A PREDICTION MODEL FOR FLOODED PACKETS IN SNMP NETWORKS

Kiruthika Devi B.S[1], Aravindhan K[2], P Srikanth Kini[3] ,G Adarsh Reddy[4], Subbulakshmi T[5]

School of Computing Science and Engineering, VIT University Chennai Campus, India

E-mail: {kiruthikadevi.bs2015@vit.ac.in[1], aravindhan.k2015@vit.ac.in[2], psrikanth.kini2015@vit.ac.in[3], gadarsh.reddy2015@vit.ac.in[4],research.subbulakshmi@gmail.com[5]}

***Abstract*—This paper implements several machine learning algorithms to determine a model of maximum accuracy for predicting flooded packets in SNMP networks. Flood attack is the method of forwarding packets from the source to destination using the outgoing path other than the source. Denial of service attack is a flooding attack that can bring down the network. So, to determine the flooding of packets we implemented algorithms like SVM (Support Vector Machines), Logistic Regression, Decision-Tree Classifier, K-Neighbors Classifier, XGB Classifier and other algorithms and have compiled the results for each algorithm in terms of accuracy.**

***Keywords*—Flooding, SNMP packets, DOS, SVM, Logistic Regression, Decision-Tree Classifier, K-Neighbors Classifier, XBG Classifier**

I. INTRODUCTION

THE dataset used for the purpose of this project is that of SNMP networks, the cloud security dataset which is built using open source cloud. It is free software and it can be redistributed and altered based on the license used by GNU, either version 3, or any later version. DDoS Attack datasets are generated using virtual instances running in a cloud security testbed. OpenStack private cloud is deployed in the cloud security testbed along with virtual instances running on them. The virtual instances are acting as zombies from different public and private networks of Open Stack private cloud. The attack generation is done using virtual instances as zombies and the data collection is performed at the victim with network monitoring software. The DDoS flood attacks that are considered in the experiment are ICMP flood, TCPSYN flood, TCPSYN-ACK flood, UDP flood and Land Flood. The attack duration is half an hour and the dump files collected are in the excel format. The Simple Network Management Protocol (SNMP) is the most widely used in the network for communication. It is used for notifying the events, depict the statistics of the networked devices and tune the settings and provides several other functionalities. There are many methods to detect flooding, but this paper gives an overview of several algorithms implemented to achieve best results.

II. RELATED WORK

SNMP protocol is used for communicating between the systems under the same networked area. The basic aspects that must be considered when using SNMP data is 1) choosing the correct variables for DDoS detection 2) how to fast the detection method can work 3) choice of the algorithm. Testbed is used for detection the DDoS attack using classification methods [1]. Yoo et al. [2] employed tcpInErrs, udpNoPorts, and icmpOutEchoReps for DDoS attack detection. Those variables are appropriate for detecting recent cloud attacks. Jaehak Yu et al. [3] built a testbed to run real time experiments. The topology consists of 1 target, 2 attacking agents, 1 handler machine, and 1 dataset collection system. The testbed is associated with the campus network where the normal traffic is initiated with the target and other host external to the test network. The Stacheldraht is the most popular tool that generates DDoS attack. It has the master and the agent nodes. The system can scan the vulnerabilities of the target and attack is initiated from multiple agents [4]. The attacks considered are TCP-SYN, UDP & ICMP kind of DDoS.

III. SNMP MIB VARIABLES

The SNMP variables considered are IP, TCP, UDP and ICMP variables. The variables that are accounted in IP protocol are ipForwarding, ipDefaultTTL, ipInReceives, ipInHdrErrors, ipInAddrErrors, ipForwDatagrams, ipInUnknownProtos, ipInDiscards, ipInDelivers, ipOutRequests, ipOutDiscards, ipOutNoRoutes, ipReasmTimeout, ip_ReasmReqds, ipReasmOKs, ip_ReasmFails, ip_FragOKs, ip_FragFails, ip_FragCreates and ip_RoutingDiscards. The variables that are accounted in TCP protocol are tcp_RtoAlgorithm, tcp_RtoMin, tcp_RtoMax, tcp_MaxConn, tcp_ActiveOpens, tcp_PassiveOpens, tcp_AttemptFails, tcp_EstabResets, tcp_CurrEstab, tcp_InSegs, tcp_OutSegs, tcp_RetransSegs, tcp_InErrs and tcp_OutRsts. The variables accounted in UDP are udp_InDatagrams, udp_NoPorts, udp_InErrors and udp_OutDatagrams. The variables in ICMP protocol are icmp_InMsgs, icmp_InErrors, icmp_inDestUnreachs, icmp_InTimeExcds, icmp_InParmProbs, icmp_InSrcQuenchs, icmp_InRedirects, icmp_InEchos, icmp_InEchoReps,

icmp_InTimestamps, icmp_InTimestampReps, icmp_InAddrMasks, icmp_InAddrMaskReps, icmp_OutMsgs, icmp_OutErrors, icmp_OutDestUnreachs, icmp_OutTimeExcds, icmp_OutParmProbs, icmp_OutSrcQuenchs, icmp_OutRedirects, icmp_OutEchos, icmp_OutEchoReps, icmp_OutTimestamps, icmp_OutTimestampReps, icmp_OutAddrMasks and icmp_OutAddrMaskReps.

Three important formulas used for providing the performance of the system are the attack detection, false positive and false negative rate (FNR) as follows: [5]

$$\text{Attack Detection Rate} = \frac{\sum_{i=1}^{n} T_i}{\sum_{i=1}^{n} I_i}$$

$$\text{False Positive Rate} = \frac{\sum_{i=1}^{n} P_i}{\sum_{i=1}^{n} N_i}$$

$$\text{False Negative Rate} = \frac{\sum_{i=1}^{n} F_i}{\sum_{i=1}^{n} I_i}$$

.

IV. PROPOSED SYSTEM

The dataset used for the project contains several attributes which have been elaborated in the next section.

A. Correlation between the attributes

The relationship between the attributes was studied by plotting a heat-map or correlation map which essentially showed how attributes with similar values are clustered [7].

B. Pre-processing of the data

1) Dropping of columns

There were several data columns with similar values which in terms of analysis have no significance. Redundant columns with repeating insignificant values were eliminated to create a reduced data set. Out of the 66 columns we had, 36 were dropped. So, a total of 30 columns were left to work with. The class labels were replaced with integers ranging from 0-9 with the different types of floods receiving the latter set of values.

2) Making frames

Since it is not possible to determine if a single packet is flooded, it was necessary to group packets into frames for analysis. We created a function TimeDiff() using Python which takes the data column and splits the attribute in terms of hours, minutes and seconds to find the difference between consecutive packets sent.[8] We grouped 5 packets at a time and calculated the difference in their arrival time. This function precisely calculates time difference between consecutive packets and creates a separate data set with the updated values. The packet values are considered as a single data frame or point. This is used to reduce the overall dataset and find the time difference between packets[9].

V. ARCHITECTURE

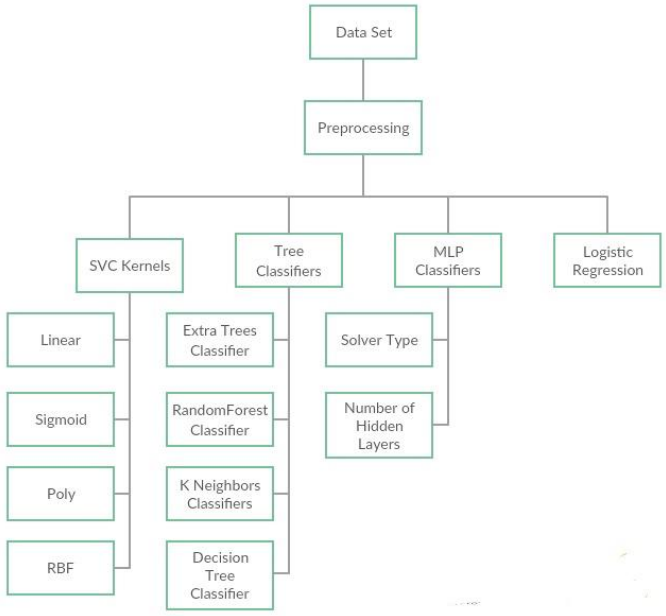

Fig. 1: Architecture Diagram

A. Training and test sets

The reduced dataset was is divided into training and test data sets. 70% of the dataset was used for training and 30% was used for testing which were subjected to several classification algorithms and tested for accuracy. Before classification we used the Standardscaler () function in the sklearn preprocessing library, to scale the dataset across the attributes[10].

B. CLASSIFIERS USED

Under the sklearn library, several classifier algorithms used are compiled below:

1) SVC

The objective of a Linear SVC (Support Vector Classifier) provides the most suited fit using a hyperplane by dividing the given data. With the hyperplane, few features can be included within the classifier for class prediction [11]. We used the SVC classifier on the dataset with the linear, sigmoid, poly and RBF kernels.

2) MLPClassifier

Multi-layer Perceptron (MLP) is a supervised method of learning the function $f(\cdot) : R^m \rightarrow R^o$ with a dataset, where 'm' is the no.of input dimension and 'o' is the no.of output dimension. Given a set of features $X = x_1, x_2, \ldots, x_m$ and a target y, it can learn a non-linear function approximate for either classification or regression. The MLP has the input, output and the hidden layer and a feedback system to correct the errors. Figure 1 shows a one hidden layer MLP with scalar output. [12].

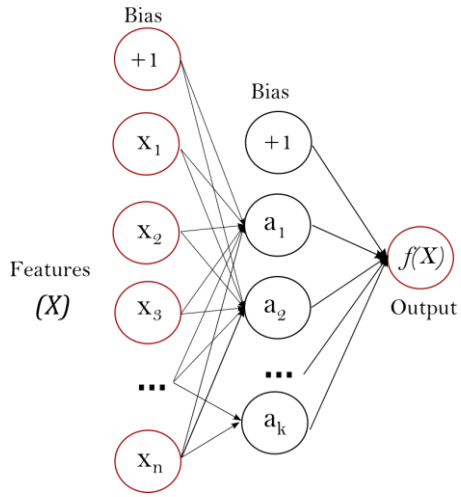

Fig. 2: *One hidden layer MLP*

3) Logistic Regression

This class implements regularized logistic regression using the 'liblinear' library and 'newton-cg'. It can be utilized for the high density and low density data. It uses C array or CSR matrix with 64-bit floating value for optimum output.

4) DecisionTreeClassifier

Decision Trees (DTs) are also classified as the supervised method that can provide prediction of the output with decision rule form the input features. Based on the rules, the input data can be classified [13].

5) KNeighborsClassifier

NearestNeighbors is one of the unsupervised method. **BallTree**, **KDTree**, and a brute-force are the basic three algorithm. The search can be carried out by the keyword and it has the default method also [14]. The keywords are auto, ball tree, kd tree and brute.

6) Random Forest Classifier

The **sklearn.ensemble** module includes two averaging algorithms based on randomized decision trees: the RandomForest algorithm and the Extra-Trees method. Both algorithms are used for classification. It can include varied level of classifier by incorporating the random factor into the model. The prediction of the ensemble is given as the averaged prediction of the individual classifiers. [15]

In random forests, each tree is constructed from the sample arrived with the training data. The split that is not the best split of the features are not considered. Only the best split is chosen from the sub pool of the features. The bias will increase with the random factor. The variance will decrease with the average so that it compensated for the increased bias [16].

7) XGBClassifier

XGBoost gives a wrap class to permit the model as the classifier or regression within the scikit-learn. **XGBClassifier** can process the test and training dataset. XGBoost uses the probability as the default one. Since, it is binary classifier, the probability of the input value can be either in any one class. The binary value can be rounded to either zero or one [17].

8) ExtraTreesClassifier

The class implement an estimator that fit a no.of random decision trees on various sub-samples of the dataset and use averaging to improvise the accuracy and control over-fitting [18].

9) GradientBoostingClassifier

Gradient Boosting for classification. GB builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions. In each stage n_classes_ regression trees are fit on the negative gradient of the binomial or multinomial deviance loss function. Binary classification is a special case where only a single regression tree is induced. [19]

10) DATA USED

The SNMP parameters that are used for data collection are as follows:
SNMP Parameters:
These parameters include all the protocols like IP, ICMP, UDP and TCP OID's where the following metrics can be obtained:[20]

VI. Results and Discussion

The below comparison in Fig 3 is between the different types of SVC Kernels of which best accuracy was given by the Linear SVC Kernel

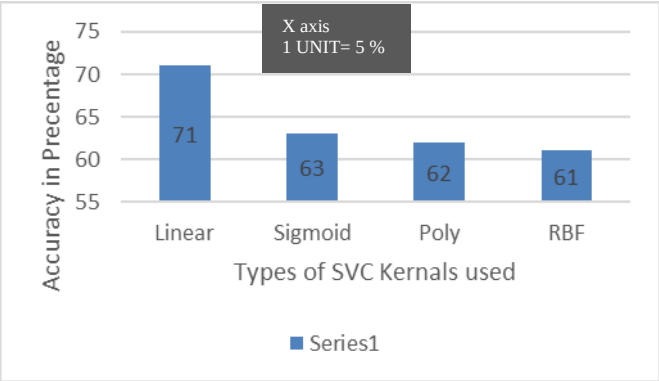

Fig. 3: Comparison Between Different SVC Kernels

The below comparison in Fig 4 is between the different types of Tree Classifier Algorithms of which best accuracy was given by the Extra Trees Class. The below comparison in Fig 5 is between MLP Classifier with different parameters of which best accuracy was given by the LBFGS Solver with 50 Hidden Layers

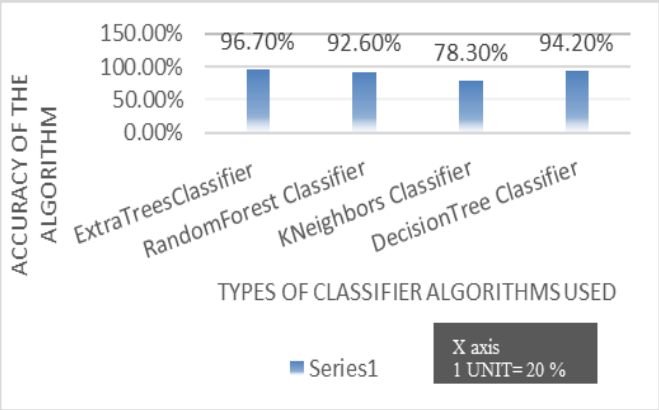

Fig. 4: Comparison Between Different Tree Classifiers

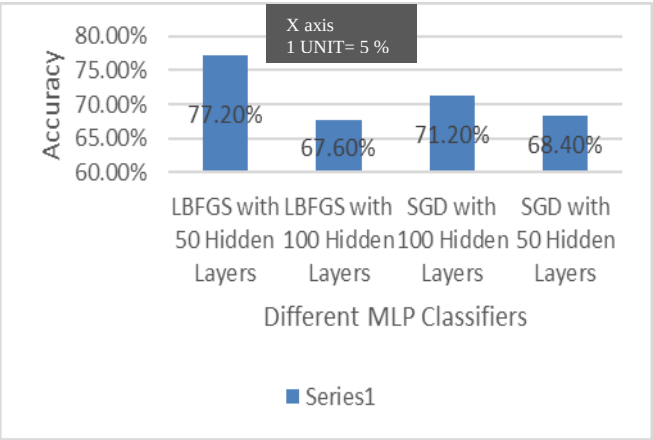

Fig. 5: Comparison Between MLP Classifiers with Different Number of hidden layers

The objective of the experiment conducted was to measure the accuracy difference between various classifiers for the dataset. Hence the focus was not in improve the accuracy but to compare the various classifier algorithms. In order to compare the accuracy of the classifiers, the accuracies were measured for individual classifiers. The measured accuracies for individual classifiers are displayed in the Fig 6.

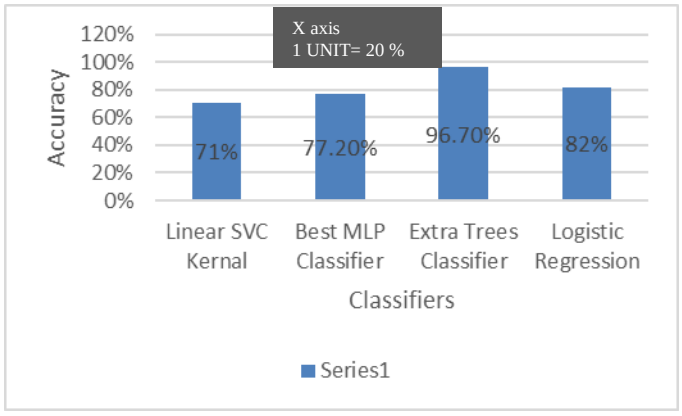

Fig. 6: Comparison Between Best Algorithms in their classifications

REFERENCES

[1] P. Suresh Kumar,S, Pranavi, "Performance analysis of machine learning algorithms on diabetes dataset using big data analytics"
[2] M. Kim, H. Kang, S. Hong, S. Chung, J.W. Hong, A flow -based method for abnormal network traffic detection, in: Proceedings of NOMS 2004, Seoul, Korea, April 2004 , pp. 559–612
[3] Cui-Mei Bao,"Intrusion detection based on one-class svm and snmp mib data"
[4] https://www.python-course.eu/machine_learning.php
[5] https://www.datacamp.com/community/tutorials/python-numpy-tutorial
[6] Jianwu Zhang," Design and implementation of test IP network intelligent monitoring system based on SNMP"
[7] https://www.datacamp.com/community/tutorials/pandas-tutorial-dataframe-python
[8] Marina K. Thottan, George K. Swanson, Michael Cantone,"SEQUIN: An SNMP-based MPLS network monitoring system"
[9] https://www.analyticsvidhya.com/blog/2016/01/complete-tutorial-learn-data-science-python-scratch-2/
[10] Lei Li, Yabin Wu, Yihang Ou,"Research on machine learning algorithms and feature extraction for time series"
[11] Heng Zhou, Guangpu Shan, Song Liu," A Monitoring Method of Network Power Consumption Information Based on SNMP"
[12] https://www.geeksforgeeks.org/numpy-in-python-s

et-1-introduction/
[13] Achmad Affandi, Dhany Riyanto , Istas Pratomo ,“Design and implementation fast response system monitoring server using Simple Network Management Protocol”
[4] https://www.hackerearth.com/practice/machine-learning/data-manipulation-visualisation-r-python/tutorial-data-manipulation-numpy-pandas-python/tutorial/
[15] L.P. Gaspary, R.N. Sanchez, D.W. Antunes,” A SNMP-based platform for distributed stateful intrusion detection in enterprise networks”
[16] Yoo, D.-S., Oh, C.-S.: Traffic Gathering and Analysis Algorithm for Attack Detection. In: KoCon 2004 Spring Integrated conference, vol. 4, pp. 33–43 (2004)
[17] Kittikhun Thongkanchorn, Sudsanguan Ngamsuriyaroj, Vasaka Visoottiviseth, “Evaluation studies of three intrusion detection systems under various attacks and rule sets”
[18] Hao Wei, Micheal Baechler, Fouad Slimane, “Evaluation of SVM, MLP and GMM Classifiers for Layout Analysis of Historical Documents”
[19] Xuan Zhou, Wenjun Wu, Yong Han,” Modeling multiple subskills by extending knowledge tracing model using logistic regression”
[20] Chris Yakopcic, Tarek M. Taha, “Memristor crossbar based implementation of a multilayer perceptron”