Network Intrusion Identification Using Deep Learning

Nicholas Bashour (#06449756) – CS 230 – Spring 2021

https://github.com/nick-bash/network-intrusion-detector

I. Introduction

In an increasingly digital world, organizations across the globe are facing the ever more challenging task of defending their digital assets and infrastructure. Malicious actors are progressively more pernicious – be it through infrastructure attacks, ransomware, or supply chain compromises.

With the rise of cheap computing resources and significant advances in machine learning algorithms, cybersecurity researchers have recently begun to leverage deep learning and other related algorithms to bolster the defenses of enterprises. This project aims to contribute to the work of such researchers by creating a deep learning algorithm to detect network intrusions based on data originating from network flows. The hope is that this work can contribute to potentially improving Intrusion Detection Systems (IDS's) and Intrusion Prevention Systems (IPS's) as deployed in the field.

II. Dataset

The Canadian Institute for Cybersecurity, based at the University of New Brunswick in Fredericton, is an interdisciplinary institute that hosts various researchers from computer sciences, business, law, and other domains to collaborate on research efforts related to cybersecurity. Their team recently created a dataset known as the Intrusion Detection Evaluation Dataset (CIC-IDS2017) which serves as the basis for this project.

As outlined in detail in [1], the CIC-IDS2017 dataset was created to supplant various datasets that had been relied upon in cybersecurity literature for research to improve intrusion detection techniques. The foremost of these historical datasets are the KDD Cup 1999 (a database of 18M instances of network activity, simulated in a military environment in 1999) and the NSL-KDD (a refined version of the KDD Cup 1999 data). The researchers in [1] cite various shortfalls of the KDD data and other data common in the literature, such as: a lack of traffic diversity, constraints on the variety of the attacks present in the data, and a lack of a broad feature set available for analysis.

The CIC-IDS2017 dataset was created by simulating a variety of attacks over the course of a week in a realistic testbed infrastructure. The simulated network included 4 attacker and 10 victim machines, with 25 users, using multiple protocols, such as HTTPS, HTTP, FTP, SSH, and email. The attacking machines attempt various attacks over the simulation, including brute force, Heartbleed, Botnet, DoS, DDoS, and infiltration attacks, for a total of 14 unique attack types.

The size of the CIC-IDS2017 dataset is significant, with a total of 78 features measured across over 2.8 million instances of network flows. A flow represents "a sequence of packets with the same values for Source IP, Destination IP, Source Port, Destination Port and Protocol (TCP or UDP)." Along with basic data around the network flow, such as the destination port, the flow duration, and the total number of packets, the CIC-IDS2017 dataset contains a significant number of other fields captured using CICFlowMeter. CICFlowMeter is an open source tool that extracts additional features from network flows in .pcap files. Some of these features include the max, min, mean, and standard deviation of packet length, the flow's active / idle time, and the interarrival time between flows. See Figure 1 in the appendix for a few examples of the collected feature set.

III. Approach

The model uses a deep learning approach to predict class labels based on the input features. As of now, the model uses 5 fully connected layers with 30, 24, 16, 10, and 15 hidden nodes in each layer, respectively. The last layer uses a softmax function to compute the predicted class.

The ultimate metric which will be used to measure the model's performance is an average of the test accuracies measured for each of the 14 attack types. To elaborate, I will construct 14 test samples representing each of the 14 attack types, measure the accuracy of each, and then average the result.

IV. Progress to Date and Work Outstanding

As of the project milestone, I have completed the items below. Note that, due to constraints of computation / processing speed, the below items have only been completed for 1 of the 8 data files (approximately ~8% of the total dataset).

- Created pre-processing script
  - Processed raw .csv files into vectorized features and labels
  - Randomly sorted data into training and test sets; normalized data using the training set's mean and standard deviation
- Created baseline model
  - Implemented and trained a five layer fully connected model using tensorflow

To complete the project, the following items are outstanding:

- Repeat the steps above for the remaining 7 of 8 data files (data pre-processing and model training)
- Create the remaining 13 test sets for each of the untested 13 labels
- Tune the model to improve class-wise average accuracy. Potential hyperparameters to tune include model depth, number of hidden nodes, and learning rate. Additionally, I may explore changing the network structure to a CNN; this approach has been demonstrated in an intrusion detection context in [2].

V. References

1. Sharafaldin I., Lashkari AH, Ghorbani AA. "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization", 2018.

2. Chowdhury, MMU., Hammond, F., Konowicz, G., Xin, C., Wu, H., Li, J. "A Few-shot Deep Learning Approach for Improved Intrusion Detection", 2017.

Appendix

Figure 1: Example Data from CIC-IDS2017

	Example 1	Example 2	Example 3	Example 4
Destination Port	389	80	80	80
Flow Duration	113,095,465	5,185,118	5,006,127	5,638,432
Total Fwd Packets	48	7	4	3
Total Backward Packets	24	7	4	1
Total Length of Fwd Packets	9,668	1,022	447	-
Total Length of Bwd Packets	10,012	2,321	530	-
Fwd Packet Length Max	403	372	447	-
Fwd Packet Length Min	-	-	-	-
Fwd Packet Length Mean	201	146	112	-
Fwd Packet Length Std	204	184	224	-
Bwd Packet Length Max	923	1,047	530	-
Bwd Packet Length Min	316	-	-	-
Bwd Packet Length Mean	417	332	133	-
Bwd Packet Length Std	231	440	265	-
Packet Length Variance	54,678	109,776	46,831	-
Subflow Fwd Bytes	9,668	1,022	447	-
Subflow Bwd Packets	24	7	4	1
Subflow Bwd Bytes	10,012	2,321	530	-
Init_Win_bytes_forward	571	29,200	29,200	29,200
Init_Win_bytes_backward	2,079	252	235	28,960
act_data_pkt_fwd	23	3	1	-
min_seg_size_forward	32	32	32	32
Active Mean	203,986	-	-	-
Active Std	575,837	-	-	-
Active Max	1,629,110	-	-	-
Active Min	379	-	-	-
Idle Mean	13,800,000	-	-	-
Idle Std	4,277,541	-	-	-
Idle Max	16,500,000	-	-	-
Idle Min	6,737,603	-	-	-
Label	Benign	Brute Force	SQL Injection	XSS

Note: the examples shown above do not include all input features