Report
For
CS351 Lab Semester Project

Group Details:

Registration Number	Name
2018271	Hanzla Javaid
2018352	M. Usman Zeb

Supervisor:
Sir Usman Haider

TABLE OF CONTENTS

LIST OF FIGURES

1. Abstract

This report will cover the different techniques and models we implemented in classifying the various cyber-attacks that occur within the network traffic. The dataset was instantiated as a pandas data frame with 23 label classes being reduced to 5 classes in the pre-processing stage for the first task while dropping and re-labelling the dataset for the clustering task. Feature engineering was also carried out during this process to reduce the dimension of the dataset matrix to remove any outliers. An ensemble model was developed and used for the classification and line graphs were being plotted for performance evaluation. Furthermore, the ensemble model had five artificial neural networks (ANNs) with different parameters which were utilized as base models that were first trained and tested independently and then ensembled by adding a Support Vector Machine (SVM).

2. Introduction

Network security is one of the most essential components in the era of the Internet that network experts and users alike must take care of to keep their data secure or prevent any sort of denial-of-service. Identifying the type of attack used by an attacker is the first step to prevent it from happening in the future. As today we have the privilege of having access to large datasets and more computational power, it is a very viable solution to detect the different attacks through machine and deep learning.

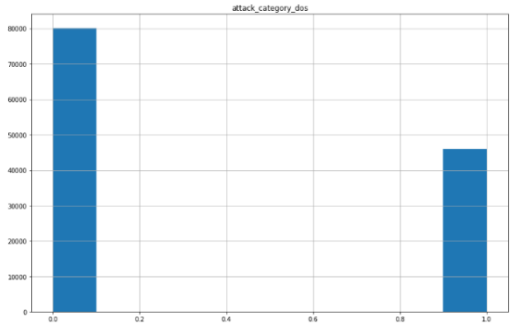
3. Data pre-processing

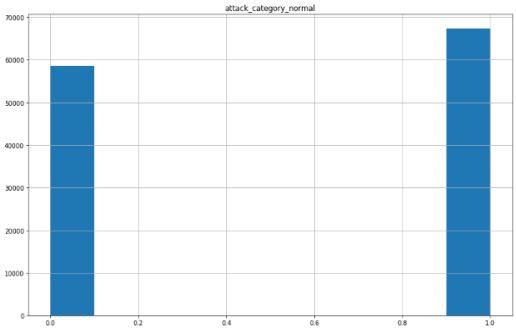
Data pre-processing was the initial step to cleanse the dataset and make it ready for the models. The dataset and the "Attack_types.txt" were imported by mounting our google drive. The dataset was initially given in a text file, so we had to convert it into a .csv (comma separated values) file using Microsoft Excel.

The dataset was then loaded as a pandas data frame. Next, the "Attack_types.txt" was loaded then split based on whitespaces and extracted into "attackTypes" list as value pairs. Furthermore, after feature engineering (discussed later), a for-loop was used to update the dataset's categories into 5 classes of attack types by using pandas' doc function. Next, we One-hot encoded the "attack_category" label as well as the "protocol_type" feature, since they were categorical, and we required numerical values in our dataset.

After extracting and making our dataset prepared, we ended with a matrix of 125973x27. Furthermore, in clustering, we dropped the label columns from the dataset from previous parts. Before training, the dataset was split into 25% of test data, and the rest being used for training.

Below are some histograms visualizing the dataset:





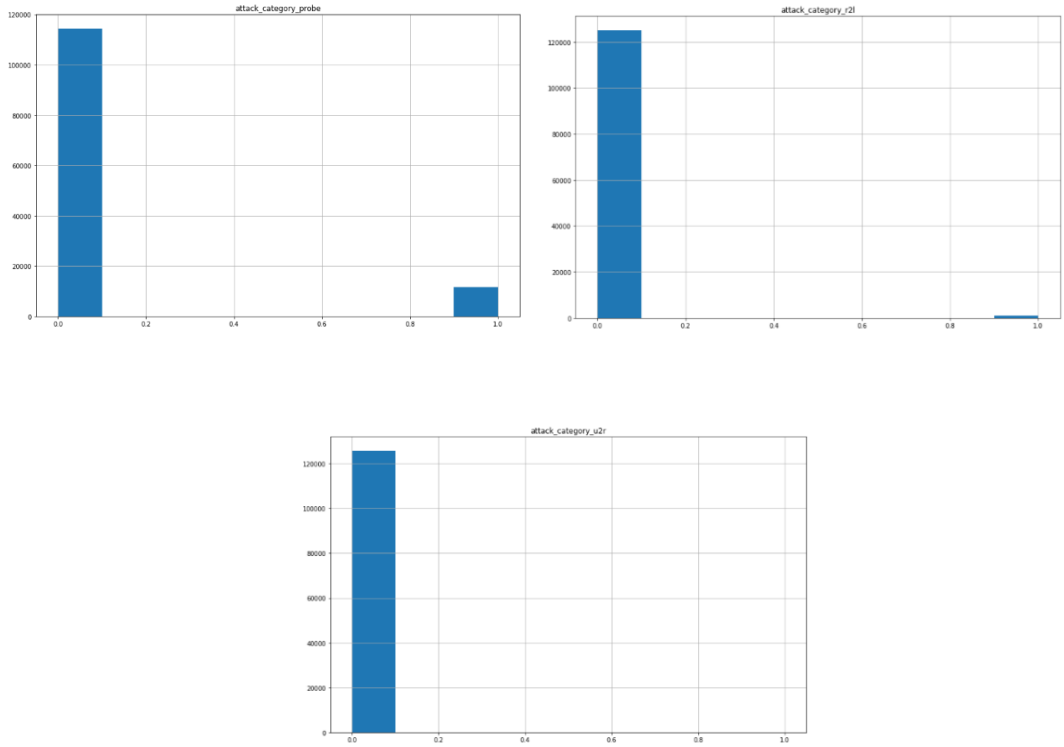


Fig 3.1 – 3.5 Dataset visualization

4. Feature Engineering

Feature engineering was carried out during the pre-processing stage to reduce the dataset matrix dimensions. It was based on the observation of columns having low mean (indicating low frequency), and columns having sparse classes with disproportionate data i.e., service, flag, etc. A total of 23 columns were dropped from the raw dataset based on the above observations.

Fig 4.1 Raw dataset heatmap snippet

5. Classification and Clustering Algorithms

5.1. Ensemble Learning

Ensemble Learning is a technique where multiple base models are first independently trained, saved, and then concatenated together to produce the final classification for improved performance in some metric like accuracy. We made an ensemble model using five different variants of Artificial Neural Network (ANN), and utilized it for both the tasks of supervised learning and the clustered dataset.

After training the ANN models independently and loading them into “models_list” list, the “create_ensembled_model” function sets the already trained parameters/layers of the models to un-trainable, concatenates the output layers, and adds an output SVM classifier for the prediction.

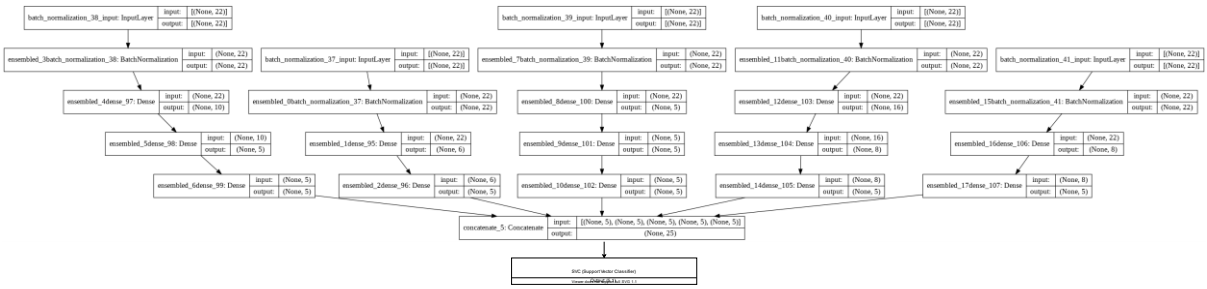


Fig 5.1.1 Ensemble model architecture

5.2. Artificial Neural Network (ANN)

As mentioned earlier, five different variants of ANN were used as base models in the ensemble classifier. Here we will discuss, how each of the ANN differed in architecture. Each ANN performed early stopping implemented using callbacks and used the same SoftMax activation function for output layer, ReLU for other layers, cross-entropy loss, Adam optimizer, and batch normalization. Furthermore, the changeable parameters were stored in the lists of n_models, n_hidden, n_neurons, n_callback_monitors, n_patience, n_epochs.

Variant #	Hidden Layers	Neurons	Callback Monitor	Patience	Total Epochs
0	1	[6]	val_loss	2	5
1	2	[10, 5]	val_accuracy	3	5
2	2	[5, 5]	loss	2	5
3	2	[16, 8]	accuracy	1	5
4	1	[8]	loss	1	5

5.3. Support Vector Machine (SVM)

Support Vector Machine (SVM) was used in the ensemble model, after concatenating the output layers of the ANN and feeding that [none, 25] feature vector as an input to the SVM classifier. Furthermore, since SVM works on categorical labels, we had to drop the one-hot encoding labels and categorically label it during runtime. One-hot encoding was still necessary for training the independent ANNs, so it cannot be opted out during pre-processing.

5.4. K-means Clustering

In the last part, we had to commit clusters and then train the newly clustered dataset on the ANNs and ensemble models created previously. We used MinMax scaling to normalize our dataset as it is very essential for K-means to work properly. Then after dropping previous labels, we set total clusters = 5 and max iterations of 100. The dataset was then fitted, and the new labels were added into “predicted_unsupervised”. After appending the dataset with the new labels, one-hot encoding was performed again on the labels and resulted as label columns “attack_category_<insert number>”.

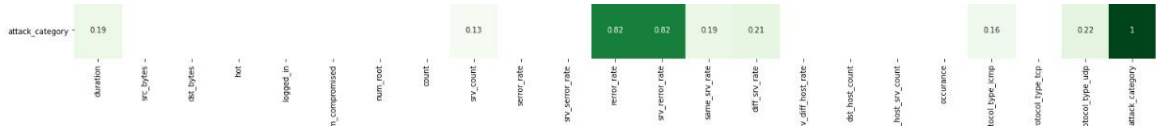


Fig 5.4.1 Clustered dataset heatmap snippet

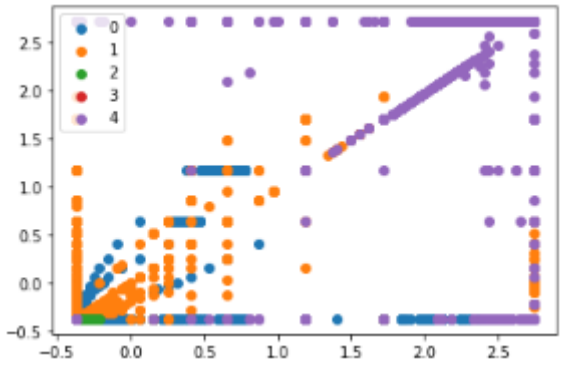


Fig 5.4.2 Clusters w.r.t rerror_rate(X-axis) and srv_rerror_rate(Y-axis)

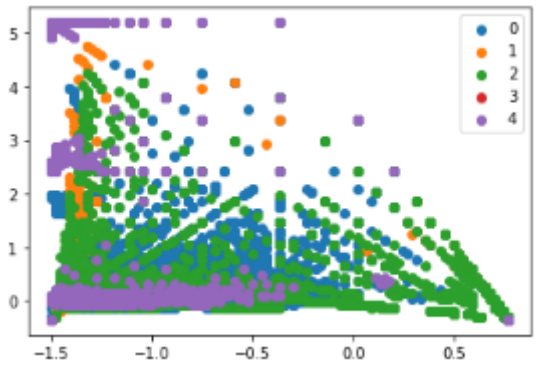


Fig 5.4.3 Clusters w.r.t same_srv_rate(X-axis) and diff_srv_rate(Y-axis)

6. Comparison and Performance Evaluation

Performance was being evaluated for both the tasks. The accuracy in each epoch of the ANN, as well as the average accuracy of the independent models, were contrasted with the ensemble model.

6.1. Supervised Learning Results

The labels were given for the dataset in this task, and the following were the accuracy results:

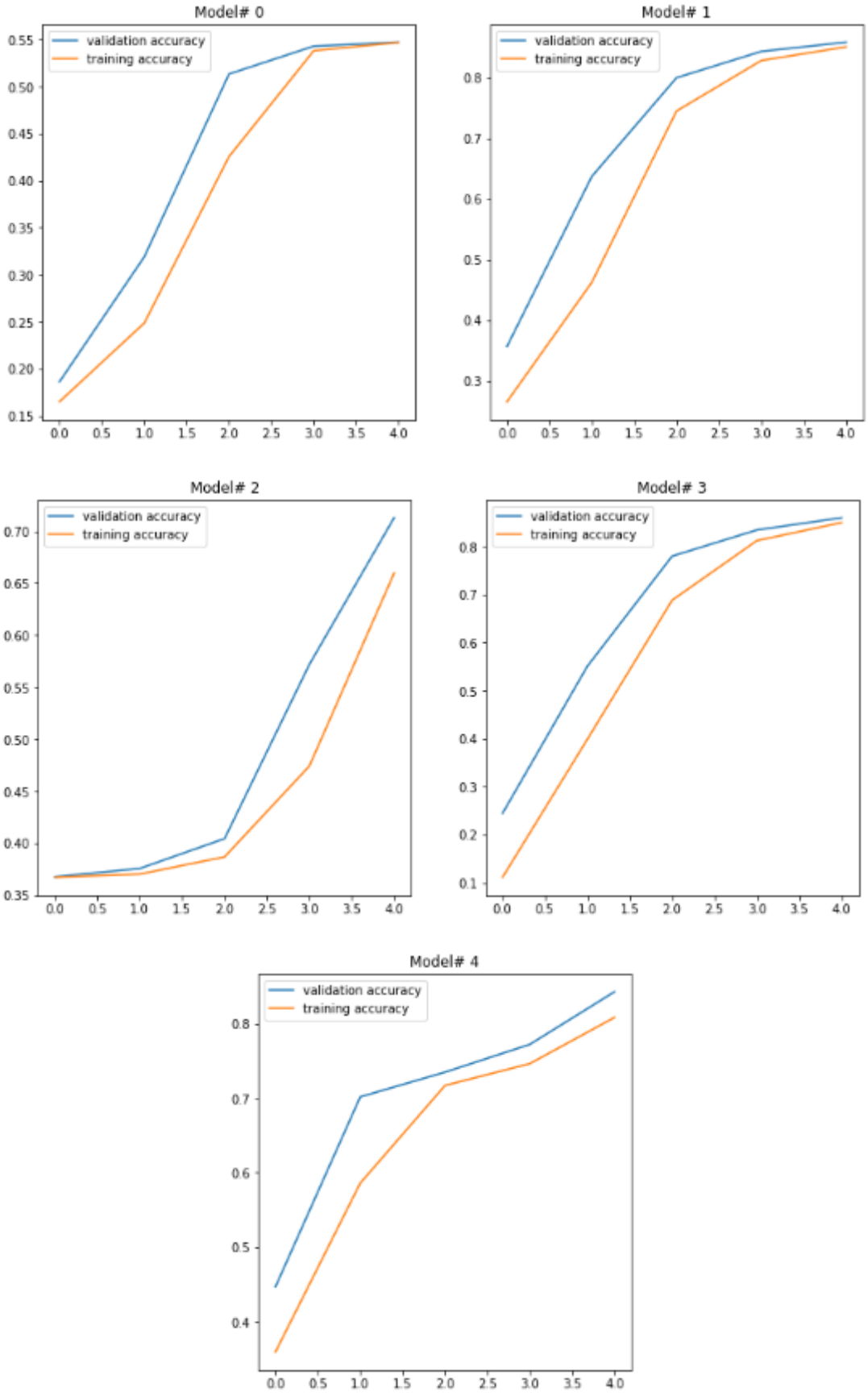


Fig 6.1.1 – 6.1.5 Training accuracy of base models during epochs

```
Evaluating model# 0
Loss: 1.0624526739120483 Accuracy: 0.5455642342567444
Evaluating model# 1
Loss: 0.9902317523956299 Accuracy: 0.8585444688796997
Evaluating model# 2
Loss: 1.0697047710418701 Accuracy: 0.7202006578445435
Evaluating model# 3
Loss: 0.9312862157821655 Accuracy: 0.860767126083374
Evaluating model# 4
Loss: 0.6426429152488708 Accuracy: 0.8396520018577576
```

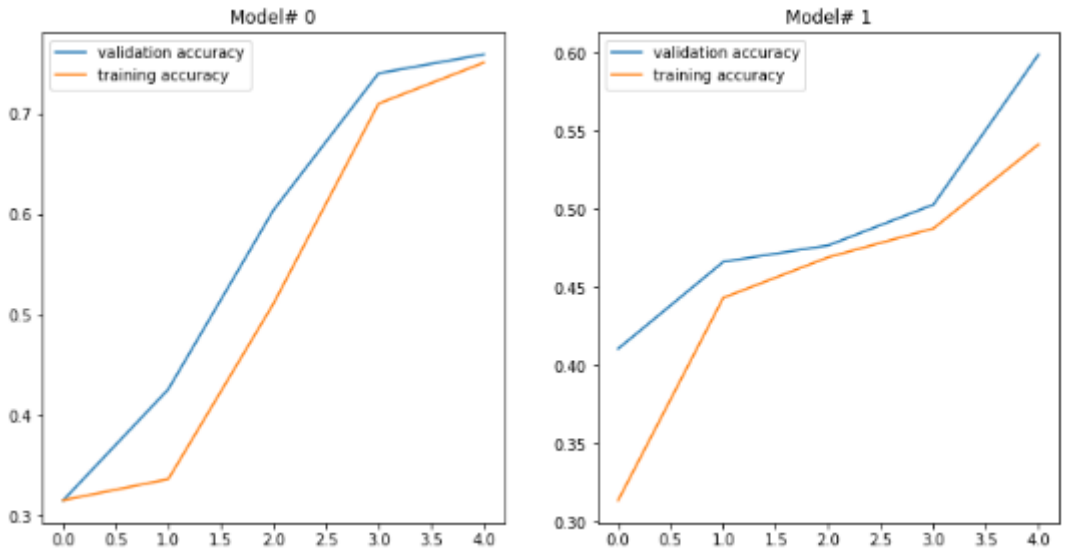
Fig 6.1.6 Test accuracy of base models

```
Ensemble model accuracy: 0.9845367371562838
```

Fig 6.1.7 Test accuracy of ensemble model

6.2. Clustered Dataset Results

The labels were learnt for the dataset in this task, and the following were the accuracy results:



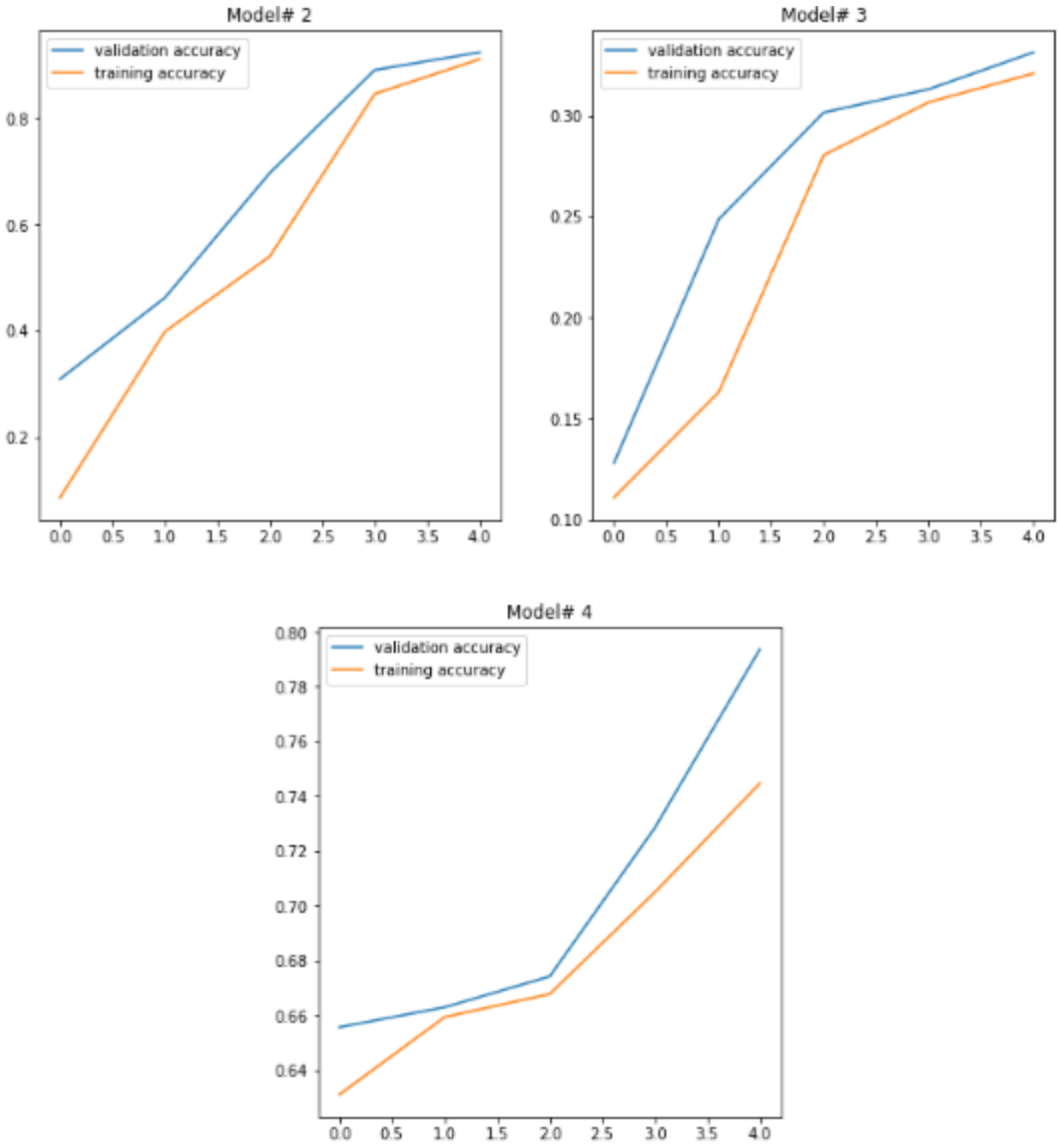


Fig 6.2.1 – 6.2.5 Training accuracy of independent models during epochs

```
Evaluating model# 0
Loss: 35.92356872558594 Accuracy: 0.10096423327922821
Evaluating model# 1
Loss: 38.70772171020508 Accuracy: 0.004921728745102882
Evaluating model# 2
Loss: 21.971418380737305 Accuracy: 0.7179902195930481
Evaluating model# 3
Loss: 28.131938934326172 Accuracy: 0.8635146617889404
Evaluating model# 4
Loss: 18.06977081298828 Accuracy: 0.9761322736740112
```

Fig 6.2.6 Test accuracy of base models

```
Ensemble model accuracy: 0.9985711564107449
```

Fig 6.2.7 Test accuracy of ensemble model

Task #	Base Model %Accuracy (Avg)	Ensemble Model %Accuracy
1	76.5	98.5
2	53.3	99.9

7. Conclusion

It was concluded that Ensemble learning increases the performance of the models drastically as shown by the results. Also, although the ensemble model showed a better accuracy using the clustered dataset, it will produce inaccurate results on a real-world dataset. It is because, if you compare both the heatmap snippets above, the clusters obtained are not based on features on which the real data is classified.