Executive Summary- A Neural Network Model for Detecting Intrusions on a Computer Network

Overview

This report delivers an evaluation of the instruction detection system (IDS) designed for the experiment. The system can detect and classify intrusion attacks, even according to their categories. The analysis is based on the graphical results obtained while testing.

Problem

The ten percent KDD dataset [1] containing 42 columns, and 494021 rows, was used for the experiment. The last column contains the type of attack, which is used as a target for the neural network. The neural network after getting trained on it is expected to be able to predict the attack type when new data is presented to it, with high accuracy.

Data Pre-Processing

The system was built by fetching the dataset directly from the URL link of the KDD dataset to have the capability to adapt to a dynamically changing dataset if tried for some other. Then for the data pre-processing, the binary data was treated as categorical data and some columns turned out to have constant values and were removed. The null values were checked as well. As the numerical data was different, in range, for each column, it was normalized using the z-score method, trying both absolute deviation and standard deviation to bring the values close to each other for them to be compared effectively with each other. The correlation was used as a method to find out the correlation between numerical input columns to get rid of correlated inputs. Heat maps were used for the data frames before and after the correlation to visualize the correlation between the inputs. The entire data was encoded using the one-hot vector encoding method which increased the column size from twenty-three to one hundred and two. The principal component analysis was performed to reduce the dimensions back to twenty. The data was split into groups of training data (eighty percent) and the rest as the testing data to be used for the training and testing of the neural model.

Model Creation

For the model creation, the sequential model was used so that the batches of inputs could be passed to it sequentially. Instead of having a fixed training data set, chosen at random, K-fold cross-validation [2] was used, in two ways, first by finding the best data split using it and then running the entire training on that data split. The other way tried was by calculating the accuracy by taking the average of the K-1 r-squared scores.

Tuning and Results

The multi-class [3] and the binary class intrusion detection system were tuned separately as the first one was far more complex. But the tuning of each of the systems was tuned very carefully starting with just 1 neuron and 1 layer and gradually increased to find the best-suited topology for the neural network. Different activation functions were tried, 'Relu' was finalized for activation hidden layers and 'SoftMax' for the final layer. Finally, the batch size of 128, learning rate at 0.1, Adam optimizer, five K-folds, correlation threshold of 0.7, z-score using standard deviation, four intermediate neural layers with the number of neurons as sixteen, sixty-four, sixteen, and five respectively, gave the highest, most stable and reliable accuracy of 99.91 percent.

Future work recommendations

Many more classifiers like Decision tree, Naive Bayes, Random Forest, Support Vector Machine, K-Means clustering, K-Nearest neighbors can be used along with the correlation and PCA used in the experiment to compare the results. A deeper knowledge of the dataset to know the importance of each feature could be very useful. For example, the number of the occurrence of some of the attacks were extremely low. If they were not important, they could have been removed to get even higher accuracy. Since, in the end, there is a trade-off between the accuracy and the topology of the neural network. A better balance could have been achieved if the severity of each attack was known.

References

[1] Revathi, S. and Malathi, A., 2013. A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection. *International Journal of Engineering Research & Technology (IJERT)*, *2*(12), pp.1848-1853.

[2] Anguita, D., Ghelardoni, L., Ghio, A., Oneto, L. and Ridella, S., 2012, April. The'K'in K-fold Cross Validation. In *ESANN* (pp. 441-446).

[3] Ou, G. and Murphey, Y.L., 2007. Multi-class pattern classification using neural networks. *Pattern Recognition*, *40*(1), pp.4-18.

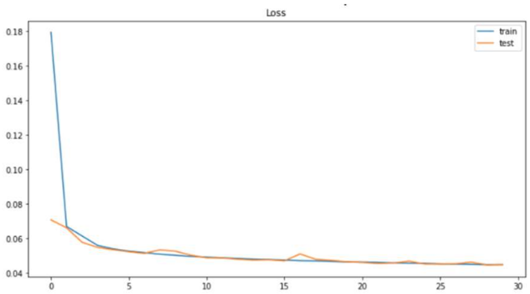
APPENDIX-Tuning and Testing

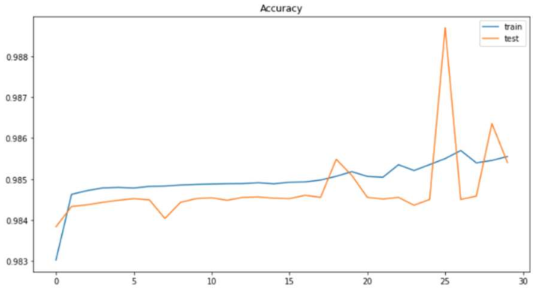
Bare minimum Configeration

For the Number of Neurons

Single Layer using Adam optimiser neural network with 'Adam' optimizer and 'Cross-Entropy' Loss function run for FULL 30 epochs.

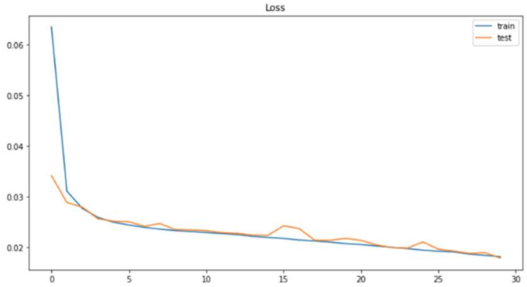
Number of neurons	Training Accuracy	Testing Accuracy
1	0.986	0.985

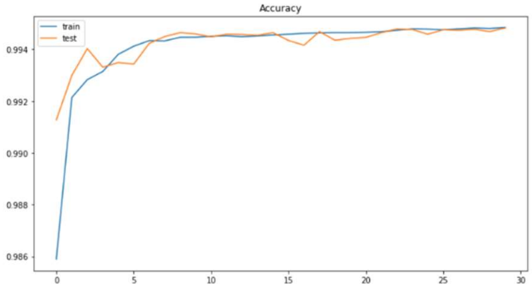




Better Results

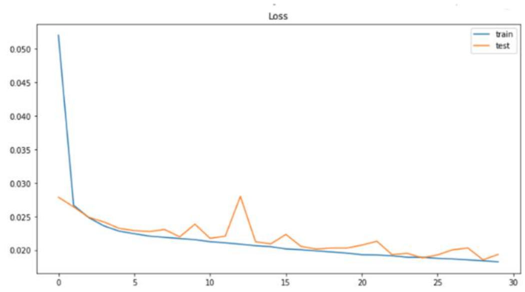
Number of neurons	Training Accuracy	Testing Accuracy
5	0.995	0.995

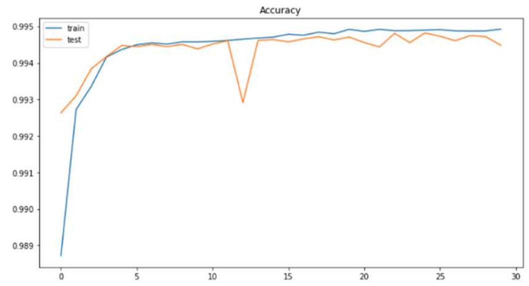




Number of neurons	Training Accuracy	Testing Accuracy
10	0.995	0.994

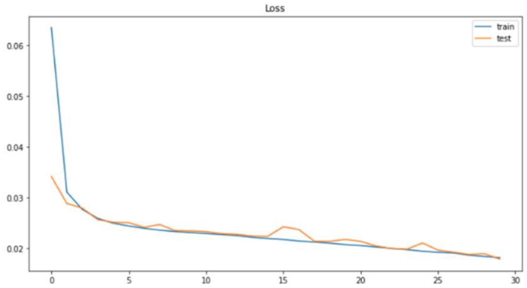
worse Results.

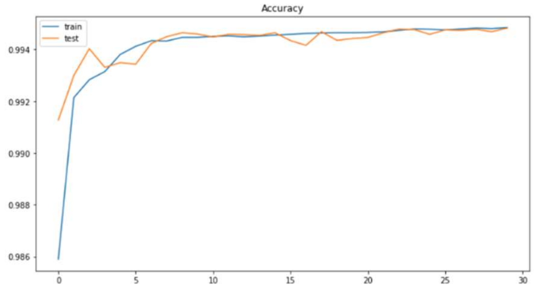




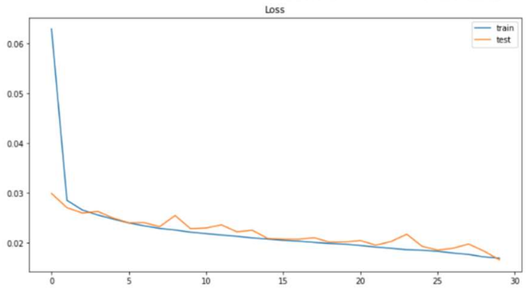
Number of Layers

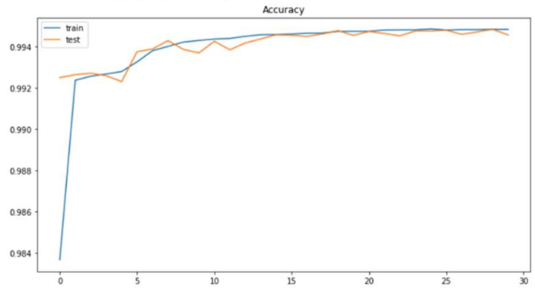
Number of Layers	Training Accuracy	Testing Accuracy
1	0.995	0.995



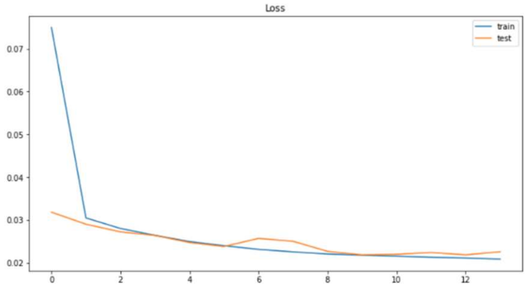


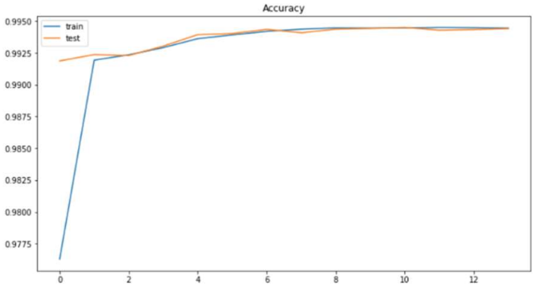
Number of Layers	Training Accuracy	Testing Accuracy
2	0.995	0.994



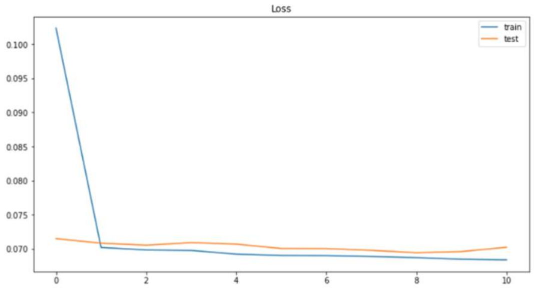


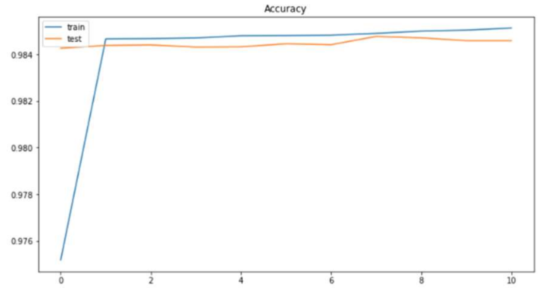
Number of Layers	Training Accuracy	Testing Accuracy
3	0.99457	0.99441





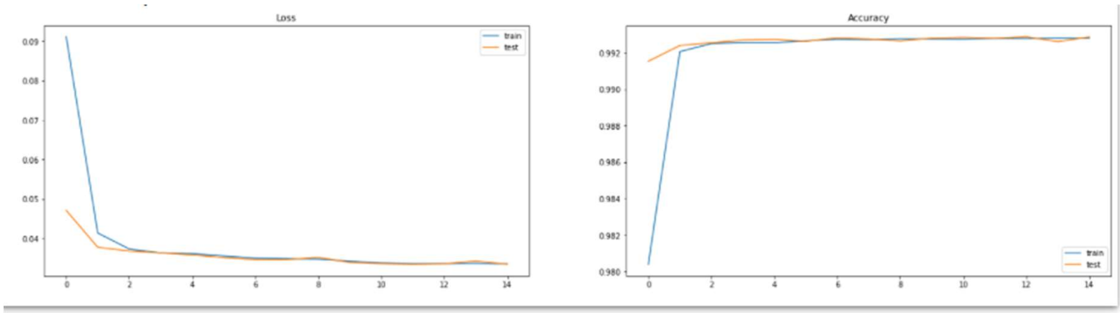
Number of Layers	Training Accuracy	Testing Accuracy
4	0.98499	0.98459





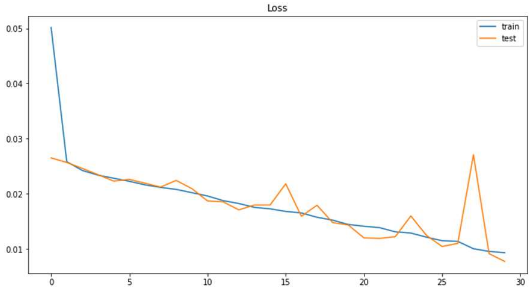
PCA CONFIGERATION

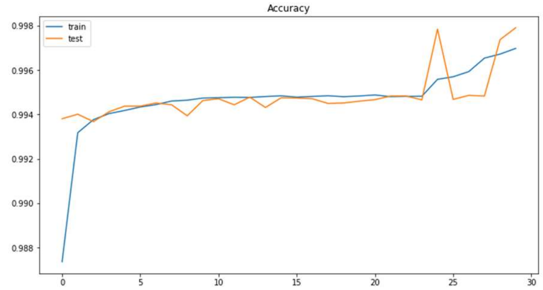
PCA	Training Accuracy	Testing Accuracy
20	0.99286	0.99286



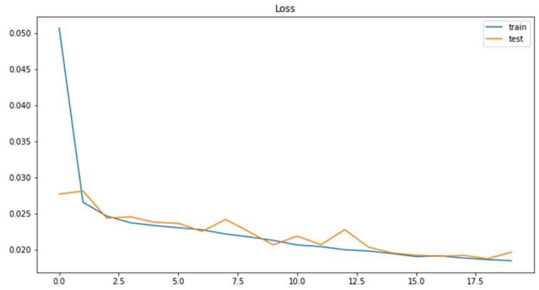
Higher Configuration

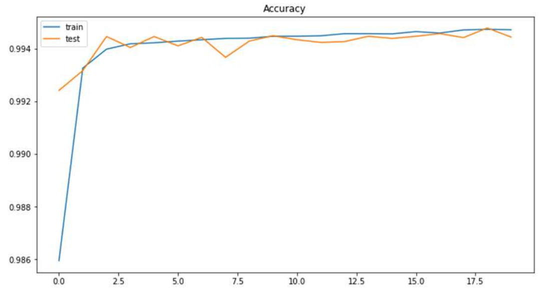
Number of layers	Number of neurons	Train accuracy	Test accuracy	PCA dimensions
3	16,16,16	0.99816	0.99790	20



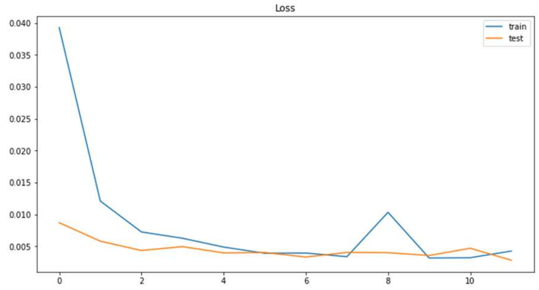


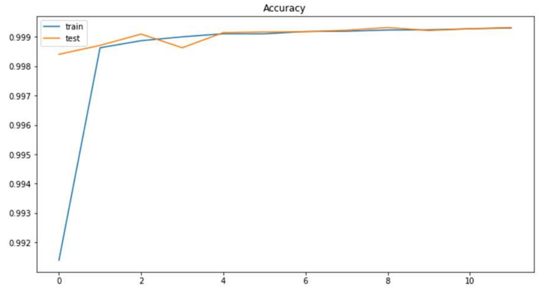
Number of layers	Number of neurons	Train accuracy	Test accuracy	PCA dimensions
3	16,64,16	0.99816	0.99790	20



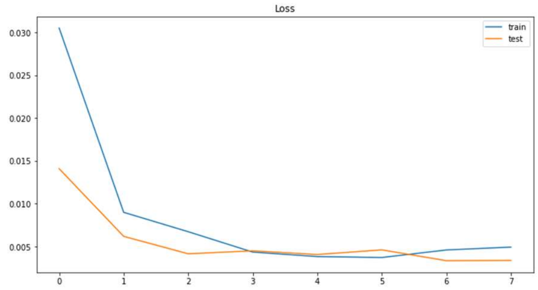


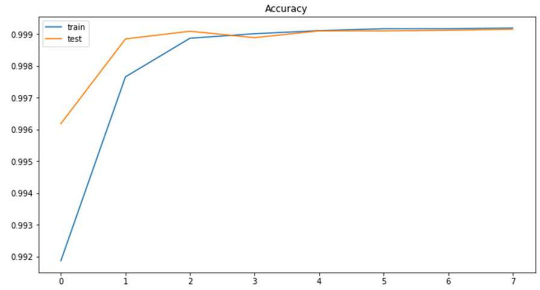
Number of layers	Number of neurons	Train accuracy	Test accuracy	PCA dimensions
3	16,32,16	0.9993	0.99931	Disabled



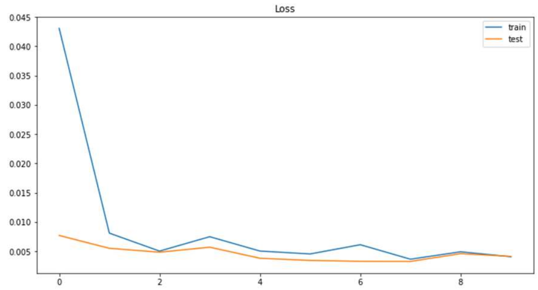


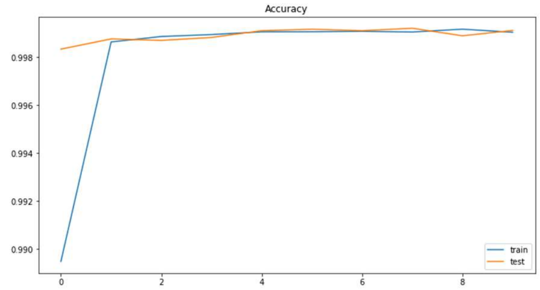
Number of layers	Number of neurons	Train accuracy	Test accuracy	PCA dimensions
3	16,64,16	0.9993	0.99931	Disabled





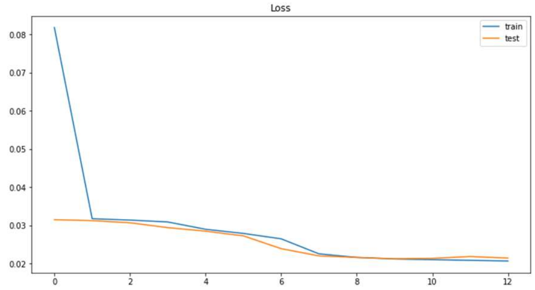
Number of layers	Number of neurons	Train accuracy	Test accuracy	PCA dimensions
4	16,32,16,5	0.99915, 0.99912	0.99931	Disabled

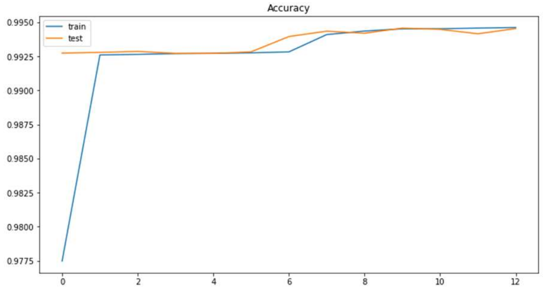




Applying same configuration to the Binary model

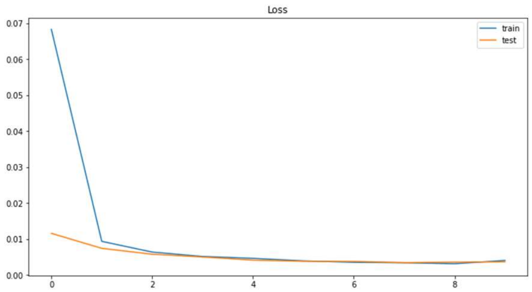
Number of layers	Number of neurons	Train accuracy	Test accuracy	PCA dimensions
4	16,64,16,5	0.99469	0.99454	Disabled

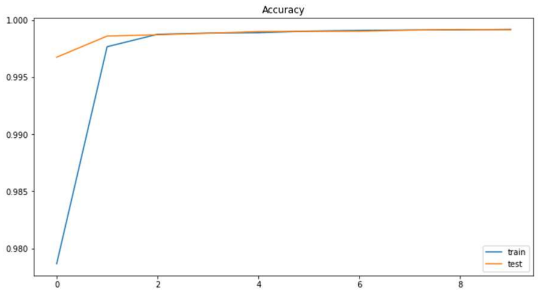




The final configerations

Number of layers	Number of neurons	Train accuracy	Test accuracy	PCA dimensions
4	16,64,16,5	0.99923	0.99914	Disabled


Loss
train
test


Accuracy
train
test