Intrusion Detection using Deep Learning

Keerthi Kumar S (1811EE06), Sandip Kishore (1711CS12), Mandeep Rathee (1811MC07)

Project abstract

With the computer networks being prevalent, there is always a concern for securing these networks. These networks are often vulnerable to attacks which mainly aim at hogging the network resources such as bandwidth, memory, CPU resource of the server etc. When such entities consume the available resources of the network, the actual users might face the degraded service or in most cases they'll be denied of the available service. Hence these kind of attacks are commonly called as 'Denial of Service'(DoS) attacks. When such attacks come from the systems that are distributed across the networks, often referred to as DDoS, it becomes difficult for the network elements to characterize and eliminate them. Also, a new technique is being invented daily for such attacks. Hence detection of DDoS attack is often a challenging task that requires some degree of adaptability. In this project we have developed a 'Network Intrusion Detection System' (NIDS) based on Deep Learning. We have used Deep Neural Network (DNN) architecture for the classification of packets into 'Benign' and 14 different attack categories from the available **CICIDS2017** dataset [1]. The model is then evaluated on the test samples generated by combining the benign traffic captured from the live network and the attack samples from the dataset. Further, scope for testing different models using Convolutional Neural networks(CNN) and Recurrent Neural Networks(RNN) have been considered.

1 Introduction

Intrusion Detection Systems(IDS) act as the first line of defense in the network security by forewarning the network administrators about the malicious behaviors such as intrusions, malware etc. These intrusions on a network might have technical and non-technical consequences. It might cause the network resources become unavailable for the rightful users for a brief or long duration of time. This may in turn affect the normal operation of the organization that is dependent on this network. Hence, IDS are a mandatory line of defense to secure networks against these ever increasing intrusions. Also, as there are new techniques being evolved everyday to launch these attacks, the IDS must be adaptable and should be up to date. These systems must also be scalable and must cater to the security requirements of heterogeneous networks.

In this project, we propose a DNN based IDS that is trained using the latest IDS dataset **CICIDS2017** made available by the *Canadian Institute of Cybersecurity*. This dataset contains most common attacks based on the 2016 McAfee report, such as Web based, Brute force, DoS, DDoS, Infiltration, Heart-bleed, Bot and port Scan. The model is then evaluated on the traffic captured from the live campus network.

2 Literature survey

In recent years, there has been a growing trend in application of Artificial Intelligence(AI) techniques to solve complex problems. Performance of several Machine Learning(ML) algorithms have been studied for the IDS[2][3]. Since the beginning of the Deep Learning(DL) era, people have studied the feasibility of application of DL models for the IDS. For example, the paper by N. Shone et al.[4] discusses the application of auto encoders for the Network Intrusion Detection Systems(NIDS). The paper by C. Yin et al.[5] discusses the application of RNNs for the IDS. Both of them used the KDD Cup '99 and NSL KDD 2009 dataset for the evaluation of their models.

These datasets have their own shortcomings and has been addressed in the paper[1]. No significant work is available on the latest **CICIDS2017** dataset using the DL approach.

3 Objectives

The objective of this project is to build and train a neural network model for Network Intrusion Detection. Once the model is trained on the available dataset, we tested it on the traffic captured from the live network along with the traffic data available in the dataset, and evaluated its performance. With this we can have an IDS system that acts as a first line of defense for the campus network, thereby warning the network admin about the adversarial attacks.

4 Summary of contributions

In this project, a real world problem has been identified pertaining to live campus network. A new paradigm for IDS based on anomaly detection against classical signature based approach has been explored. Majority of the work in this area has been carried out using the old KDD data set as benchmark. This data set is not only outdated but also had many limitations. These limitations has been addressed in the CICIDS 2017 dateset which is relevant to new attack scenarios. Moreover, this team did not find any standard work which has used this CICIDS dataset with deep learning approach.

Unique selling proposition of the project

The methodology and architecture to generate the CICIDS data set has been explored in this project. CICIDS uses a basic custom built architecture to generate the benign and attack profiles, whereas through this project a more novel, advanced and realistic method and architecture to generate the dataset based on simulators has been identified and can be taken up a separate project. This new approach may generate a more practical, realistic and comprehensive data set.

Further exploration of this project with CNN, RNN and other neural networks will help to evaluate the existing signature based enterprise equipments like firewalls, peripheral devices, gateways etc for efficiency and performance in detecting and mitigating the attack profiles.

5 Methodology

Python language is used for programming. The `tensorflow` library is used for implementing the model.

Data pre-processing

The dataset contained few of the features with values such as 'NaN' and 'Infinity'. Eliminating these as outliers would reduce the number of samples available for some of the classes of data. Also the the Infinite value for the feature such as 'flow Bytes' or 'Inter Arrival Times' could add some meaning to the sample for the particular class. Hence, we retained these datapoints by replacing the 'NaN' by 0 and the 'Infinity' by maximum float32 value(3.4028235e+38) using the built-in numpy function `nan_to_num`.

Model

The model consisted of an Input Layer with 78 neurons, an Output Layer with 15 neurons and three densely connected hidden layers with 78, 50 and 25 neurons respectively. ReLU activation function is used in the hidden layer and softmax activation function is used in the Output layer. The model is as shown in the figure 1.

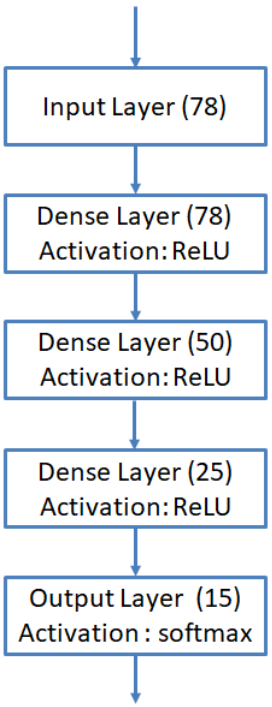

Figure 1: Deep Neural Network model

To train the model, we used the Stochastic Gradient descent(SGD) optimizer with the learning rate set to 0.1. The categorical cross-entropy function is used for the loss function i.e.

$$Loss = -\frac{1}{N}\sum_{i=1}^{N} \log p_i[y \in M]$$

where,
M = number of classes
N = number of samples
p_i = probability that the given sample i belongs to one of the M classes

6 Results

The following sections contain the brief description of the dataset along with thedetails of the train and test samples used. The results are provided with the confusion matrix, accuracy, precision, recall, F1-score for multi-class classification. Also, the results are discussed for binary classification.

Description of dataset

The **CICIDS2017** dataset consists of labeled network flows, including full packet payloads in pcap format, the corresponding profiles and the labeled flows and CSV files. This dataset was generated in the *Canadian Institute of CyberSecurity*(CIC) in the year 2017. The CICIDS2017 dataset contains benign and the most up-to-date common attacks, which resembles the true real-world data (PCAPs). It also includes the results of the network traffic analysis using CICFlowMeter with labeled flows based on the time stamp, source and destination IPs, source and destination ports, protocols and attack (CSV files). These data were captured over a period of 5 days in their custom set up lab. The generated attack diversity included the most common attacks based on the 2016 McAfee report, such as Web based, Brute force, DoS, DDoS, Infiltration, Heart-bleed, Bot and port Scan. About 80 network flow features from the generated network traffic were extracted using *CICFlowMeter* tool that can deliver the network flow dataset as a CSV file. More details on the set up topology, captured data and the tools used can be found here[6].

For training our model, we decided to drop the the destination and source ip fields as these fields may vary with the source of the packets. Hence the number of features used are 78.

Train and Test split

Certain classes of data had very less number of samples compared to others. Oversampling of these classes affected the accuracy of the BENIGN class. Hence, we undersampled the BENIGN class. First, the test samples for the attack classes were built using the following approach:

$$N_{test} = min(1000, (0.5 * N))$$

where, N = number of samples for each class.
These attack samples were merged with the live BENIGN traffic captured from the campus network to obtain the final test samples.

The rest of the attack samples were merged with the BENIGN samples captured on Monday from the available dataset and used for training the model. The details are provided in the table 1.

Category	Training Samples	Test Samples
BENIGN	528911	22501
FTP-Patator	6938	1000
SSH-Patator	4897	1000
DoS slowloris	4796	1000
DoS slowhttptest	230073	1000
DoS GoldenEye	4499	1000
DoS Hulk	9293	1000
Heartbleed	6	5
Web Attack XSS	352	300
Web Attack Sql Injection	11	10
Web Attack Brute force	1007	500
Infiltration	21	15
Bot	966	1000
PortScan	127027	1000
DDoS	157930	1000
Total	**1076727**	**32331**

Table 1: Train and Test data sample numbers

Results & discussion

The average test accuracy obtained was 97.16%. The Loss vs. Epoch curve for train and test data is as shown in the figure 2

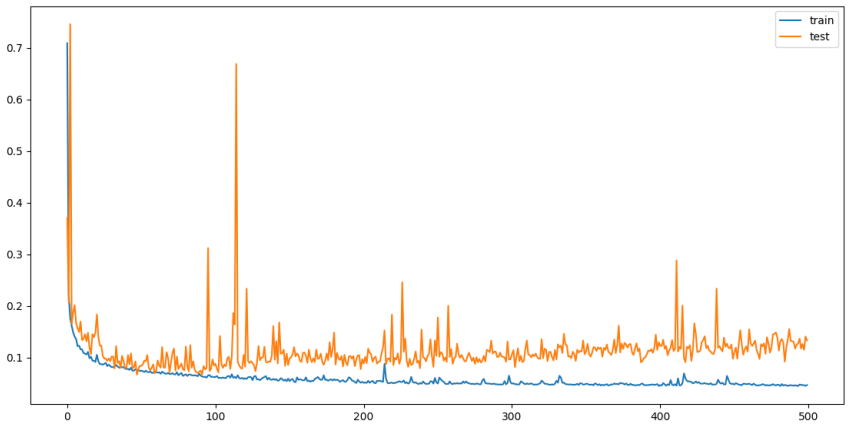

Figure 2: Loss vs. Epoch curve

The confusion matrix obtained for the test samples is given in table 2

22501	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	993	6	0	0	0	0	0	0	0	1	0	0	0	0
1	2	997	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	988	1	2	0	0	0	0	5	0	0	0	0
1	0	0	71	928	0	0	0	0	0	0	0	0	0	0
23	0	32	0	2	943	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	998	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	4	0	0	0	0	0	0	0
1	0	0	0	0	3	0	0	16	0	280	0	0	0	0
1	0	1	0	0	4	0	0	0	0	4	0	0	0	0
5	0	0	0	0	3	1	0	1	0	490	0	0	0	0
11	0	0	0	0	0	0	0	0	0	3	1	0	0	0
335	0	0	0	0	0	0	0	0	0	0	0	665	0	0
3	0	0	0	0	3	1	0	6	0	73	0	0	906	8
3	0	0	0	1	7	0	0	0	0	0	0	0	6	983

Table 2: Confusion Matrix

The classification report for the multiclass classification is given in table 3

Category	Precision	Recall	F1-Score	Support
BENIGN	0.98	1.00	0.99	22501
FTP-Patator	1.00	0.99	1.00	1000
SSH-Patator	0.96	1.00	0.98	1000
DoS slowloris	0.93	0.99	0.96	1000
DoS slowhttptest	1.00	0.93	0.96	1000
DoS GoldenEye	0.98	0.94	0.96	1000
DoS Hulk	1.00	1.00	1.00	1000
Heartbleed	1.00	0.80	0.89	5
Web Attack XSS	0.7	0.05	0.10	300
Web Attack Sql Injection	0.00	0.00	0.00	10
Web Attack Brute force	0.57	0.98	0.72	500
Infiltration	1.00	0.07	0.12	15
Bot	1.00	0.67	0.80	1000
PortScan	0.99	0.91	0.95	1000
DDoS	0.99	0.98	0.99	1000

Table 3: Classification report for multiclass

Binary Classification results

This section provides the results for the two-class classification i.e. BENIGN and Attack. The average accuracy obtained was 98.79%. The confusion matrix is given by the table 4

	BENIGN	**Attack**
BENIGN	22501	0
Attack	391	9439

Table 4: Confusion matrix for binary classification

The classification report for the binary classification is given in table 5

	Precision	Recall	F1-Score
BENIGN	0.98	1.00	0.99
Attack	1.00	0.96	0.98

Table 5: Classification report for binary classification

GitHub link

The source code for this project can be found at:
`https://github.com/sandip-kishore/CS551-Deep-Learning-Apr-2019-Group11.git`

7 Conclusions

In this project, we have explored a deep learning approach to solve a real world problem of Intrusion detection. This problem is well known and it is being handled by a signature based approach which we strongly believe that it will become outdated in near future. The reason being there has been rapid evolution in attack mechanisms and it is very difficult to keep up with signature based approach. A more proactive and preventive anomaly detection method is call of hour to tackle this problem. It will lay foundation to the more efficient game theoretic approach to solve this global problem since this is a first line of defense for any IT set up.

References

[1] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani, *"Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization"*, *4th International Conference on Information Systems Security and Privacy (ICISSP)*, Portugal, January 2018

[2] M. Almseidin, M. Alzubi, S. Kovacs and M. Alkasassbeh, *"Evaluation of machine learning algorithms for intrusion detection system," 2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY)*, Subotica, 2017, pp. 000277-000282.

[3] P. Mishra, V. Varadharajan, U. Tupakula and E. S. Pilli, *"A Detailed Investigation and Analysis of Using Machine Learning Techniques for Intrusion Detection,"* in *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 686-728, Firstquarter 2019.

[4] N. Shone, T. N. Ngoc, V. D. Phai and Q. Shi, *"A Deep Learning Approach to Network Intrusion Detection,"* in *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41-50, Feb. 2018.

[5] C. Yin, Y. Zhu, J. Fei and X. He, *"A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks"*, in *IEEE Access*, vol. 5, pp. 21954-21961, 2017

[6] `https://www.unb.ca/cic/datasets/ids-2017.html`