Anomaly Based Intrusion Detection System Using Neural Network

[1]Ramaprabha J, [2]S. Raghav Pillai, [3]Digaant Garg

ABSTRACT*—Intrusion detection system (or IDS) is an integral part of any Information and Communication Technology (or ICT) system. Building an efficient and reliable IDS that accurately detects an attempt to compromise the network using some known or unknown vulnerability in real time is still a huge challenge. We attempt to create a state-of-the-art Deep Neural Network that analyses the network traffic in real time, identifies an attempt to compromise a network, classifies the type of attack and then compare its accuracy and efficiency with that of conservative and existing models.*

Keywords—*Intrusion Detection, Neural Networks, Cyber Security, Machine Learning, Deep Learning*.

I. INTRODUCTION

In today's era, computers play a very important role in almost every sector. Every organization big or small has computers or servers networked together or connected to the internet performing functions such as storing organization data or providing a service. This network of computers while being a boon to the organization often makes the organization vulnerable to attacks which may sometimes even originate within the organization itself. This creates a demand for an intrusion detection system that can analyse the network traffic in real time and handle any known attempt to compromise the network as well as any unknown attempt (new type of attack).

An intrusion detection system is a piece of software or a network device which is often deployed in strategic positions throughout the network to detect and prevent any and all malicious ventures that may be attempted by someone from both within and outside the organization. The intrusion detection system generally reports any malicious venture detected by it to an administrator or centrally collects the logs using a Security Information and Event Management (SIEM) system. A SIEM system collects and consolidates outputs from various sources and then through alarm filtration techniques classifies malicious activity from false alarms.

II. STATE OF THE ART (LITERATURE SURVEY)

We used this paper [1] to decide which kind of Intrusion Detection System we wanted to design and decided to go with Anomaly Based Intrusion Detection Systems. The taxonomy and survey conducted in this paper helped us narrow down on to Neural Networks and SVMs (Support Vector Machines) as the algorithms to go forward with in order to design the Intrusion Detection System.

[1]*Dept. of Computer Science, SRM Institute of Science and Technolog y*.
[2] *Dept. of Computer Science, SRM Institute of Science and Technology*.
[3]*Dept. of Computer Science, SRM Institute of Science and Technology*.

This paper [2] helped us choose our data set for training our neural network and also helped us get insights into designing our neural networks. This paper proposes a 3 layered neural network which has an accuracy of 93% with the KDD cup 99' data set.

The paper [3] suggests different techniques and method to use for feature selection and elimination for features specific to Intrusion Detection Systems. Inferences from this paper helped us decide which algorithm was to be used to narrow down our feature set. Elimination of the features was done through Correlation and p-value methods. A feature set of 41 was reduced to a feature set of 31.

Following paper [4] gave us insights into how efficient the current and conventional machine learning models are. A comparative study in the paper suggests that an ensemble model of Random Forest Trees paired with Naive Bayes gives the best accuracy of 92.7%. We inferred that a neural network can beat these conventional methods as it allows the establishment of complex relations between unknown parameters in the model.

The implementation of the conventional Machine Learning method of SVM (Support Vector Machine) in this paper [5] gave us inferences to draw analogies between a Neural Network and a Support Vector Machine. The statistics presented in the paper helped us finalize our executing algorithm as a Neural Network.

III. PROPOSED WORK

(1) We propose a Neural Network architecture which uses network parameters as its input and uses them to predict whether the network is being compromised or not.

(2) The Neural Network provides multiple class classification unlike the present state of the art which only provides binary classification.

(3) Tensorflow frame work is used to design the Neural Network.

(4) The output is either the name of the attack taking place on the network or 'normal' if the network is not under any attack.

A. Abbreviations and Acronyms

(1) ***Denial-of-Service-Attack (DoS):*** It's a type of attack in which a person attempts to make the host unreachable through exerting a flood of requests from the target machine and therefore exhausting the host making the host unable to provide services temporarily or in some cases permanently.

(2) Remote-to-Local-Attack (R2L): In this type of attack the attacker who has no user accounts on the target machine, sends data packets to the target and tries to exploit one vulnerability to obtain local access portraying themselves as one of the existing users on the target machine.

(3) User-to-Root-Attack (U2R): In this type of attack the attacker starts by trying to gain access to a user's pre-existing access and exploiting a certain vulnerability to gain root access. This kind of attack of trying to gain root access is also widely known as privilege escalation.

(4) Probing-Attack: In this type of attack the attacker tries to gather information about the computers, and the services running on them, of the network. This is generally done by done by sending a connection request to the server and not responding with an ack packet. This basically tricks the server into giving us information about the kind of service being run on it.

(5) Deep Neural Network (DNN): This algorithm consists of neurons arranged in multiple layers in various amount of numbers according to the need of the model.

The complexity and abstraction of data increases with every layer and unlike orthodox algorithms this algorithm uses nonlinear relations and calculations in order to establish certain relations between parameters which factor in while predicting or classifying data. Hence neural networks require a lot of data in order to test every possible combination of relations between each parameter.

B. Equations

The following mathematical and statistical equations have been used in order to design and optimize our Intrusion Detection System:

(1) Instead of initializing our parameters randomly as most models do we used Xavier Initialization[6] technique to initialize our weight and bias parameters. In the case of Xavier Initialization (also called "Glorot normal"), the parameters are randomly initialized with the mean as zero and standard deviation being :

$$\sigma = \sqrt{\frac{2}{a+b}}$$

Where a is the number of input units in the weight tensor and b is the number of output units in the weight tensor for that layer.

C. Dataset

We have used the KDDCup-99' data set to train our neural network. It consists of 42 parameters including the label of the observation. The following features are present in the dataset:

1	duration
2	protocol_type
3	service
4	flag
5	src_bytes
6	dst_bytes
7	land
8	wrong_fragment
9	urgent
10	hot
11	num_failed_logins
12	logged_in
13	num_compromised
14	root_shell
15	su_attempted
16	num_root
17	num_file_creations
18	num_shells
19	num_access_files
20	num_outbound_cmds
21	is_host_login
22	is_guest_login
23	count
24	srv_count
25	serror_rate
26	srv_serror_rate
27	rerror_rate
28	srv_rerror_rate
29	same_srv_rate
30	diff_srv_rate
31	srv_diff_host_rate
32	dst_host_count
33	dst_host_srv_count
34	dst_host_same_srv_rate
35	dst_host_diff_srv_rate
36	dst_host_same_src_port_rate
37	dst_host_diff_srv_host_rate
38	dst_host_serror_rate
39	dst_host_srv_serror_rate
40	dst_host_rerror_rate
41	dst_host_srv_rerror_rate

After performing data pre-processing and feature elimination we came up with the following features to include in our feature set:

1	duration
2	protocol_type
3	service
4	flag
5	src_bytes
6	dst_bytes
7	land
8	wrong_fragment
9	urgent
10	hot
11	num_failed_logins
12	logged_in
13	num_compromised
14	root_shell
15	su_attempted
16	num_file_creations
17	num_shells
18	num_access_files
19	is_guest_login
20	count
21	srv_count
22	serror_rate
23	rerror_rate
24	same_srv_rate
25	diff_srv_rate
26	srv_diff_host_rate
27	dst_host_count
28	dst_host_srv_count
29	dst_host_diff_srv_rate
30	dst_host_same_src_port_rate
31	dst_host_diff_srv_host_rate

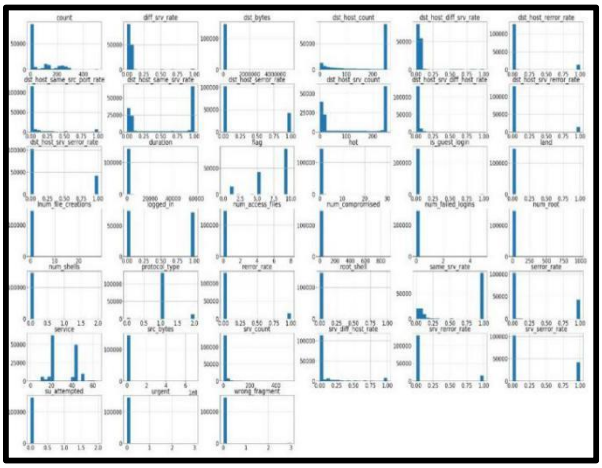


Figure 1: Univariate Histogram of the Features

Received: 22 Sep 2019 | Revised: 13 Oct 2019 | Accepted: 15 Jan 2020

International Journal of Psychosocial Rehabilitation, Vol. 24, Issue 06, 2020
ISSN: 1475-7192

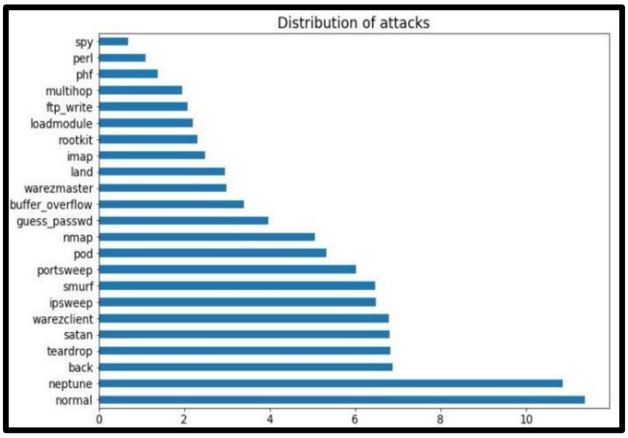


Figure 2: Distribution of attacks in the dataset

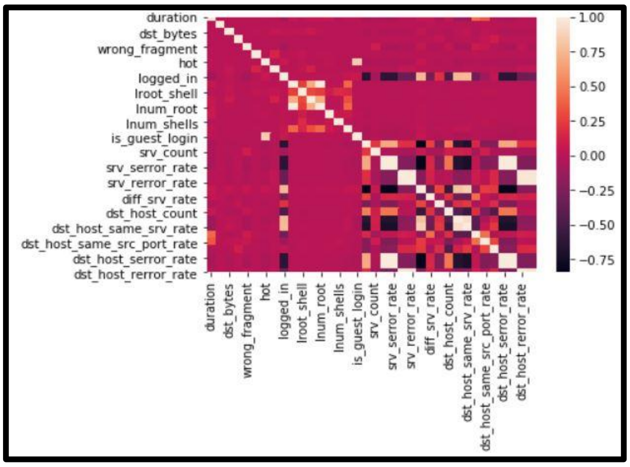


Figure 3: Heat map of the correlation between the features

IV. IMPLEMENTATION

A 4 layered model was defined in TensorFlow as Keras back-end. We trained the neural network with 1000 epochs with a dropout rate of 0.1 for every layer to add regularization to the model. From the inferences of previous papers, we decided to go with a 3 hidden layer model as the task at hand does not require higher complexity to be solved.

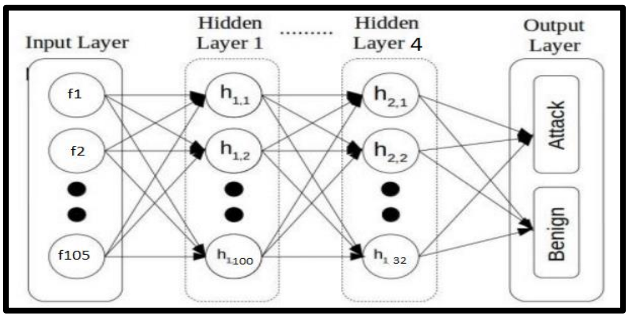
The model gives out an output classifying the observation into the following 23 categories:

'normal', 'buffer_overflow', 'loadmodule', 'perl', 'neptune', 'smurf', 'guess_passwd', 'pod', 'teardrop', 'portsweep', 'ipsweep', 'land', 'ftp_write', 'back', 'imap', 'satan', 'phf', 'nmap','multihop', 'warezmaster', 'warezclient', 'spy', 'rootkit'

Received: 22 Sep 2019 | Revised: 13 Oct 2019 | Accepted: 15 Jan 2020

The proposed Neural Network Architecture:

```
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 100)               10600
_________________________________________________________________
dropout (Dropout)            (None, 100)               0
_________________________________________________________________
dense_1 (Dense)              (None, 64)                6464
_________________________________________________________________
dropout_1 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_2 (Dense)              (None, 32)                2080
_________________________________________________________________
dropout_2 (Dropout)          (None, 32)                0
_________________________________________________________________
dense_3 (Dense)              (None, 23)                759
=================================================================
Total params: 19,903
Trainable params: 19,903
Non-trainable params: 0
```



As seen above the Neural Network takes in 105 inputs in the input layer (one-hot encoding of categorical data lead to 105 inputs) and gives 23 outputs (the type of attack).

V. RESULTS DISCUSSION

We fed the dataset into various conventional as well as the state-of-the-art models and compared it with ours. As the

output is multi class, parameters such as precision, f1 score and recall which are used to measure the success of binary classification cannot be used

Algorithm	Accuracy
Previous State of The Art Model[2]	93%
ADA Boost	92.5%
Decision Tree	92.8%
K Nearest Neighbors	92.9%
Naive Bayes	92.9%
Linear Regression	84.8%
SVM-Linear	81.1%
Svm-rbf	81.1%
Random Forest	92.7%

Our Model	98.64%

As seen above our model outperforms others in case of accuracy.

VI. CONCLUSION

Our paper has successfully established that Deep Learning can be used for the betterment and optimization of Cyber Security. Unfortunately, the Neural Network has been trained on a bygone benchmarking dataset, which is a disadvantage for this methodology.

Even though the statistics provided here are exemplary we need to further conduct studies in integration of deep learning models into real-time networking environment in order to identify and avoid zero-day attacks. This work of study promises to remain a stagnant pointer of direction to further studies in this domain in the near future.

REFERENCES

1. Shallow and Deep Networks Intrusion Detection System: A Taxonomy and Survey by Elike Hodo, Xavier Bellekens, Andrew Hamilton, Christos Tachtatzis and Robert Atkinson.
2. Evaluating Shallow and Deep Neural Networks for Network Intrusion Detection Systems in Cyber Security by Rahul Vigneswaran K, Vinayakumar R, Soman KP and Prabaharan Poornachandran
3. Feature Extraction Methods for Intrusion Detection Systems by Hai Thanh Nguyen, Katrin Franke, Slobodan Petrović
4. Study on implementation of machine learning methods combination for improving attacks detection accuracy on Intrusion Detection System (IDS) by Bisyron Wahyudi Masduki, Kalamullah Ramli, Ferry Astika Saputra, Dedy Sugiarto
5. A Comparative Analysis of SVM and its Stacking with other Classification Algorithm for Intrusion Detection by Nanak Chand, Preeti Mishra, C. Rama Krishna, Emmanuel Shubhakar Pilli and Mahesh Chandra Govil

Received: 22 Sep 2019 | Revised: 13 Oct 2019 | Accepted: 15 Jan 2020