Report

Angelos Kelekoglou

Project Ttile: Network Anomaly Detection With Assistance of Deep learning.

Email: aggeloskel@outlook.com

Academic email : akelekog@csd.auth .gr

Github : https://github.com/akelekog

Network security is becoming more important as a result of the massive growth in computer network traffic and the abundance of applications. All computer systems are prone to security flaws, automating the detection of anomalies is an efficient way of approaching the problem. This project is an indication of the ability of neural networks to simplify anomaly detection.

The project has been implement on jupyter noteboook. In total there have been created one model This data is KDDCUP'99 data set(see https://www.kaggle.com/datasets/anushonkar/network-anamoly-detection for the listing of the features), which is widely used as one of the few publicly available data sets for network-based anomaly detection systems. In particular the model created consists of:

An input layer

Dense layer with 32 neurons and relu activation

Dense layer with 64 neurons and relu activation

Dense layer with 128 neurons and relu activation

Dense layer with 64 neurons and relu activation

Dense layer with 32 neurons and relu activation

Dense layer with 1 neurons and sigmoid activation

This model was chosen for it’s short training time and adequate results.

We start by reading the dataset and assigning the features dropping all the ones that mostly have zero values. We then use a label encoder for the categorical features and proceed to created the training set. The problem is approached with a binary classification with a class distribution of :

```
Class distribution: 1: 67343, 0: 58630
```

Where 0 is normal and 1 is attack.

The data is scaled and we have a training set of 125973 samples with 38 features.

The training is done for 10 epochs with a batch size of 32.

The results from the training indicate overfitting , with an accuracy on the test set of 86%

Note:

The attack ATTACK activity is : normal or DOS or PROBE or R2L or U2R
but since the problem is approached with binary classification the values become normal and attack.

The directory of the python notebook and the file containing the dataset should be the same.

Assets used:

https://www.kaggle.com/datasets/anushonkar/network-anamoly-detection