2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS) | 978-1-7281-8524-8/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICIIS51140.2020.9342634

An Approach to Reduce Uncertainty Problem in Network Intrusion Detection Systems

Gargi Kadam
Department of Information Technology
Sardar Patel Institute of Technology
Mumbai, India
gargi.kadam@spit.ac.in

Sahil Parekh
Department of Information Technology
Sardar Patel Institute of Technology
Mumbai, India
sahil.parekh@spit.ac.in

Priyanka Agnihotri
Department of Information Technology
Sardar Patel Institute of Technology
Mumbai, India
priyanka.agnihotri@spit.ac.in

Dayanand Ambawade
Department of Electronics and Telecommunication
Sardar Patel Institute of Technology
Mumbai, India
dd_ambawade@spit.ac.in

Prasenjit Bhavathankar
Department of Information Technology
Sardar Patel Institute of Technology
Mumbai, India
p_bhavathankar@spit.ac.in

***Abstract*—In the current scenario pertaining to cyberattacks, Denial of Service attacks are the most common type. Denial of Service (DoS) has now become an attack category that has different types of attacks such as Back, Neptune, Smurf, Teardrop, etc. As common as these attacks are, they are one of the most troublesome to deal with and have become an annoyance in the industry. Along with those, attacks like User-to-Root (U2R), Remote-to-Local (R2L) and Probe are used to gain access to the system and hence form the cycle of an attack. A network intrusion detection system is proposed which is tailored to detect these attacks. The main objective is to classify the aforementioned types of attacks with minimum uncertainty and reduce the number of false positives for more reliable detection. With data mining coupled with machine learning and deep learning algorithms, a feature selection and a classification model is built by primarily training it on the KDDCup99 dataset and the ISTS Dataset, then tweaking the models by testing it on real-time data gathered from tcpdump. Real-time data collected using the ISTS dataset is firstly labelled using unsupervised machine learning methods and also by matching the data with the KDDCup99 dataset records. A model with the most optimum algorithms used for feature selection and classification procedure is developed. Also, different algorithms used on various parameters are compared.**

***Index Terms*—Intrusion Detection System, Network Intrusion Detection System, Denial of Service, Uncertainty of Intrusion Detection Systems, Probe, User-to-Root, Remote-to-Local**

I. Introduction

With the increase in the digitization of businesses, the risk associated with the confidentiality of data has increased. As more and more companies take their services online, they expose themselves to various attacks. The demand for an intrusion detection system to reduce the uncertainty problem is on the rise. The uncertainty of intrusion detection systems refers to the probability that a malicious packet has been missed by the IDS due to reasons like, heavy traffic, inaccurate classification of packets, etc.

A method to tackle this problem and improve the reliability of Network Intrusion Detection Systems with respect to Denial of Service attacks, User-to-Root, Probe and Remote-to-Local attacks is proposed. Having the potential to bring down an entire system within a short period of time and being easy to launch has made DoS attacks very common, whereas U2R, R2L and Probe attacks are not as common as DoS. Hence, we restrict our scope to various types of DoS attacks.

The proposed method makes use of machine learning and deep learning algorithms to train a model on the KDDCup99 dataset and publicly available PCAP (Packet Capture) files of the ISTS dataset. This model is later tested on a combination of 10% of the KDDCup99 dataset, a sample of PCAP files of ISTS dataset and real-time traffic collected using tools like tcpdump by simulating different types of DoS attacks, using hping3, namely, Ping of Death (PoD), Land, Smurf, Teardrop, Back and Neptune.

The proposed system is divided into two parts. An overview of the two is given as follows:

A. Feature Selection

A genetic algorithm that uses a Support Vector Machine (SVM) classifier for DoS attack and Random Forest (RF) classifier for U2R, R2L and Probe attack is implemented. There are five stages to this algorithm which run for a particular number of epochs:

- Population evaluation and calculation of fitness score
- Selection
- Crossover
- Mutation
- Comparison and merging of current and previous population

B. Collection of Real-time data

After training the model on the KDDCup99 dataset and publicly available PCAP files of the ISTS dataset, we have tested our model on a combination of 10% of the KDDCup99 dataset, a sample of PCAP files of the ISTS dataset, and real-time trafc. We use both supervised and unsupervised learning

methods for verification. The data collected will be first run through supervised learning algorithms and then through unsupervised learning algorithms and the outputs will be compared. Since some of the testing data will be labelled by comparing with the dataset, we use these different algorithms to show the difference in accuracies and select the best algorithm for classification. We proceed with the collection of data as follows:

- Sniff packets using tcpdump
- Use Bro IDS to convert PCAP files (from tcpdump) to list files
- Drop the first 6 columns to convert it into KDDCup99 format
- Feed it into the classification algorithm

II. Related Work

This field, being currently in demand, has enormous scope and significant work has already been done in it. We studied published papers in the field of security as well as those suggesting ways to improve the performance of intrusion detection systems, during our research. Papers that analysed DoS attacks and their characteristics as well as the dataset we used were also referred to. The following are some notable inferences we made.

The drawbacks of SVM with big data such as long training and testing times, high error and low true positive rates are identified as a base problem [1]. However, SVM is used coupled with Genetic Algorithm (GA) to overcome the drawbacks and obtain an optimal feature subset, since SVM performs better with small samples, non-linearity and high dimensionality. The proposed method involves the improvement of the training speed of SVM using GA population search strategy.

Major phases of the proposed model are selection of dataset, pre-processing, classification and result evaluation since they significantly affect the performance in varied ways [2]. The focus of this work is to analyse and compare different classifiers in intrusion detection. A comparison is made between SVM, RF and Extreme Learning Machine (ELM) by using different metrics such as accuracy, precision and recall. The paper concludes that ELM performs better in terms of accuracy, precision and recall.

A system focuses on specific attacks, namely DoS and R2L [3]. The most important features for the detection and analysis of these attacks and that due to the appropriate selection of parameters, the accuracy and the speed of detection are observably improved. A new algorithm for the implementation of SVM in IDS is proposed by combining feature selection and parameter optimization.

In [4], the NSL-KDD dataset is segregated into three subsets in the Linux terminal by using a script. Genetic algorithm is used for selection of strong and most influential features. For classification, the classifier is constructed using the WEKA environment. A multilayer perceptron is used because most exploited vulnerabilities are in the protocol. Hence, it is focused on making a protocol-based Genetic Algorithm and Multilayer Perceptron Network Intrusion Detection System. It is found that the proposed system gives a slight increase in the accuracy of the IDS.

A comprehensive evaluation is done between various neural networks and other machine learning algorithms in this paper [5]. Hyperparameters are selected on basis of the network topologies and parameters for DNNs (Deep Neural Network). They use different variants and versions of the KDDCup99 dataset. This paper combines both, an NIDS (Network Intrusion Detection System) and an HIDS (Host-Based Intrusion Detection Systems) by proposing a deep learning model for the detection of cyberattacks. The system has the goal of systematically alerting the network admin and in the context of the NIDS, DNNs with tweaked hyperparameters are run for 1000 epochs with a learning rate in the range of 0.01 and 0.5. Hence, more emphasis is given on accuracy and detection and not on performance and classification time.

A hybrid double-layer network intrusion detection model is proposed in [6] to achieve high performance and accuracy on the KDD99 dataset. The model is divided into three parts data process module, detection and classification module and a notice module. A Gradient Boosting (GDBT) classifier is used to classify the traffic into DoS or Non-DoS. The results are accepted by KNN (K-Nearest Neighbours), Stacking Ensemble Classifier (SEC) and SVM classifier in the following layer where KNN further classifies the DoS attack into subtypes such as Smurf, Neptune, PoD, Teardrop, Back.

The system [7] focuses on dimensionality reduction to optimize the process of intrusion detection. Dimensionality reduction is optimized by restraining the cluster size and sub-medoids usability. Accuracy, specificity, and sensitivity are the evaluation metrics considered in the paper. From the experiments, we can infer that using the right feature subsets and using distance to sub-medoid to replace the use of sub-centroid distance leads to performance enhancement.

According to the study [8], anomaly detection is yet to gain momentum as it is still not the ideal preferred technology because of some inherent flaws in the KDDCUP99 dataset. A huge number of redundant records is an important deficiency in the KDDCUP99 dataset. 78% and 75% of the records in the train and test set are duplicates. This causes the learning algorithms to be biased towards more frequent records and prevents them from learning non-frequent records which is harmful and leads to biased evaluation results of the test set.

III. Methodology

A. Feature Selection Algorithm

For feature selection, we use Genetic Algorithm [1,4]. Initially, all categorical features like protocol-type, service, flag, attack-type of the KDDCup99 and ISTS dataset are label encoded. Label encoding refers to the process of converting all labels (nominal values of the feature) into numeric form i.e. in a machine-readable format.

An initial population of 20 chromosomes is considered. A chromosome is a list of bits of size 42, the number of features. This list is randomly filled with 1s and 0s. In a chromosome, the 1s represent the inclusion of the indices of the columns

Part No: CFP2058A-ART; ISBN No:978-1-7281-8524-8/20

corresponding to them and 0s represent the indices of the columns to be dropped. Thus, we make sure that the first 6 bits and the last bit are always 1 because these correspond to the most important features, which are, duration, protocol, service, flag, source-bytes and destination-bytes. The last column states the type of attack.

Further, we use the SVM classifier[1,2,3] classifier and RF classifier, given their advantages. The initial population is passed through a k-fold cross-validation process using 3 folds.

An average of metrics such as True Positive Rate (TPR), Error and False Positive Rate (FPR) and feed it to the fitness function. The fitness function is defined as

0.6*TPR - 0.4*Error - 0.5*FPR - 0.005*No of features

A list of fitness values of size equal to the size of the initial population is obtained. The initial population, now termed as the current population is sent through a process of selection, crossover and mutation.

In selection, 60% of the total chromosomes are selected on basis of their fitness values. Higher the fitness value, the greater the chance of selection.

For crossover, a two-point crossover function [1] is used with a crossover probability as defined in the following equation. The crossover probability function determines the crossover probability based on the current epoch and it decreases as the distance between the current epoch and the final epoch decreases. A random number between 0 and 1 is taken for every chromosome pair and if this random number is greater than or equal to the crossover probability, the pair undergoes crossover. The crossover probability function is defined as follows:

$$\frac{\frac{(NumEpochs - currentEpoch)*0.9}{numEpochs} + \frac{0.9*minFitnessScore}{maxFitnessScore}}{2} \quad (1)$$

Next, we mutate the population. For mutation, the resulting population is sent through a mutation function. One point mutation [1] is used with a mutation probability defined in the following equation. The mutation probability function determines the mutation probability based on the current epoch and it decreases as the distance between the current epoch and the final epoch decreases. A random number between 0 and 1 is taken for every chromosome and if this random number is greater than or equal to the mutation probability, the chromosome undergoes mutation. The mutation probability function is as follows:

$$\frac{(NumEpochs - currentEpoch)*0.1 + currentEpoch*0.001}{NumEpochs} \quad (2)$$

Mutation is only allowed between columns 6 and 40. After mutation, a new population is obtained. We then calculate the fitness values of the new population and the current and the new populations are merged and the top 62.5% chromosomes are then selected to be the current population for the next epoch. Here, use 62.5% to restore the initial size of the population. They are sent through the processes of k-fold cross-validation, selection, crossover and mutation again. This whole process is repeated for a specified number of epochs. In this system, the algorithm is executed for 20 epochs.

After the completion of the final epoch, the best chromosome is selected from the population and the columns corresponding to 1s in the list are the final, selected features.

B. Classification Algorithm

Three types of classification algorithms to present a comparative study of their results. The algorithms used are K-Nearest-Neighbours, RF classification and Articial Neural Network.

Before sending the data through the classification algorithms, the attack-type label is label encoded. For the KNN algorithm, we set the number of neighbours to 5 representing the 4 types of attacks and packets of normal type and for the RF classification algorithm, the number of trees generated is 100. We use a neural network with three hidden layers along with the input and output layers. The size of the input layer is equal to the number of features selected whereas the size of the hidden layers is twice the number of features selected. The size of the output layer is equal to the number of attack types, including 'normal'. The input and hidden layers use a ReLU activation function and the output layer uses a Softmax activation function. A dropout layer is also added before the output layer with dropout equal to 0.3 to prevent overfitting. The model uses a Stochastic Gradient Descent (SGD) optimizer and a categorical cross-entropy loss function.

Following are the steps we followed for the real-time data collected for DoS type attacks:

- A dataset is created by taking out rows which are either of DoS type or normal type from the KDDCup99 10% dataset. It is fed through the feature selection and classification model
- The first batch of real time data is collected where different types of DoS attacks are simulated and dumped into PCAP files
- These PCAP files are converted into the KDDCup99 format by using Bro IDS. The first six columns of the output obtained by passing the PCAP files through Bro IDS are dropped
- The resulting file is then converted into a comma-separated files
- The records in this file are labelled by mapping them to the KDDCup99 dataset into either of the DoS types or normal type. We divide the KDDCup99 dataset into 3 types based on the protocol and set a threshold value for each protocol-type. Every row of the unlabelled dataset is matched to the corresponding subset of the KDDCup99 dataset. If the number of matching values of features exceeds the threshold value then the attack-type of the unlabelled dataset record is set to the corresponding attack-type of the KDDCup99 dataset. Else, it is set to 'normal'.

Part No: CFP2058A-ART; ISBN No:978-1-7281-8524-8/20

```
424119/424119 [==============================] - 28s 67us/step - loss: nan - accuracy: 0.0042 - val_loss: nan - val_accuracy: 0.0041
Epoch 96/100
424119/424119 [==============================] - 28s 65us/step - loss: nan - accuracy: 0.0042 - val_loss: nan - val_accuracy: 0.0041
Epoch 97/100
424119/424119 [==============================] - 28s 66us/step - loss: nan - accuracy: 0.0042 - val_loss: nan - val_accuracy: 0.0041
Epoch 98/100
424119/424119 [==============================] - 28s 65us/step - loss: nan - accuracy: 0.0042 - val_loss: nan - val_accuracy: 0.0041
Epoch 99/100
424119/424119 [==============================] - 28s 67us/step - loss: nan - accuracy: 0.0042 - val_loss: nan - val_accuracy: 0.0041
Epoch 100/100
424119/424119 [==============================] - 29s 67us/step - loss: nan - accuracy: 0.0042 - val_loss: nan - val_accuracy: 0.0041
```

Fig. 1. Output using ANN

- The model is also trained on this first batch of real time data by using various supervised and unsupervised classification algorithms
- After obtaining a satisfactory result of the performance metrics, new unlabelled data is passed through the classification model and the output is recorded
- Similarly, for the U2R, R2L and Probe types of attacks obtained from the ISTS dataset, they are first converted to the KDDCup99 format using Bro IDS and then are passed through the labelling algorithm. The labelling algorithm compares the columns of the ISTS dataset comma-separated files with the already labelled KDDCup99 dataset which also contains data for Probe, U2R and R2L.
- If the matching columns exceed a threshold criterion, then the attack-type of the ISTS dataset record is set to the corresponding attack-type of the KDDCup99 dataset. If the threshold criteria are not met for a particular row, it is set to normal

IV. Results and Discussion

After running the feature selection algorithm for 50 epochs and with an initial population of 20, we get a different feature subset for all 4 types of attacks. A union of these features subsets is taken and is used in the classification algorithm where the whole KDDCup99 dataset (with records of DoS attack type and normal type), a portion of the ISTS PCAP file data (with records of Probe, U2R, R2L, a small amount of DoS attack type and normal type) and 50% of the labelled real-time data is used as the training set and the remaining 50% of the labelled real-time data is used as the test set.

These training and test sets are fed to different algorithms and their accuracies are compared.

The output of the ANN (Artificial Neural Network) algorithm is shown in fig.1 The KNN algorithm uses a k value of 5, refer fig.3

The RF algorithm uses 100 estimators, refer fig.2

The performance of algorithms that we used to classify the real-time data collected after simulating attacks is summarised in table II. It is observed that using Random Forest Algorithm gives the highest accuracy.

The selected features of KDDCup99 dataset for the following types of attacks are:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	462
1	0.72	0.78	0.75	440
3	1.00	0.46	0.63	13
4	0.11	0.09	0.10	80
5	0.80	0.70	0.75	391
6	1.00	0.83	0.91	6
7	0.65	0.80	0.72	173
8	0.17	0.10	0.12	183
9	1.00	1.00	1.00	21451
10	0.68	0.46	0.55	89
11	0.99	0.99	0.99	25183
12	0.00	0.00	0.00	1
13	1.00	0.50	0.67	2
14	1.00	1.00	1.00	47
15	1.00	0.94	0.97	238
16	0.71	0.81	0.76	364
17	0.99	0.98	0.98	327
18	1.00	1.00	1.00	56089
20	1.00	1.00	1.00	182
21	0.92	0.90	0.91	239
22	0.17	0.09	0.11	70
accuracy			0.99	106030
macro avg	0.76	0.69	0.71	106030
weighted avg	0.99	0.99	0.99	106030

Accuracy score : 0.991681599547298

Fig. 2. Output using Random Forest

	precision	recall	f1-score	support
0	0.99	1.00	1.00	434
1	0.47	0.66	0.55	414
2	0.00	0.00	0.00	5
3	0.78	0.82	0.80	17
4	0.19	0.05	0.08	91
5	0.89	0.70	0.78	361
6	1.00	1.00	1.00	4
7	0.50	0.53	0.51	156
8	0.12	0.05	0.07	173
9	1.00	1.00	1.00	21564
10	0.72	0.47	0.57	102
11	0.98	0.99	0.98	25026
12	1.00	1.00	1.00	1
14	0.88	0.96	0.91	45
15	1.00	0.93	0.96	234
16	0.48	0.43	0.45	374
17	0.99	0.98	0.99	312
18	1.00	1.00	1.00	56249
20	0.96	0.99	0.98	164
21	0.92	0.91	0.91	215
22	0.58	0.08	0.14	89
accuracy			0.99	106030
macro avg	0.74	0.69	0.70	106030
weighted avg	0.99	0.99	0.99	106030

Accuracy score : 0.988144864660945

Fig. 3. Output using KNN

DoS: duration, protocol-type, service, flag, src-bytes, dst-bytes, land, num-file-creations, serror-rate, rerror-rate, dst-host-same-srv-rate, dst-host-diff-srv-rate, dst-host-same-src-port-rate, dst-host-srv-serror-rate, attack-type

U2R: duration, protocol-type, service, flag, src-bytes, dst-bytes, urgent, hot, logged-in, num-outbound-cmds, is-host-login, count, srv-count, serror-rate, srv-serror-rate, rerror-rate, dst-host-count, dst-host-same-srv-rate, dst-host-srv-diff-host-rate, dst-host-srv-serror-rate, dst-host-srv-rerror-rate, attack-type

R2L: duration, protocol-type, service, flag, src-bytes, dst-

Part No: CFP2058A-ART; ISBN No:978-1-7281-8524-8/20

TABLE I
MODEL PARAMETERS

Parameter	*Value*
Feature Selection Algorithm	
Initial Population Size	20
No. of Epochs	10
Crossover Type	Two Point
Crossover Probability Range	0.4 - 0.9
Mutation Type	Single Point
Mutation Probability Range	0.0001 - 0.1
Classification Algorithm (ANN)	
No. of Epochs	100
No. of Hidden Layers	4
Classification Algorithm (KNN)	
Neighbours	5
Classification Algorithm (Random Forest)	
Estimators	100

TABLE II
PERFORMANCE ANALYSIS OF DIFFERENT CLASSIFICATION ALGORITHMS

Algorithm	*Accuracy*	*Precision*	*Recall*	*F-Score*
KNN	98.81	0.99	0.99	0.99
Random Forest	99.17	0.99	0.99	0.99
ANN	42%	-	-	-

bytes, urgent, hot, logged-in, num-file-creations, num-shells, num-access-files, is-host-login, count, srv-count, serror-rate, same-srv-rate, diff-srv-rate, dst-host-same-srv-rate, dst-host-same-src-port-rate, dst-host-srv-diff-host-rate, dst-host-serror-rate, attack-type

Probe: duration, protocol-type, service, flag src-bytes, dst-bytes, hot, num-compromised, root-shell, su-attempted, num-access-files, count, srv-count, serror-rate, srv-serror-rate, same-srv-rate, diff-srv-rate, dst-host-same-src-port-rate, dst-host-srv-diff-host-rate, dst-host-srv-serror-rate, attack-type.

The following features represent the union of all of the above feature subsets that are finally passed on to the classifier:

duration, protocol-type, service, flag, src-bytes, dst-bytes, urgent, hot, logged-in, num-outbound-cmds, is-host-login, count, srv-count, serror-rate, srv-serror-rate, rerror-rate, dst-host-count, dst-host-same-srv-rate, dst-host-srv-diff-host-rate, dst-host-srv-serror-rate, dst-host-srv-rerror-rate, attack-type, num-file-creations, num-shells, num-access-files, same-srv-rate, diff-srv-rate, dst-host-same-src-port-rate, dst-host-serror-rate, num-compromised, root-shell, su-attempted, land, dst-host-diff-srv-rate.

V. CONCLUSION

The KDDCup99 dataset is analysed for DoS, U2R, R2L and Probe type of attacks and the performance of different classification algorithms like KNN, ANN and RF, used on it, is compared. Thus, we propose a model that is divided into two sub-models: Feature Selection and Classification.

Feature Selection is used to reduce features irrelevant to DoS, U2R, R2L and Probe attacks and also to reduce the training time and the time taken for real-time prediction of the classification algorithm.

REFERENCES

[1] P. Tao, Z. Sun and Z. Sun, "An Improved Intrusion Detection Algorithm Based on GA and SVM," *in IEEE Access,* vol. 6, pp. 13624-13631, 2018.

[2] I. Ahmad, M. Basheri, M. J. Iqbal and A. Rahim, "Performance Comparison of Support Vector Machine, Random Forest, and Extreme Learning Machine for Intrusion Detection," *in IEEE Access,* vol. 6, pp. 33789-33795, 2018.

[3] B. W. Masduki and K. Ramli, "Improving intrusion detection system detection accuracy and reducing learning time by combining selected features selection and parameters optimization," *2016 6th IEEE International Conference on Control System, Computing and Engineering (ICCSCE),* Batu Ferringhi, 2016, pp. 397-402.

[4] Htwe T.T., Kham N.S.M. (2019) "Improving Accuracy of IDS Using Genetic Algorithm and Multilayer Perceptron Network," *Bhattacharyya S., Hassanien A., Gupta D., Khanna A., Pan I. (eds) International Conference on Innovative Computing and Communications. Lecture Notes in Networks and Systems,* vol 56. Springer, Singapore

[5] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat and S. Venkatraman, "Deep Learning Approach for Intelligent Intrusion Detection System," *in IEEE Access,* vol. 7, pp. 41525-41550, 2019.

[6] C. Sun, K. Lv, C. Hu and H. Xie, "A Double-Layer Detection and Classification Approach for Network Attacks," *2018 27th International Conference on Computer Communication and Networks (ICCCN),* Hangzhou, 2018, pp. 1-8.

[7] I. Z. Muttaqien and T. Ahmad, "Increasing performance of IDS by selecting and transforming features," *2016 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT),* Surabaya, 2016, pp. 85-90.

[8] M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications,* Ottawa, ON, 2009, pp. 1-6.

[9] K. N. Mallikarjunan, K. Muthupriya and S. M. Shalinie, "A survey of distributed denial of service attack," *2016 10th International Conference on Intelligent Systems and Control (ISCO),* Coimbatore, 2016, pp. 1-6.

[10] Sonali Rathore, Amit Saxena and Dr. Manish Manoria, "Intrusion Detection System on KDDCup99 Dataset: A Survey," *(IJCSIT) International Journal of Computer Science and Information Technologies,* Vol. 6 (4) , 2015, 3345-3348.