|  |  |
| --- | --- |
| **Executive Summary**  This report analysis the performance of Neural network based on the provided KDD cup dataset, the algorithm used here ANN, which contains dense layer of network where each neuron in the first layer is connected to each other neuron in the next layer. The language used here is python and libraries used is sklearn and keras.  The dataset used here carries 1048576 rows × 42 columns, rows and columns filled with numerical values, binary values, and categorical data. The computer cannot process string values, so categorical data where first converted into numerical data using panda’s library which replace categorical data with its index value. The second step is to drop duplicates, dataset roughly carries some duplicate values which means some rows and columns will be more likely to each other. Then input features and label were separated to different variable name, input features carry information about list of protocols, services, etc used by the network and label set carries last columns which is the type of attack happened. Based on this input feature the neural network is going to predict the labels. The labels set carries different types of attacks, though it is multiclass function, but here we converted multiclass to binary by keeping no attacks as “normal” and attacks as ‘attack’, hence problem becomes binary. And label encoder is enabled to convert this label values into binary values. The dataset was split into 70% for training and 30% for testing  Model used here was ANN, which carries 1 input layer, 3 hidden dense layers and 1 output layer. Before sending the input features to model, normalization of input set was taken place to eliminate the outliers, outliers reduce the system performance because it does not have any relation to the input features, and normalization helps to maintain correlation between input features and eliminate duplicates. 1st hidden layer was set to be 128 hidden neuron and 2nd layer contains 64, and 3rd layer carries 32 neurons. Each layer relates to activation function at the end called ReLu activation. The network carrying dense network, while doing backpropagation which means weight updates in each neuron to minimise the loss. Currently relu was good in handling vanishing gradient problems and introduce non-linearity to the input features, this non-linearity helps the neurons to differentiate the loss function while in backpropagation stage. Backpropagation was done using Adam optimiser. Thus, output layer was connected with sigmoid activation function, because its performance was good in binary classification problems. This function converts the values from the hidden layers between 0 and 1.  The model performance was calculated by 99.82% accuracy, 0,0153 loss, 99.81% precision and 99.8% recall for training set and testing performance was calculated more accurately using confusion matrix. Which shows out of 184335 labels, 320 true negatives and 17 false negatives are only misclassified. Thus, system learned the training set without overfitting, the precautions were made in the training phase to prevent overfitting using validation set which is 20% taken from training dataset.  This report explains the working architecture of ANN and shows the most possible promising results. Further research needed to test the model with large datasets and multiclassification problems. | **Subject**  **Data Pre-processing**  **Model Description**  **Performance Analysis**      **Conclusion** |