GHULAM ISHAQ KHAN INSTITUTE OF ENGINEERING SCIENCES AND TECHNOLOGY • PAKISTAN •

Submitted To :

Sir Usman Haider

Submitted By:

Syed Salman Shah 2018470

DATE:

May 15,2021

TABLE OF CONTENTS

Abstract:

This is a classification and a clustering task. The dataset contains different columns which can be used as features while classification. First, we preprocess the dataset because it has some categorical values. The preprocessing is done to convert these values to numerical values which are used for classification. Log scaling is done to reduce the range of some of the wide-spread values. We have built 3 ANN models by using different parameters. All the models returned an accuracy of about 99 percent. Then k-means clustering is performed to return the labels of the dataset.

Introduction:

The dataset has 43 features that are used as features to classify the types of attacks. The attack type file has 22 features which can then be classified into 5 types namely: DoS, U2R, R2L, Probe and Normal type. The attack types are first converted into these 5 classes. The dataset has some categorical values that must be converted to numerical values.

Preprocessing:

The dataset is not ready for the classification yet. Some preprocessing is done to prepare it for the task. Firstly, some attributes such as protocol_type, service and flag are categorical values. As we know that neural network cannot process categorical data and only work on numerical data. So, these columns are converted to numerical values. Furthermore, some attributes have very wide-spread values such as 'duration', 'src_bytes', 'dst_bytes' which affect the performance of the neural network. So, these attributes are scaled into a small range through log scaling. At the end we have used the MinMaxScaler function to minimize the range of all the values into the range of 0 and 1.

Artificial Neural Network:

The second task of our project is to use the ANN for the classification of the classification. We have used Keras to build simple ANN architectures. Three models are built for classification using different parameters. Early Stopping function is used in these models. This function stops the training when there is no improvement in the performance of the network over the course of few iterations. This prevents the overfitting of the model. The summary of the three models built is given in the following table:

Model	No. of Layers	Activation Functions	Batch Size	Validation Split	Epochs	Optimizer
Model 1	6	ReLu, SoftMax	128	0.1	10	Adam
Model 2	4	ReLu, SoftMax	128	0.2	15	Adam
Model 3	4	ReLu, SoftMax	64	0.1	20	Adam

Number of layers Neurons in Each Layer:

Model 1:

Layer	Number of Neurons
Layer 1	128
Layer 2	16
Layer 3	256
Layer 4	128
Layer 5	32
Layer 6	5

Model 2:

Layer	Number of Neurons
Layer 1	128
Layer 2	128
Layer 3	64
Layer 4	5

Model 3:

Layer	Number of Neurons
Layer 1	128
Layer 2	32
Layer 3	128
Layer 4	5

K-Means Clustering:

The third task of our project is using the k means clustering on this dataset. Firstly, we drop the attack_category column of the dataset because k means clustering is an unsupervised learning and in unsupervised learning, we do not provide labels to the dataset. We have used matplotlib and seaborn libraries for plotting the result and sklearn module of the scikit for implementing k means clustering. We chose 5 clusters because we have five classes in the dataset. All the other parameters of the built-in K-means function are used by default. The result of the clustering is plotted using matplotlib and seaborn. The predicted labels are stored in a variable and then added to the data frame.

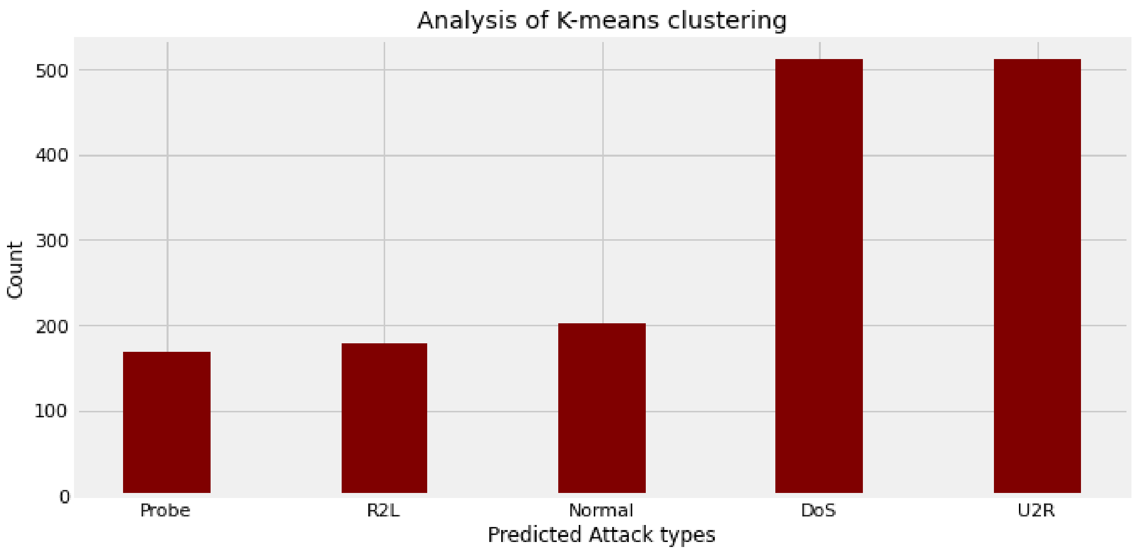
Performance Evaluation:

ANN:

The three models which we built; all returned an accuracy of around 99 percent. Their performance is summarized in the following table:

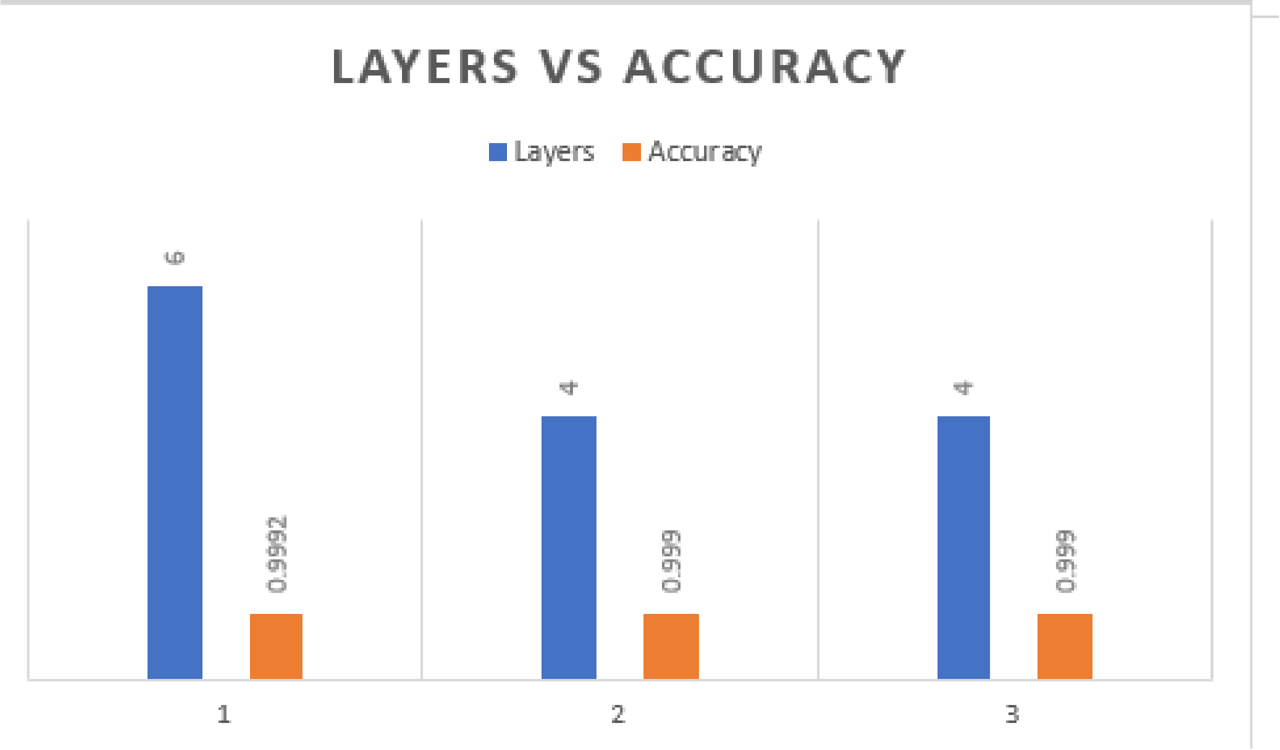
Model	Epochs Before Early Stopping	Accuracy
Model 1	9	0.9990
Model 2	11	0.9992
Model 3	9	0.9990

K Means:

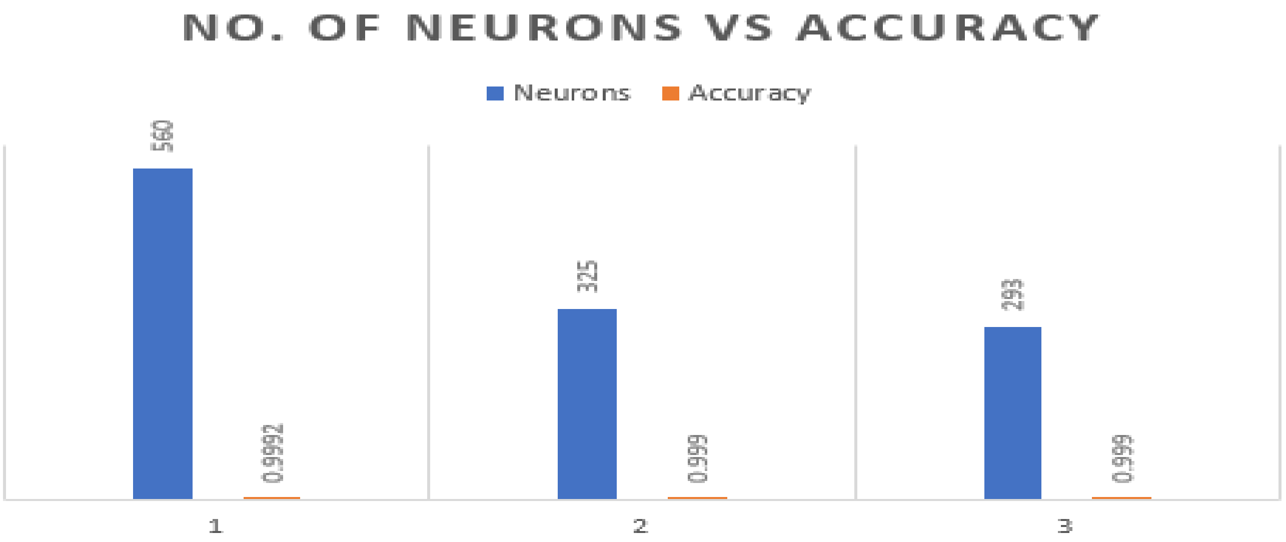


Analysis of Different Factors and Accuracy:

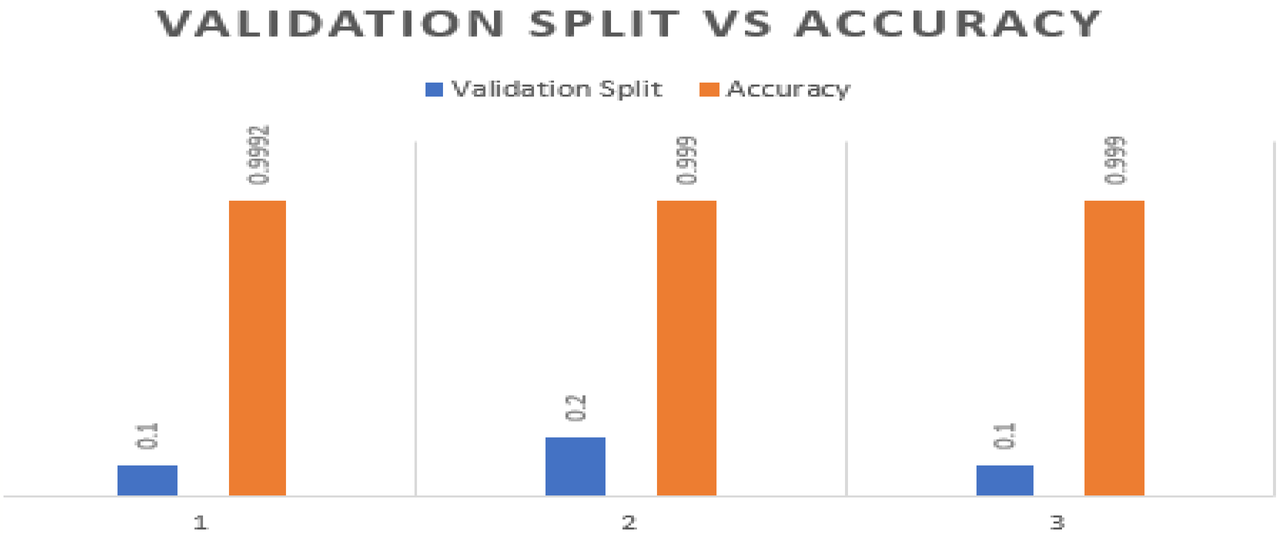
Number of Layers and Accuracy:



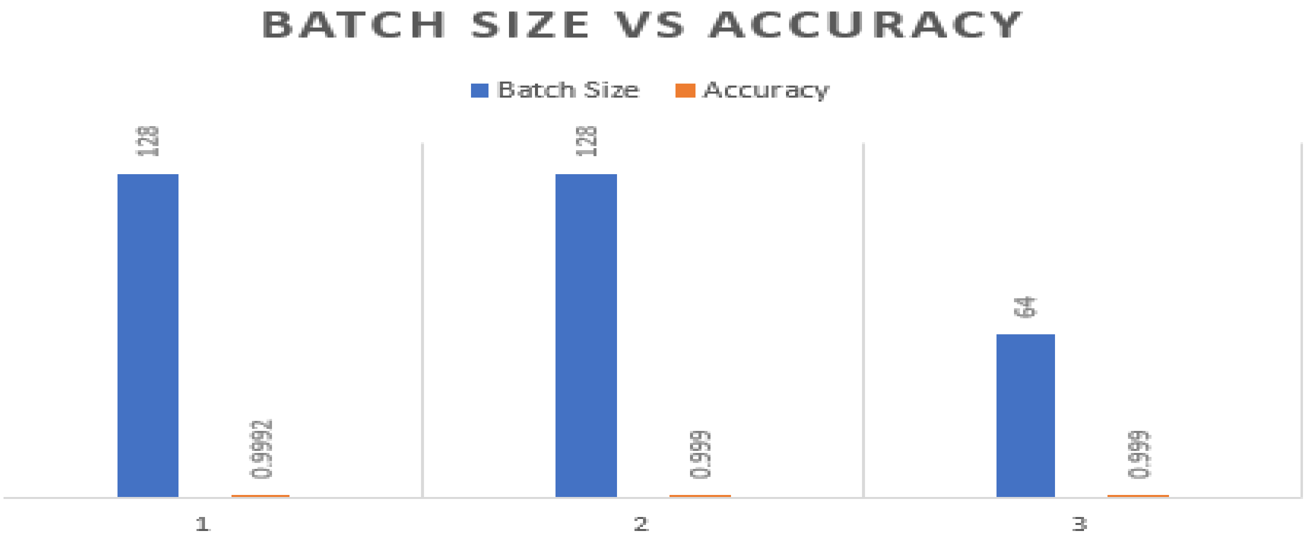
Number of Neurons and Accuracy:



Validation Split and Accuracy:

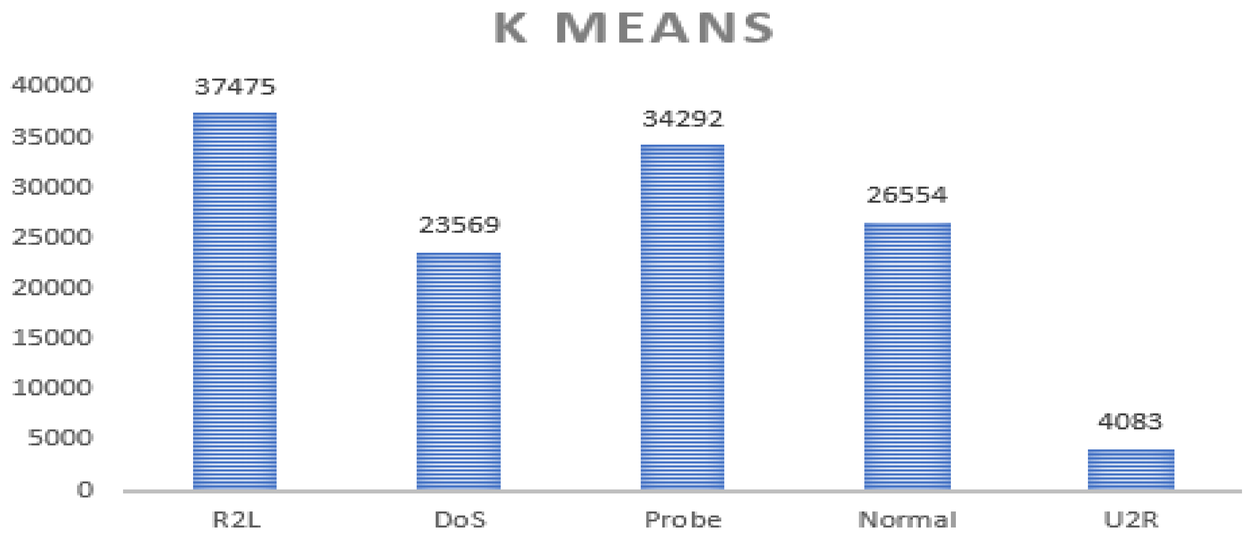


Batch Size and Accuracy:



Conclusion:

Looking at the above data, the accuracy has gone up when we increase the numbers of layers. The batch has no clear impact on the accuracy. Even at 128, the accuracy is variable in both models 1 and 20. Validation split has no clear impact on the accuracy as can be noted that in model 1 and 3, it is constant at 0.1 while accuracy has gone down. The number of neurons, however, have an impact on the accuracy. As we increase the number of neurons to 560, the accuracy goes up to 0.9992. In the clustering task, the summary is given in the following figure:



Link to the Presentation Video:

https://drive.google.com/file/d/1g7Z5hs8luC8WCF8FrefKa1L3OXhOOCaF/view?usp=sharing