CSC 180-01 Intelligent Systems

Mini-Project 2: Network Intrusion Detection System

Due at 2:00 pm, Friday, October 16, 2020

Tran Ngoc Bao Huynh (Student ID: 219763298)

Ong Thao (Student ID: 219467431)

Problem Statement

The ability to detect network intrusions from unauthorized users and even insiders is imperative in the field of cybersecurity as it can help system administrators and security specialists better defend against cyber attacks. In this project, we aim to help address this problem by building an AI-based Network Intrusion Detection System (IDS) that is able to distinguish between good (e.g. normal) and bad (e.g. intrusions or attack) connections. Our problem is modeled as a binary classification problem using both Fully-Connected and Convolutional Neural Networks (CNN).

Methodology

Our first step was data preprocessing. When we first loaded in our dataset, some things that we did to clean and better organize our data includes: defining column names, dropping redundant records, and removing records with missing values. We then performed label encoding by looping through each record in the "outcome" column and assigning "0" to good or normal connections and "1" for everything else. After this, we normalized all of our categorical and numerical features, converted our data into x and y inputs, and divided the dataset into training and test data to later feed into our models.

Following the completion of data preprocessing, we started the implementation of our models. For our fully-connected neural network, we decided to go with a five-layer neural network with three hidden layers containing 50, 25, and 10 neurons respectively. Meanwhile, for the Convolutional Neural Network, we try out different numbers of layers, and different kernel sizes. By the result from fully-connected neural networks, we found that Relu/Adam is one of the best combinations that provide high accuracy. Therefore, we use these two to implement 2 and 3 Convolutional Layers with 2x2, 3x3, and 4x4 kernels. After

regularizing the input, there are 100 column features, so we choose the image dimension as 10 x 10. We tuned the size of kernels, layers number, and also referred to some other academic paper (link at the end of the report) for these variables to test the performance**.** Additionally, similar to the last project, we also tried out all combinations of the hyperparameters to see which activation function and optimizer combination produces the best results for each model.

Experimental Results and Analysis

In terms of the fully-connected neural networks, the precision, recall, f1-score, and final accuracy for every one of our models is 1.0 meaning that our models were in general very accurate in predicting which connections were bad and which ones were good. This also shows that despite which activation function and optimizer we used, the results were roughly the same with only a few decimal point differences in performance (as seen in the table below). From our results below, the Relu/Adam combination had the best performance while Sigmoid/SGD had the worst performance.

Fully Connected Neural Network Results			
Activation	**Optimizer**	**Loss**	**Accuracy**
Relu	Adam	0.00532911904156208	0.9985713108223205
Sigmoid	Adam	0.00642536161467433	0.9984339368629283
Tanh	Adam	0.005836235359311104	0.9984614116548067
Relu	SGD	0.007775162346661091	0.9984064620710498
Sigmoid	SGD	0.009153603576123714	0.9979119158172377
Tanh	SGD	0.006958499550819397	0.9982416133197791

For the Convolutional Layer Model, we got 1.0 for most of the recall, f1 score, and precision. Accuracy of the models are around 0.997 (99.7%). We found that the model with 2 2-D Convolutional Layers and 4x4 size kernel provides the highest accuracy:

Convolutional Neural Network Results	
Model	**Accuracy**
2 Conv Layers & 2x2 Kernels Number of Kernels: 32 (1st Layer) and 64 (2nd layer)	0.9981317141522653
2 Conv Layers & 3x3 Kernels Number of Kernels: 32 (1st Layer) and 64 (2nd layer)	0.9985438360304421
2 Conv Layers & 4x4 Kernels Number of Kernels: 32 (1st Layer) and 64 (2nd layer)	0.9984064620710498
3 Conv Layers & 2x2 Kernels Number of Kernels: 32 (1st Layer), 64 (2nd layer) and 128 (3rd Layer)	0.9984339368629283
3 Conv Layers & 3x3 Kernels Number of Kernels: 32 (1st Layer), 64 (2nd layer) and 128 (3rd Layer)	0.9985438360304421
3 Conv Layers & 4x4 Kernels Number of Kernels: 32 (1st Layer), 64 (2nd layer) and 128 (3rd Layer)	0.9983515124872929

From the result, with 2x2 and 3x3 dimension kernels, the accuracy is increased when we increase the number of Convolution Layers. However, this does not apply for 4x4 kernel size. 4x4 kernel size with 3 convolution layers has the least accuracy result.

Task Division and Project Reflection

For the project, all members contributed to all parts of the project. We each had our own implementation of the whole project then merged our findings and results together while brainstorming about how we can better improve our models by tweaking some parameter values and doing lots of testing/experimentation.

For the Convolutional Neural Network, we planned to implement Auto Encoding, and also test the performance when turning the data into RPG images as the study from IEEE paper [1]. However, we haven't succeeded. We will continue to try out this method. The project is really interesting and useful. Collectively as a team, we were able to better learn how to approach a binary classification problem and use Fully-Connected and Convolutional neural networks to solve this problem. Being able to deep-dive and learn more about other neural networks and approaches along with getting more well versed with Python was a very valuable experience we obtained through this project. Additionally, we were able to practice better time management skills and complete the project in a more timely manner than the first project.

REFERENCES

[1] Y.Xiiao, C. Xing, T. Zhang, and Z. Zhao, "An Intrusion Detection Model Based on Feature Reduction and Convolutional Neural Networks," https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8666014 ; March 2019.

[2]Jiyeon Kim, Jiwon Kim, H.Kim, M. Shim, and E. C, "CNN-Based Network Intrusion Detection against Denial-of-Service Attacks," https://www.mdpi.com/2079-9292/9/6/916; June 2020