Turkish Journal of Computer and Mathematics Education Vol.12 No.12 (2021), 1647-1655

Research Article

DDoS Detection Using Machine Learning Ensemble

Arpit Kumar Jain[a], Himanshu Dhawan[b], B.Sowmiya[c]

[a,b,c]SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu
email: Nadu aa8423@srmist.edu.in[a] ,hp8319@srmist.edu.in[b],sowmiyab@srmist.edu.in[c]

Article History: Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 23 May 2021

Abstract: In the era of internet and online connectedness, where data is the most valuable asset, it is ever important for an organization to protect itself and it's assets from various security threats. One of these threats is a Distributed Denial of Service (DDoS) attack that can cut off the network service by overwhelming the targeted server or network by flooding it with superfluous requests in an attempt to overload the server to prevent legitimate requests from being fulfilled. DDoS attacks utilize multiple compromised systems as sources of internet traffic to increase their effectiveness. What makes DDoS attacks more lethal is that fighting them requires differentiating legitimate requests from illegitimate ones. A site or service unexpectedly being sluggish or

inaccessible is the most obvious symptom of a DDoS attack. But since a number of causes like legitimate spike

in network traffic can create similar issues , further investigation is necessary.)

Keywords: DDoS Detection, Intrusion Detection, Machine Learning, Distributed Denial of Service, Naive Bayes, SVM, KNN, Random Forest

1. Introduction

Distributed Denial of Service (DDoS) attacks have been one of the most prominent attacks over the last decade. Distributed denial of service (DDoS) attack is an effort to make an online service unavailable to legitimate users by overwhelming it with traffic from multiple sources. These sources are generally computers that are infected and used as a bot in botnets. DDoS attacks are costly to an organization or company as they prevent legitimate users from accessing the resources.

Consequently, it is important to propose an effective method for detecting DDoS attacks from massive data traffics. The existing methods, however, do have limitations, such as the need for large labeled dataset for supervised learning methods, and the relatively low accuracy and high false positive rate for unsupervised learning algorithms. In order to combat these issues, this paper presents a hybrid approach that uses a combination of four different classifiers - Naive

Bayes, SVM, KNN, Fuzzy c-means and Random Forest. Most of the research has been done

using older datasets like KDD99 , NSL KDD and DARPA. The models that are trained on these older datasets are found to be less accurate and inconsistent. Therefore, we are using a new and improved dataset - CICDDoS2019.

2.Literature Review

Wu Zhijun, Xu Qing, Wang Jingie, Yue Meng, Liu Liang [2] proposes a multi feature DDoS attack detection based on FM and uses a combination of Support Vector Machine(SVM) and Self-Organizing Mapping(SOM) to detect DDoS. This works for special data instead of general prediction tasks.

Shi Dong, Mudar Sarem [3] used an improved K-Nearest Neighbour(KNN) and four features(flow length, flow size, flow ratio) to detect DDoS attacks. Uses a combination of DDADA and DDAML algorithms but some further research is needed.

Sabah Alzahrani, Liang Hong [5] had a signature based artificial neural network(ANN). It has a signature based approach where if the attack has a known signature then a predefined approach is followed and if not then an anomaly detection distributed neural network will be used to detect the unknown DDoS attack.

Saikat Das and team [6] uses an ensemble of different techniques like Artificial Neural Network (ANN), Support Vector Machine (SVM), Multilayer Perceptron (MLP), Na¨ıve Bayes (NB), Multivariate Adaptive Regression Splines (MARS), K Nearest Neighbor (KNN).

Suman Nandi,Santanu Phadikar, Koushik Majumder [7] uses combination of Naive Bayes, Bayes Net, Decision Table, J48 and Random Forest and a five feature selection method like information gain, gain ratio, chi-squared, reliefF, and symmetrical uncertainty.

Petr Blazek, Tomas Gerlich, and Zdenek Martinasek [8]uses forward Neural Networks (NN) as an anomaly detection method and a signature based intrusion detection system Suricata which is an open source threat detection engine and has been owned and implemented by Open

Information Security Foundation.

Wenwen Sun, Yi Li, Shaopeng Guan [9] first uses the entropy to detect whether the flow is abnormal or not. The BLSTM-RNN neural network algorithm is used to train the data set, and the model is used to detect DDoS attacks on real time traffic.

Roshni Mary Thomas,Divya James [10] uses a traffic monitoring method in the server to check the traffic for a specific amount of time. iftop is a traffic monitoring tool to find the bandwidth of incoming packets along with the address.

Swati Sahu, Amit Verma [12] detects the type of DDoS attack then the traffic is categories into normal or malicious. Again a filtering is done to categories it as either suspicious or normal. If suspicious then passed to a honey pot. It is set up on a server level and not on the subscriber level. The Honey Pot assumes that the attack must be detectable using a signature based detection tool.

Ruchi Vishwakarma and Ankit Kumar Jain [13] use a honey pot to intentionally lure in attackers with the purpose to capture the malware properties, the signature, the style of invading and capture the whole information inside of a log file. A detection framework is used to predict the abnormal activities based on the log files generated in the Honey Pot.

Yuze SU and team [14] used a phase space reconstruction technique to denoise the original traffic. RBF neural network is used to train the network traffic sample and predict the future incoming network traffic for DDoS attacks.

S.Shanmuga Priya, M.Sivaram, D.Yuvaraj, A.Jayanthiladevi [15] uses three classification algorithms that are KNN, RF, NB to classify DDoS packets from normal packets into two features which are delta time and packet size.

Obaid Rahman and [16] uses a combination of J48, RF(Random Forest), SVM(Support Vector Machine), KNN(K-Nearest Neighbour) to detect and block the DDoS attack in the SDN network.

Gaganjot Kaur, Prinima Gupta [17] makes the use of Bayesian Network, Wavelets, SVM(Support Vector Machine) and KNN(K-Nearest Neighbour). It has parameters like packet flow, time duration, accuracy and precision rate which is applied on the KNN dataset.

V.Deepa, K.Muthamil Sudar, P.Deepalakshmi [18] uses a hybrid approach combining both SVM(Support Vector Machine) and SOM(Self Organized Map) or using them stand alone also.

Shuang Wei, Shuaifu Dai, Xinfeng Wu, Xinhui Han [19], it uses a two stage hierarchical architecture. In the first stage it inputs the flow information gathered controller to do clustering. DDoS attacks are detected using KL distance between the real time flow distribution with the distribution during normal time and the packet rate.

Sanjeetha. R and team [20] uses a flow entry table for sending requests as PACKET IN which is then compared to the global table. It is observed that the request is received before the set idle time out for that particular flow table entry, the switch is identified as compromised and the mitigation is done by just blocking the compromised switch.

3.Proposed Work

The purpose of our research is to build an accurate DDoS detection model with a low false positive rate . Here , we propose a model based on an ensemble of five machine learning classifiers . The selected classifiers are Na¨ıve Bayes , SVM , KNN , Fuzzy c-means and Random

Forest . In our model all five classifiers work independently and build a different model of the data . The outputs of the five classifiers are combined by a majority voting method to obtain a final result of the model . CIC-DDoS2019 dataset is used to train the model. This dataset has 80 features. As the dataset is very large and highly dimensional, it must be transformed into a smaller one while still keeping most of the information. This is done using Principal Component Analysis. The resultant dataset is then used to train and test the model.

The Proposed Model has three module:

1. Data Preprocessing
2. Data Classification

3. Ensemble with Majority Voting

3.1.Data Preprocessing

Dataset Used: The Dataset used for training and testing the model is "DDoS Evaluation Dataset (CIC-DDoS2019)" . The dataset was chosen based on earlier research . It fixes various issues that are in NSL KDD , DARPA 99 and CIADA 2007. The datasets have a total of 80 features. The dataset contains network traffic from 12 different DDoS attacks including NTP, DNS, LDAP, MSSQL, NetBIOS, SNMP, SSDP, UDP, UDP-Lag, WebDDoS, SYN PortScan and TFTP. The dataset has a total of 80 different network traffic features.

Attribute (Class Label)	Number of Instances
Benign (legitimate traffic)	56863
DDoS_DNS	5071011
DDoS_LDAP	2179930
DDoS_MSSQL	4522492
DDoS_NetBIOS	4093279
DDoS_NTP	1202642
DDoS_SNMP	5159870
DDoS_SSDP	2610611
DDoS_SYN	1582289
DDoS_TFTP	20082580
DDoS_UDP	3134645
DDoS_UDP-Lag	366461
DDoS_WebDDoS	439

Fig 1.0 CICDDoS 2019

Data Pre-Processing: The dataset needs to be cleaned, standardized, parsed and reduced before it can be used to train and test the model. Non numeric values are transformed into numerical. Missing values were filled in and noise and outliers were removed. The datasets are joined and then transformed into a smaller dataset using Principle Component Analysis while keeping most of the information.

Principle Component Analysis: Principal Component Analysis, or PCA, is a technique that is often used to reduce dimensionality of large datasets. This is done by converting a large set of variables into a smaller one while retaining the majority of the information from the large set.

3.2..Data Classification

The model classifies data one by one with individual classifiers . These classifiers run parallel and build a different model based on the training dataset. There are several classifiers in machine learning that can be used but we will use four classifiers: Naive Bayes, SVM, KNN, and Random

Forest.

- Naıve Bayes: It is a probabilistic learning model that is used for classification in Machine Learning . It is based on the Bayes theorem. It is fast and easy to implement but it needs the predictors to be independent.

- KNN: KNN stands for k-nearest neighbours . It is a non parametric algorithm based on supervised learning technique. It stores all the available data and classifies a new data point based on their similarity.

- SVM: Support Vector Machine (SVM) is a supervised machine learning model and it uses classification and regression. It categorizes data in different classes based on the labelled training data.

- Random Forest: It is made up of several independent decision trees, which are independently trained on a random subset of data from the labeled dataset. Random Forest works well because a large number of relatively independent trees will perform better than any of the individual models.

3.3..Ensemble with Majority Voting

The ensemble consists of five different classifiers that work parallel and each classifier builds a different model based on the training dataset. Majority Voting is an ensemble machine learning algorithm that combines predictions from multiple classifiers or models . The predictions for each label are summed and the label that has the majority vote is predicted.

3.4..Implementation

The model is coded in Python language as Python provides a set of libraries such as sklearn, pandas, and numpy. Scikit-learn is one of the most useful libraries for machine learning as it provides easy access to various classification, clustering and regression algorithms such as Naive Bayes, KNN etc.

The model has 5 different parts:

1. Process
2. Train
3. Test
4. Ensemble
5. Predict

Fig 1.1

usage: script.py [-h] [-train] [-test] [-process] [-predict] [-file FILE]

optional arguments:

-h, --help : show this help message and exit

-train : run training of model

-test : test the model and print accuracy

-process : process the dataset

-ensemble : create an ensemble of the models

-predict : predict data and save prediction in file

-file FILE : file name

3.5..Process

First, all the 12 training datasets are combined to form a single dataset. Then the dataset is cleaned which is necessary to remove noise and outliers as well fix the missing values wherever required and we are left with the highest quality of information and this also increases the overall productivity.

After the datasets are cleaned PCA(Principal Component Analysis) is used to transform the dataset by reducing the dimensionality from a large set of variables(80+ columns) to a smaller one(5 columns). This is done in order to improve visualization, increase interpretability and minimize information loss. The final dataset is then saved in a local directory.

3.6..Train

The preprocessed training dataset is used to train and build five different models independently. The five models are based on classification techniques such as Naive Bayes, KNN, SVM, Fuzzy c-means, and Random Forest. These classification algorithms are provided as methods in Scikit-learn python library. The dataset is provided as an argument to these methods and then the trained models are saved in a local directory.

3.7.Test

The preprossed testing dataset is used to test the models that were trained earlier. Accuracy, ROC curve, Recall, F-Score, Sensitivity and Selectivity are printed and saved for each model independently.

3.8.Ensemble

Each of the four models have different accuracy and are suitable for different use cases. So using an ensemble model works best as it combines the best of each model and therefore has the highest accuracy which we have

confirmed in this paper as well. The models are combined using a majority voting method. Factors such as accuracy, ROC curve, Recall, F-Score are printed and saved.

3.9.Predict

The model is fed a dataset and the results are predicted by each classification model separately. The results are then combined using a majority voting method. The final prediction is then saved in a local directory.

4.Results

The performance is evaluated based on True Positive Rate (TPR), True Negative Rate (TNR), False Positive Rate (FPR), False Negative Rate (FNR), Accuracy, Recall, F-Score and Receiver Operating Characteristics (ROC) curve, Specificity and Selectivity.

Accuracy = $(TPR+(TNRTPR++TNRFPR+)FNR)$

Precision (P) = $(TPRTPR + FPR)$

Recall = $(TPR^{TPR} + FPR)$

F -Score = $\underline{2Precision}\underline{*Precision} +\underline{*} \underline{RecallRecall}$

Sensitivity = $(TPR^{TPR}+FNR)$

Specificity = $(TNRTNR+FPR)$

Models	Accuracy	Recall	F Score	ROC AUC
Naive Bayes	99.353726%	0.99353726	0.99425571	0.49896584
Random Forest	99.747371%	0.99747371	0.99622315	0.5
KNN	99.747823%	0.99747823	0.99622641	0.49999624
SVM	99.734265%	0.99734265	0.99615862	0.49992828
Ensemble	99.748571%	0.99748571	0.99623015	0.5

Fig 1.2 Performance Comparison of the models.

5.Discussions

All the four models as well as the ensemble performed really well with each one predicting the correct classification with an accuracy of over 99%. Furthermore, the ensemble model worked the best as predicted. The Naive Bayes Model is the least accurate but it is the fastest one to train and test. The SVM model took the longest amount of time to train and test but the accuracy is comparable to the other three models.


Accuracy vs. Models
99.8
99.6
99.4
99.2
Accuracy
99.353726
99.747371
99.747823
99.734265
99.748571
Naive Bayes
Random Forest
KNN
SVM
Ensemble
Models

Fig 1.3


F Score vs. Models
0.999
0.998
0.997
0.996
0.995
0.994
0.993
F Score
0.99425571
0.99622315
0.99622641
0.99615862
0.99623015
Naive Bayes
Random Forest
KNN
SVM
Ensemble
Models

Fig 1.4

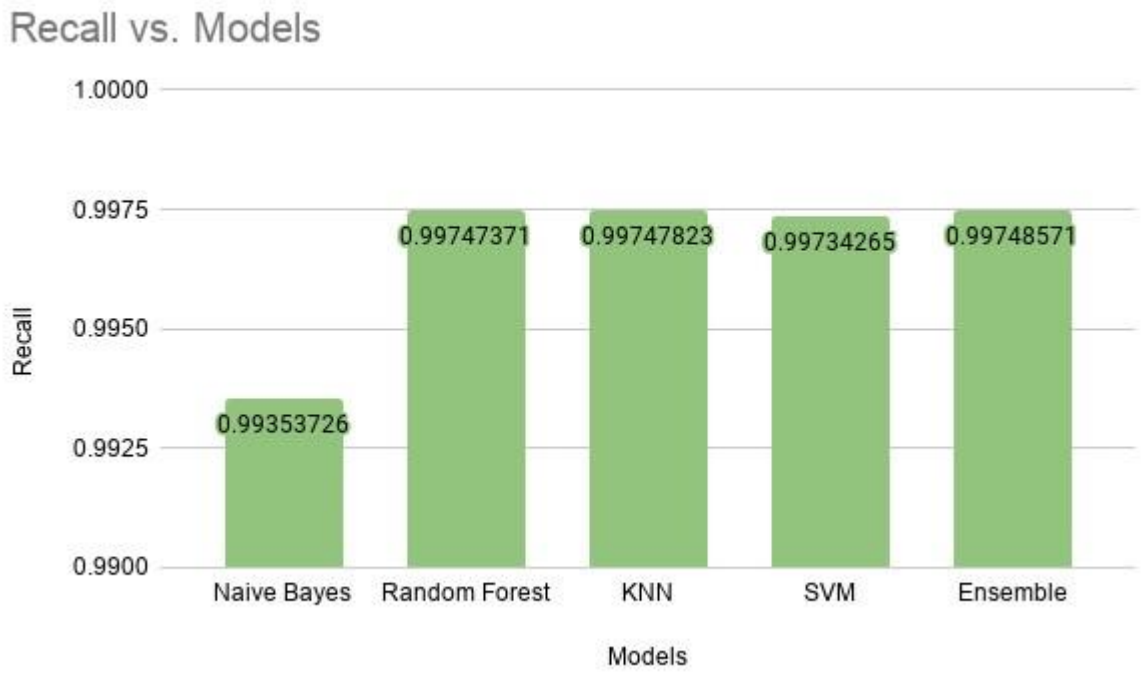

Fig 1.5

Fig. 1.6

6.Conclusion and Future Studies

In this paper to deal with the problems or issues arising for supervised and unsupervised learning methods a hybrid approach is proposed that uses four different classifiers. We firstly trained and tested separate models based on each classifier. Secondly, we used a majority voting method to create an ensemble based on all four classifiers. Two conclusions are drawn from the results of the experiment. First, the ensemble of models performed better than each individual model. Second, the SVM model might not be the best choice for DDoS

detection as it takes far more time to train compared to other three models. This is due to the large number of features in the dataset.

In the future, better and bigger dataset will be used to verify the advantages of the Ensemble model. The ensemble models can be further improved by using a combination of better algorithms.

Acknowledgement

The paper is created under the guidance of Ms. B. Sowmiya. The authors gratefully acknowledge their valuable support and suggestions.

References

1. Santos And M. Nogueira, "A Distributed Architecture For Ddos Prediction And Bot Detection" IEEE Access, Vol. 8,Pp.159756-159772,2020,Doi:10.1109/Access.2020.3020507.
2. W. Zhijun, X. Qing, W. Jingjie, Y. Meng and L. Liang, "Low-Rate DDoS Attack Detection Based on Factorization Machine in Software Defined Network" in IEEE Access, vol. 8, pp. 17404-17418, 2020, doi: 10.1109/ACCESS.2020.2967478.
3. S. Dong and M. Sarem, "DDoS Attack Detection Method Based on Improved KNN With the Degree of DDoS Attack in Software-Defined Networks" in IEEE Access, vol. 8, pp. 5039-5048, 2020, doi: 10.1109/ACCESS.2019.2963077.
4. D.Yin,L. Zhang and K. Yang, "A DDoS Attack Detection and Mitigation With Software-Defined Internet of Things Framework" in IEEE Access, vol. 6, pp. 24694-24705, 2018, doi: 10.1109/ACCESS.2018.2831284.
5. S. Alzahrani and L. Hong, "Detection of Distributed Denial of Service (DDoS) Attacks Using Artificial Intelligence on Cloud" 2018 IEEE World Congress on Services (SERVICES), San Francisco, CA, 2018, pp. 35-36, doi: 10.1109/SERVICES.2018.00031.
6. S. Das, A. M. Mahfouz, D. Venugopal and S. Shiva, "DDoS Intrusion Detection Through Machine Learning Ensemble " 2019 IEEE 19th International Conference on Software Quality Reliability andSecurityCompanion(QRSC),Sofia,Bulgaria, 2019, pp. 471-477, doi: 10.1109/QRSC.2019.00090.
7. S. Nandi, S. Phadikar and K. Majumder, "Detection of DDoS Attack and Classification Using a Hybrid Approach" 2020 Third ISEA Conference on Security and Privacy (ISEA-ISAP), Guwahati, India, 2020, pp. 41-47, doi: 10.1109/ISEA-ISAP49340.2020.234999.
8. P. Blazek, T. Gerlich and Z. Martinasek, "Scalable DDoS Mitigation System" 2019 42nd International Conference on Telecommunications and Signal Processing (TSP), Budapest, Hungary, 2019, pp. 617-620, doi: 10.1109/TSP.2019.8768869.
9. W. Sun, Y. Li and S. Guan, "An Improved Method of DDoS Attack Detection for Controller of SDN" in 2019 IEEE 2nd International Conference on Computer and Communication EngineeringTechnology(CCET),Beijing,China,2019,pp.249-253,doi: 10.1109/CCET48361.2019.8989356.
10. R. M. Thomas and D. James, "DDOS detection and denial using third party application in SDN" 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), Chennai, 2017, pp. 3892-3897, doi: 10.1109/ICECDS.2017.8390193
11. D. Erhan and E. Anarim, "Hybrid DDoS Detection Framework Using Matching Pursuit Algorithm,"inIEEEAccess,vol.8,pp.118912-118923,2020,doi: 10.1109/ACCESS.2020.3005781.
12. S. Sahu and A. Verma, "DDoS attack detection in ISP domain using machine learning," 2019 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA), Pune, India, 2019, pp. 1-4, doi: 10.1109/ICCUBEA47591.2019.9128624.
13. R. Vishwakarma and A. K. Jain, "A Honeypot with Machine Learning based Detection Framework for defending IoT based Botnet DDoS Attacks," 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2019, pp. 1019-1024, doi: 10.1109/ICOEI.2019.8862720.
14. Y. Su, X. Meng, Q. Meng and X. Han, "DDoS Attack Detection Algorithm Based on Hybrid TrafficPredictionModel"2018IEEEInternationalConferenceonSignalProcessing, CommunicationsandComputing(ICSPCC),Qingdao,2018pp.1-5,doi: 10.1109/ICSPCC.2018.8567771.
15. S. S. Priya, M. Sivaram, D. Yuvaraj and A. Jayanthiladevi, "Machine Learning based DDOS Detection," 2020 International Conference on Emerging Smart Computing and Informatics (ESCI), Pune, India, 2020, pp. 234-237, doi: 10.1109/ESCI48226.2020.9167642.
16. O. Rahman, M. A. G. Quraishi and C. Lung, "DDoS Attacks Detection and Mitigation in SDN Using Machine Learning," 2019 IEEE World Congress on Services (SERVICES), Milan, Italy, 2019, pp. 184-189, doi: 10.1109/SERVICES.2019.00051.

17. G. Kaur and P. Gupta, "Hybrid Approach for detecting DDOS Attacks in Software Defined Networks" 2019 Twelfth International Conference on Contemporary Computing (IC3), Noida, India, 2019, pp. 1-6, doi: 10.1109/IC3.2019.8844944.
18. V. Deepa, K. M. Sudar and P. Deepalakshmi, "Detection of DDoS Attack on SDN Control plane using Hybrid Machine Learning Techniques," 2018 International Conference on Smart 2 Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 2018, pp. 299-303, doi: 10.1109/ICSSIT.2018.8748836.
19. S. Wei, S. Dai, X. Wu and X. Han, "STDC: A SDN-Oriented Two-Stage DDoS Detection and Defence System Based on Clustering" 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference on Big Data Science and Engineering(TrustCom/BigDataSE), New York, NY, 2018, pp.339-347,doi: 10.1109/TrustCom/BigDataSE.2018.00059.
20. S. R, A. Pattanaik, A. Gupta and A. Kanavalli, "Early Detection and Diminution of DDoS attack instigated by compromised switches on the controller in Software Defined Networks,"2019 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics(DISCOVER),Manipal,India,2019,pp.1-5,doi: 10.1109/DISCOVER47552.2019.9007925