Ghulam Ishaq Khan Institute
of Engineering Sciences and
Technology

Kareema Batool - 2018174

M. Junaid Javed Shah - 2018293

Project Report

(CS351L)

15th May 2021

1. Abstract

An ensemble learning model was developed that contains five different ANN models to classify network attacks based on various hyperparameters provided in the dataset. The accuracy was compared with other prediction techniques such as Decision Trees and Random Forest. The data was then clustered using K Means to label it and then passed onto the ensemble learning model.

2. Introduction

Network attacks are a common occurrence in a growing digital world. Since all of our devices are now connected to one another, they provide the convenience of sharing data between them. However, that has allowed experts to exploit devices on the network. There are several different types of network attacks and the purpose of this project is to classify the type of network attack based on various features of the attack.

3. Data Pre-processing

The pre-processing on the Dataset provided was done in three stages.

3.1 Mapping Attack Types

The original dataset had 27 different attack types which needed to be mapped onto 5 different classes from the attack_types file.

3.2 One Hot Encoding

Since some of the features had categorical data, we needed to encode those integers using one hot encoding so the model would properly understand that they are categorical and not numerical data.

3.3 T-SNE

In the final task, the given dataset was projected using T-SNE to reduce its dimensions to 2 and make it fit for clustering using K means.

4. Feature Engineering:

The dataset provided to us does not contain columns with null, missing or garbage values. Hence, we have used the dataset provided as it is without any feature engineering.

5. Use of All Classification Algorithms:

5.1 ANN Models:

In our model, we have trained 5 different ANN models by slightly varying the hyperparameters. For each of this model, we have computed the validation and training accuracies.

5.1.1 Hyperparameters:

5.1.1.1 Number of hidden layers and number of neurons in each hidden layer:

To make variants of ANN models and also to tune the number of layers we have made our models with a variety of number of layers and number of neurons.

5.1.1.2 Loss Function:

The loss function that we have used here is the “categorical_crossentropy”. This loss outperforms all other loss functions in its simplicity. Also, this loss performs better while used with one-hot encoded labels.

5.1.1.3 Optimizers:

In our project, we have used different optimizers for each of our models. The optimizers used are Adam, Adamax, RMSprop and SGD.

5.1.1.4 Dropout Layers:

Other than traditional ANN layers, we have also used Dropout layers for models with higher number of neurons. These dropout layers prevent the overfitting by making the training procedure more generalized.

5.1.1.5 Initializer:

For two of our models, we have used the “Lecun” initializer to prevent the degradation problem in backpropagation process.

5.1.2 Ensembled Learning:

After training and saving 5 different variants of ANN models, an ensembled prediction is generated based on the majority vote of contributing models. Moreover, we have used Hard Voting Ensemble which predicts the class with the largest sum of votes from models.

5.2 Decision Tree Classifier:

To have a higher prediction accuracy, we have used Decision Tree Classifier with the criterion of “entropy”. However, all other parameters are set as default. Here, the decision tree iteratively selects the attributes based on their information gain and makes them the nodes. The leaf nodes give the classified labels.

5.3 Random Forest Classifier:

The second classifier for comparison used was Random Forest Classifier. We have set our random forest criterion to “entropy” and the total number of trees is set to 50. This classifier provides an ensembled prediction of 50 decision trees.

5.4 K-means Clustering:

To implement unsupervised learning, we have used K-means clustering on our pre-processed dataset. Here, the total number of centroids are 5 equal to number of classes. After clustering, it classifies the dataset and returns a list of 5 labels. Afterwards, these are passed onto ANN models for classification and an ensembled prediction is returned. The results are shown in section 6.

6. Comparison and Performance Evaluation:

Performance metrics of all previously mentioned methods are given below.

6.1 ANN models Plots:

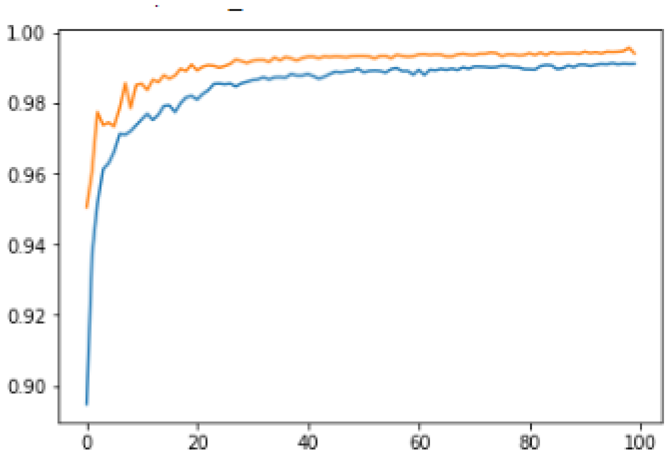


Fig.1 Model 1

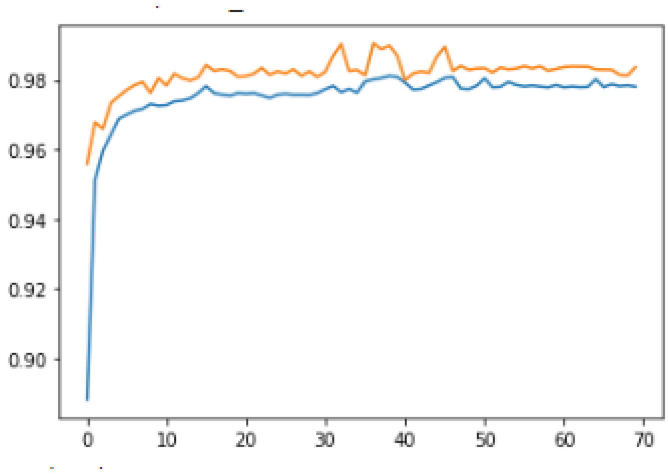


Fig.2 Model 2

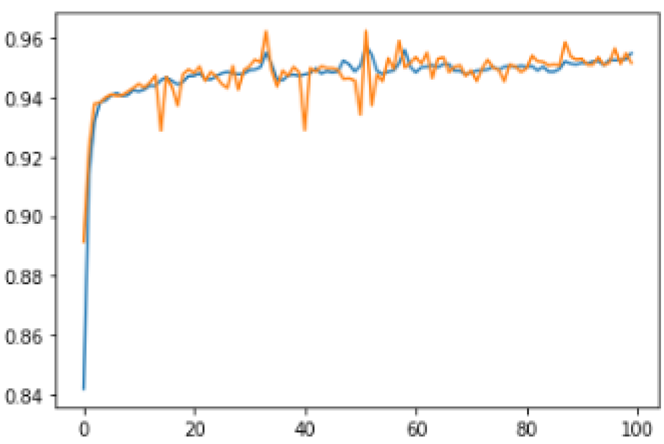


Fig 3. Model 3

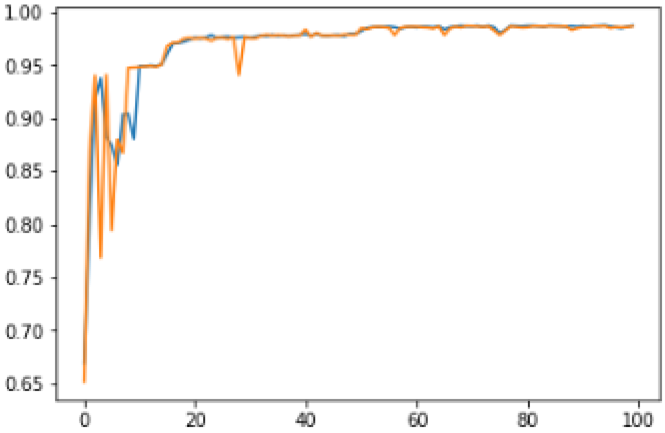


Fig 4. Model 4

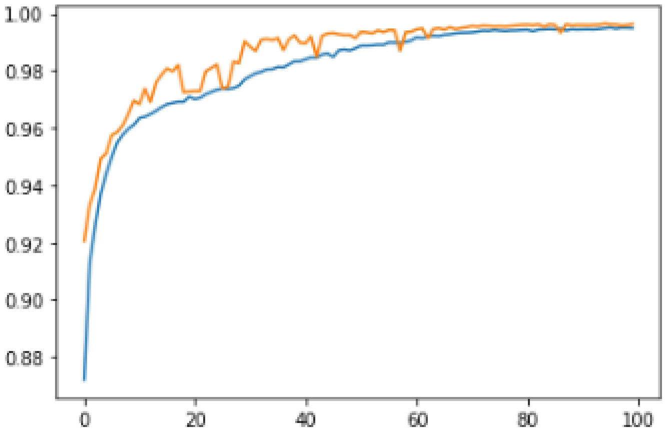
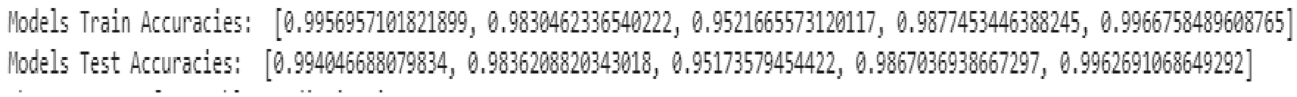


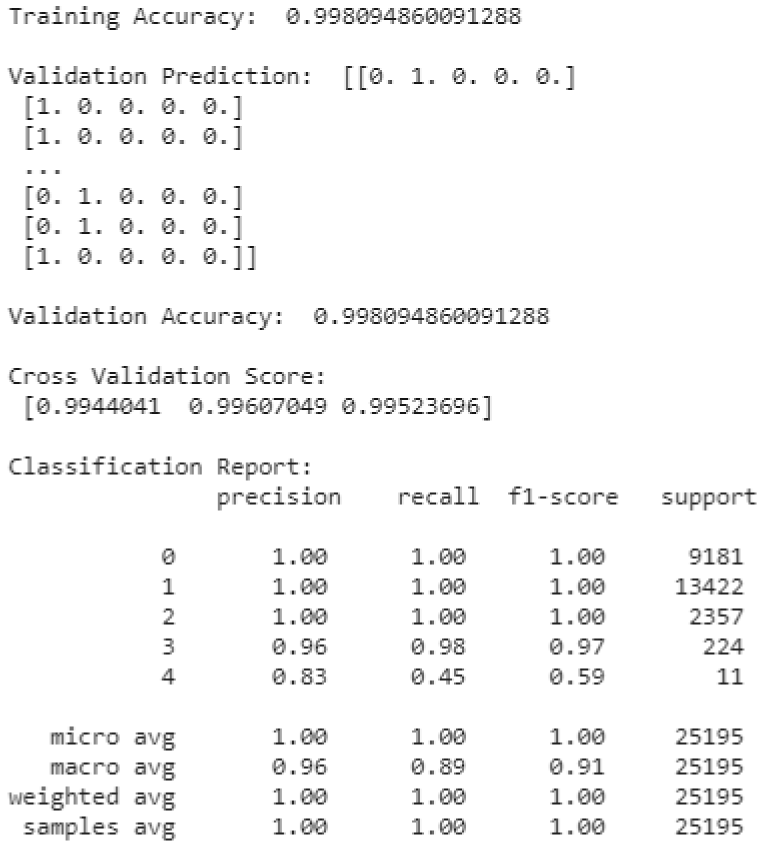
Fig 5. Model 5



```
Models Train Accuracies:  [0.9956957101821899, 0.9830462336540222, 0.9521665573120117, 0.9877453446388245, 0.9966758489608765]
Models Test Accuracies:  [0.994046688079834, 0.9836208820343018, 0.95173579454422, 0.9867036938667297, 0.9962691068649292]
```

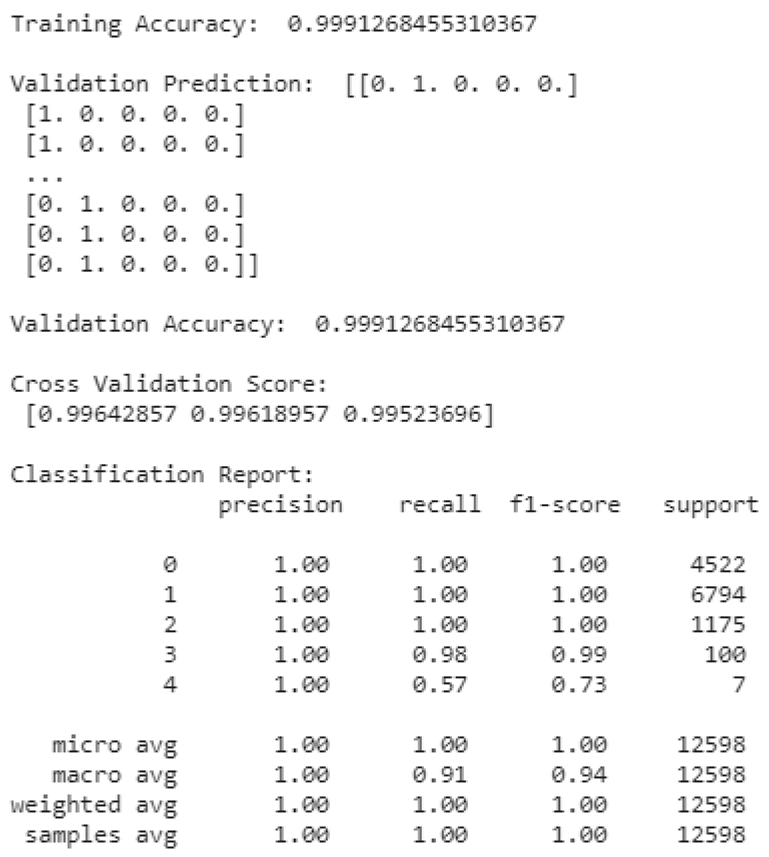
Ensemble Prediction Accuracy (Only ANN models) = 0.9898126587637596

6.2 Decision Tree Classifier:



```
Training Accuracy:  0.998094860091288

Validation Prediction:  [[0. 1. 0. 0. 0.]
 [1. 0. 0. 0. 0.]
 [1. 0. 0. 0. 0.]
 ...
 [0. 1. 0. 0. 0.]
 [0. 1. 0. 0. 0.]
 [1. 0. 0. 0. 0.]]

Validation Accuracy:  0.998094860091288

Cross Validation Score:
 [0.9944041  0.99607049 0.99523696]

Classification Report:
               precision    recall  f1-score   support

           0       1.00      1.00      1.00      9181
           1       1.00      1.00      1.00     13422
           2       1.00      1.00      1.00      2357
           3       0.96      0.98      0.97       224
           4       0.83      0.45      0.59        11

   micro avg       1.00      1.00      1.00     25195
   macro avg       0.96      0.89      0.91     25195
weighted avg       1.00      1.00      1.00     25195
 samples avg       1.00      1.00      1.00     25195
```

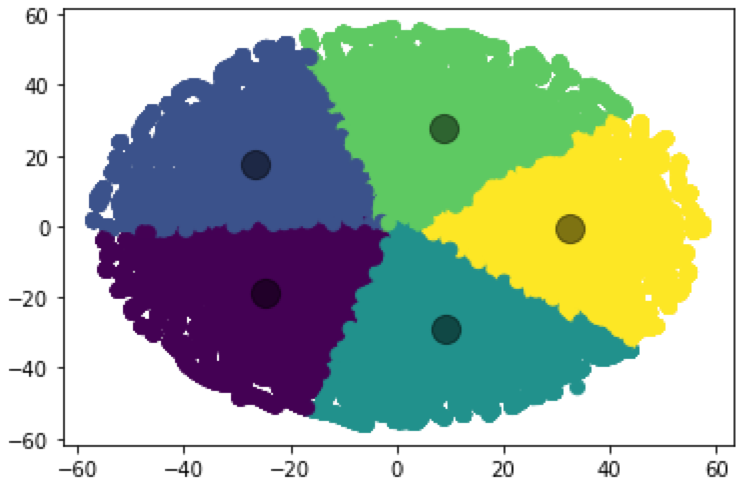
6.3 Random Forest Classifier:



```
Training Accuracy:  0.9991268455310367

Validation Prediction:  [[0. 1. 0. 0. 0.]
 [1. 0. 0. 0. 0.]
 [1. 0. 0. 0. 0.]
 ...
 [0. 1. 0. 0. 0.]
 [0. 1. 0. 0. 0.]
 [0. 1. 0. 0. 0.]]

Validation Accuracy:  0.9991268455310367

Cross Validation Score:
 [0.99642857 0.99618957 0.99523696]

Classification Report:
               precision    recall  f1-score   support

           0       1.00      1.00      1.00      4522
           1       1.00      1.00      1.00      6794
           2       1.00      1.00      1.00      1175
           3       1.00      0.98      0.99       100
           4       1.00      0.57      0.73         7

   micro avg       1.00      1.00      1.00     12598
   macro avg       1.00      0.91      0.94     12598
weighted avg       1.00      1.00      1.00     12598
 samples avg       1.00      1.00      1.00     12598
```

Ensemble Prediction Accuracy (5 ANNs, decision tree, random forest) = 0.9969836481981267

6.4 K- Means Clustering:



Cluster Accuracy = 0.6372556023909886

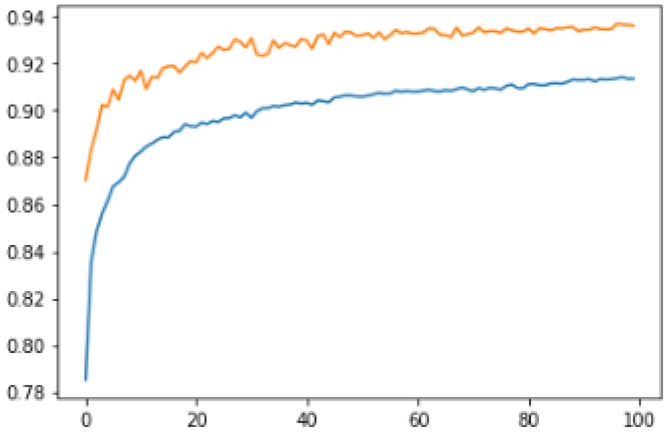


Fig 6. Model 1 (clustered labels)

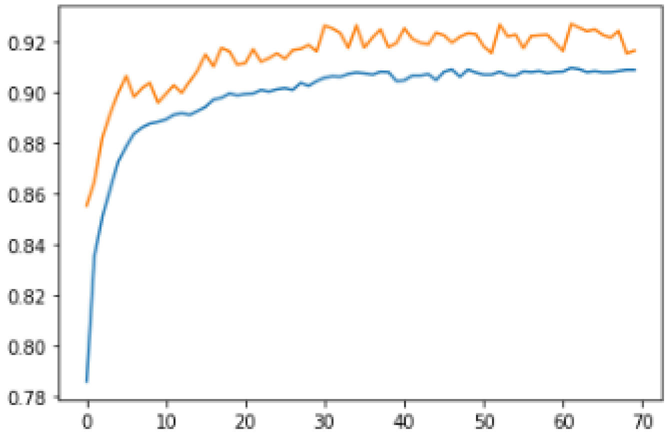


Fig 7. Model 2 (clustered labels)

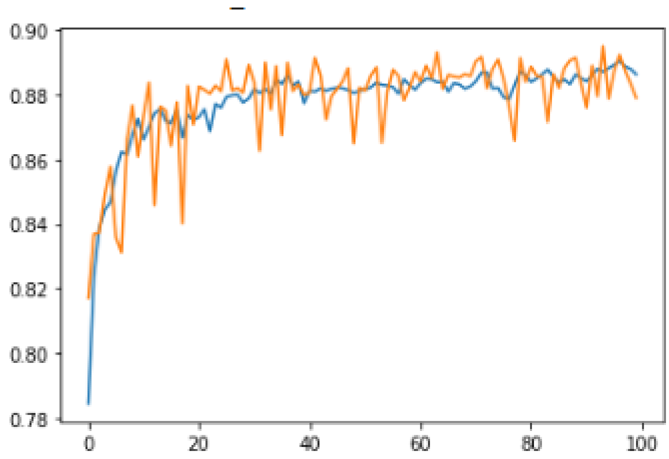


Fig 8. Model 3 (Clustered labels)

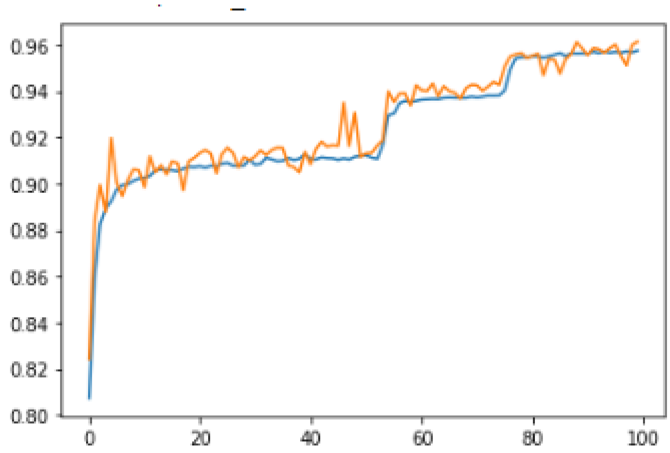


Fig 9. Model 4 (Clustered labels)

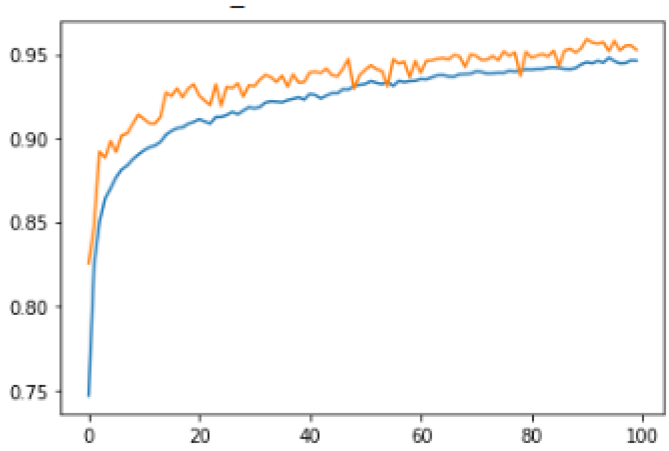
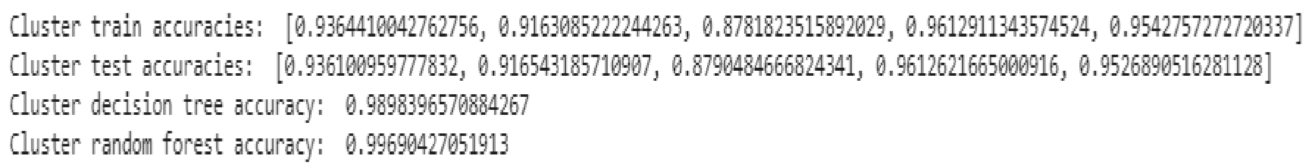


Fig 10. Model 5 (Clustered labels)



```
Cluster train accuracies:  [0.9364410042762756, 0.9163085222244263, 0.8781823515892029, 0.9612911343574524, 0.9542757272720337]
Cluster test accuracies:  [0.936100959777832, 0.916543185710907, 0.8790484666824341, 0.9612621665000916, 0.9526890516281128]
Cluster decision tree accuracy:  0.9898396570884267
Cluster random forest accuracy:  0.99690427051913
```

Ensemble Prediction Accuracy = 0.98777583743451 34

7. Conclusion

The