the machine learning algorithms, we generated a confusion matrix for identification of the model performance which is shown in below figures.

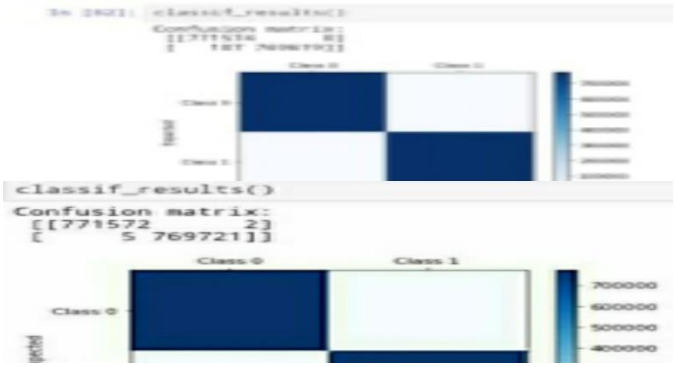
5 DISCUSSIONS

It appears that the dataset used, CICIDS2017, is well suited for DT algorithm and its derivatives, such as BT and random forest. Random forest produced the best result with the highest accuracy of 99.99%, followed by DT. This study primarily focused on common classifiers; future studies should use more advanced hybrid algorithms to test the same. In addition to much larger sample data, the full dataset includes six types of attacks.

Table - 3: Result of the five classification algorithms using the dataset

Algorithm	Accuracy (%)	TPR (%)	FAR (%)
Logistic Regression	99.97	99.95	0.04026
Support Vector Machine	99.99	99.99	0.00025
Decision Tree	99.99	99.99	0
Random Forest	99.99	99.99	0
K-Nearest Neighors	99.99	99.99	0





6 CONCLUSIONS

This paper aims to compare nine supervised algorithms' performance towards DDoS intrusion. DDoS attack will result in inaccessible to network resources, due to complete consumption of the network resources. The random forest algorithm produced the best result with an accuracy of 99.99%. This ensemble classifier, which uses the bagging method, can handle outliers and noise in the dataset, which makes it less susceptible to over-fitting. However, random forest took a relatively longer time to compute compared to the other algorithms. Therefore, there is room for improvement and fine-tuning the model could allow it to work in more efficient manner. The supervised method produced predictions and classification results by the model. Then we employed the categorization methods such as Random Forest, Decision Tree, Logistic Regression, K-Nearest Neighbors' and Support Vector Machine. In comparison to other corresponding classifiers, Random Forest has the highest accuracy of 99.99%, while Logistic Regression has the lowest accuracy.

REFERENCES

1. Mirkovic, Jelena, and Peter Reiher. "A taxonomy of DDoS attack and DDoS defense mechanisms." ACM SIGCOMM Computer Communication Review 34.2 2004, 39-53.

2. Dietrich, Sven, Neil Long, and David Dittrich. "Analyzing Distributed Denial of Service Tools: The Shaft Case." LISA. 2000, pp. 329-339.

3. Zekri, M.; El Kafhali, S.; Aboutabit, N.; Saadi, Y. DDoS attack detection using machine learning techniques in cloud computing environments. Proceedings of the 2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech); Rabat, Morocco, 24–26 October 2017; pp. 1-7.

4. Barbara Kitchenham and Pearl Brereton. A systematic review of systematic review process research in software engineering. Info. Softw. Technol. 55, 12 (2013), 2049–2075.

5. Priya, S.S.; Sivaram, M.; Yuvaraj, D.; Jayanthiladevi, A. Machine learning based DDoS detection. Proceedings of the 2020 International Conference on Emerging Smart Computing and Informatics (ESCI); Pune, India, 12–14 March 2020; pp. 234-237.

6. Saravanan, S. Performance evaluation of classification algorithms in the design of Apache Spark based intrusion detection system. Proceedings of the 2020 5th International Conference on Communication and Electronics Systems (ICCES); Coimbatore, India, 10–12 June 2020; pp. 443-447.

7. Gadze, J.D.; Bamfo-Asante, A.A.; Agyemang, J.O.; Nunoo-Mensah, H.; Opare, K.A.-B. An Investigation into the Application of Deep Learning in the Detection and Mitigation of DDOS Attack on SDN Controllers. Technologies; 2021; 9, 14.

8. Ramírez-Gallego S, Krawczyk B, García S, Woźniak M, Herrera F. A survey on data preprocessing for data stream mining: Current status and future directions. Neurocomputing. 2017; 239:39–57.

9. Informatic S, Science C, Science C, Intelligence M, Labs MIR, Roy ss. random forest, support vector machine and nearest centroid methods for classifying network intrusion. Comput Sci Ser. 2016; 14:9–17.

10. Wu SX, Banzhaf W. The use of computational intelligence in intrusion detection systems: A review. Appl Soft Comput J. 2010;10(1):1–35.

11. Chowdhury N, Ferens K, Ferens M. Network Intrusion Detection Using Machine Learning. 2010;30–5.

12. Zou X, Feng Y, Li H, Algorithm MC, Hamid IRA, Syafiqah N. Performances of Machine Learning Algorithms for Binary Classification of Network Anomaly Detection System Performances of Machine Learning Algorithms for Binary Classification of Network Anomaly Detection System. 2018;

13. Biswas SK. Intrusion Detection Using Machine Learning: A Comparison Study. 2018;118(19):101–14.

14. Kumarasamy, S., &Asokan, R. (2012). Distributed Denial of Service (DDoS) Attacks Detection Mechanism. arXiv preprint arXiv:1201.2007, pp. 41-49.

15. García S, Luengo J, Herrera F. Tutorial on practical tips of the most influential data preprocessing algorithms in data mining. Knowledge-Based Syst. 2016; 98:1–29.