Veena Mounika Ruddarraju (ID: 220269453)

Yarraguntla Saiswetha (ID:220268894)

CSc 215-01 Artificial Intelligence

Project #1

Mini-Project 1: Modern Low Footprint Cyber Attack Detection

Due Date: 09/25/2019

Problem Statement

Network intrusion is an unauthorized activity on a computer network. A Network Intrusion Detector is a software that detects suspicious traffic in the network. Whenever it encounters a spurious connection, it immediately alerts about that unauthorized user. In this project, we designed a network intrusion detector that could detect bad connections.

Methodology

We used UNSW-NB dataset created in the Cyber Range Lab of the Australian Center for Cyber Security (ACCS). The given dataset was divided as 175,341 training records and 82,332 testing records. In the first stage, we started the process of data cleaning, where we looked for null values in both training and testing data and found some null values on both the sets. We removed the rows that had at least one null value and the numbers of records reduced to 81173 training and 35179 testing records. Then we normalized the numeric values using Z-score and performed one hot encoding to the nominal and binary data in the dataset. Next, we tried different feature selection techniques and we got good results with Pearson correlation. At first, we plotted the Pearson correlation heatmap and found the correlation of independent variable with target variable 'label'. Then we selected 16 features, having correlation greater than 0.3 with the target variable. After observation, we found that some of the selected features are correlated with each other, then we removed 3 of the selected features to overcome overfitting of the model. Finally, we got 13 features which are independent and uncorrelated with each other.

We implemented Logistic Regression model and achieved 92% accuracy. For Nearest Neighbor and Support Vector Machine due to memory leak and runtime constraints, we just took only 50k rows from the entire dataset and acquired accuracy of 91% and 92% respectively.

For Neural Networks, we converted the data frame into a format where tensor flow needs. Same way we did it for the training data also. The network is build using 50 dense input layer neurons with relu activation function, 50 hidden layers neurons with relook function and the single output layer with softmax function.
To achieve best model, the model is compiled using categorical crossentropy loss and adam optimizer. The best model is saved using early stopping by monitoring the loss value.

Similarly, we built different models using parameter tuning between the number of neurons, hidden layers, activation functions, and optimizers to see the difference in precision of the model.

Experimental Results and Analysis

Model	Accuracy	Precision	Recall	f1-score
Logistic Regression	92%	0.93	0.92	0.92
Nearest Neighbor	91%	0.91	0.91	0.91
Support Vector Machine	92%	0.93	0.92	0.92

Out of Logistic Regression, Nearest Neighbor and Support Vector Machine, Logistic Regression and Support Vector Machine models gave highest accuracy of 92%.

Neural Networks

Activation Function	Adam	Sgd	#Layers	#Neurons
Sigmoid	94.3	92.1	2	50
ReLU	94.4	94.4	2	50
Tanh	94.8	94.6	2	50

We used different combinations with Adam and Sgd optimizers and Sigmoid, Relu and Tanh activation functions. Out of these combinations, Tanh and Adam combination have highest accuracy with 94.8%.

Task Division and Project Reflection:

Veena Mounika: Data Preprocessing, Logistic Regression, Nearest Neighbor

Swetha: Data Preprocessing, Support Vector Machines, Neural Networks.

Challenges

1. When we trained the model with all the features we got low accuracy for all the models.

2. During normalization of numeric values, we tried different combinations and finally it worked when we normalized all the features.

3.We then tried L1(Lasso) regularization, Chi-squared test and Pearson Correlation. Pearson Correlation gave good results.

4.Our models were overfitting at the beginning, then we realized that some of the selected features were correlating with each other and then we removed those particular features to overcome overfitting problem.

5. Initially SVM took lot of time to execute, feature selection reduced the duration.

What we have learned from this project?

We have learnt every step starting from the Data Preprocessing, like Data Cleaning, Feature Transformation and Feature Selection. We also understood how minute things in data preprocessing effects the accuracy of the model. Also, we learned how to build different models. Especially, how to build neural networks, how a neural network works and how the accuracy is getting changed for different layers count and neurons count and the importance of EarlyStopping and ModelCheckpoint.