Android is an open source free operating system and it has support from Google to publish android application on its Play Store. Anybody can developed an android app and publish on play store free of cost. This android feature attract cyber-criminals to developed and publish malware app on play store. If anybody install such malware app then it will steal information from phone and transfer to cyber-criminals or can give total phone control to criminal’s hand. To protect users from such app in this paper author is using machine learning algorithm to detect malware from mobile app.

To detect malware from app we need to extract all code from app using reverse engineering and then check whether app is doing any mischievous activity such as sending SMS or copying contact details without having proper permissions. If such activity given in code then we will detect that app as malicious app. In a single app there could be more than 100 permissions (examples of permissions are transact, API call signature, onServiceConnected, API call signature, bindService, API call signature, attachInterface, API call signature, ServiceConnection, API call signature, android.os.Binder, API call signature, SEND\_SMS, Manifest Permission, Ljava.lang.Class.getCanonicalName, API call signature etc.) which we need to extract from code and then generate a features dataset, if app has proper permission then we will put value 1 in the features data and if not then we will value 0. Based on those features dataset app will be mark as malware or good ware.

Once we got dataset then we will build machine learning SVM training model on that dataset and upon receiving new app’s features we will apply new app’s features on train model to predict whether app is malware or good ware.

In this paper author is using two machine learning algorithms such as SVM (Support Vector Machine) and NN (Neural Networks). App will contains more than 100 features and machine learning will take more time to build model so we need to optimized (reduce dataset columns size) features, to optimized features author is using genetic algorithm. Genetic algorithm will choose important features from dataset to train model and remove un-important features. Due to this process dataset size will be reduced and training model will be generated faster. In this paper comparison we are losing some accuracy after applying genetic algorithm but we are able to reduce model training execution time.

Steps involve in genetic algorithm to choose important attributes.

Step 1: Initialize the algorithm using feature subsets which are binary encoded such that if the feature is included it is represented by 1 and if it is excluded it is represented by 0 in the chromosome.

Step 2: Start the algorithm defining an initial set of population generated randomly.

Step 3: Assign a fitness score calculated by the defined fitness function for genetic algorithm.

Step 4: Selection of Parents: Chromosomes with good fitness scores are given preference over others to produce next generation of off-springs.

Step 5: Perform crossover and mutation operations on the selected parents with the given probability of crossover and mutation for generation of off-springs.

Repeat the Steps 3 to 5 iteratively till the convergence is met and fittest chromosome from population, that is, the optimal feature subset is resulted.

SVM Working Procedure

Machine learning involves predicting and classifying data and to do so we employ various machine learning algorithms according to the dataset. SVM or Support Vector Machine is a linear model for classification and regression problems. It can solve linear and non-linear problems and work well for many practical problems. The idea of SVM is simple: The algorithm creates a line or a hyperplane which separates the data into classes. In machine learning, the radial basis function kernel, or RBF kernel, is a popular kernel function used in various kernelized learning algorithms. In particular, it is commonly used in support vector machine classification. As a simple example, for a classification task with only two features (like the image above), you can think of a hyperplane as a line that linearly separates and classifies a set of data.

Intuitively, the further from the hyperplane our data points lie, the more confident we are that they have been correctly classified. We therefore want our data points to be as far away from the hyperplane as possible, while still being on the correct side of it.

So when new testing data is added, whatever side of the hyperplane it lands will decide the class that we assign to it.

How do we find the right hyperplane?

Or, in other words, how do we best segregate the two classes within the data?

The distance between the hyperplane and the nearest data point from either set is known as the margin. The goal is to choose a hyperplane with the greatest possible margin between the hyperplane and any point within the training set, giving a greater chance of new data being classified correctly.

A neural network (NN) is a computational model based on the structure and functions of biological neural networks. Information that flows through the network affects the structure of the NN because a neural network changes - or learns, in a sense - based on that input and output.

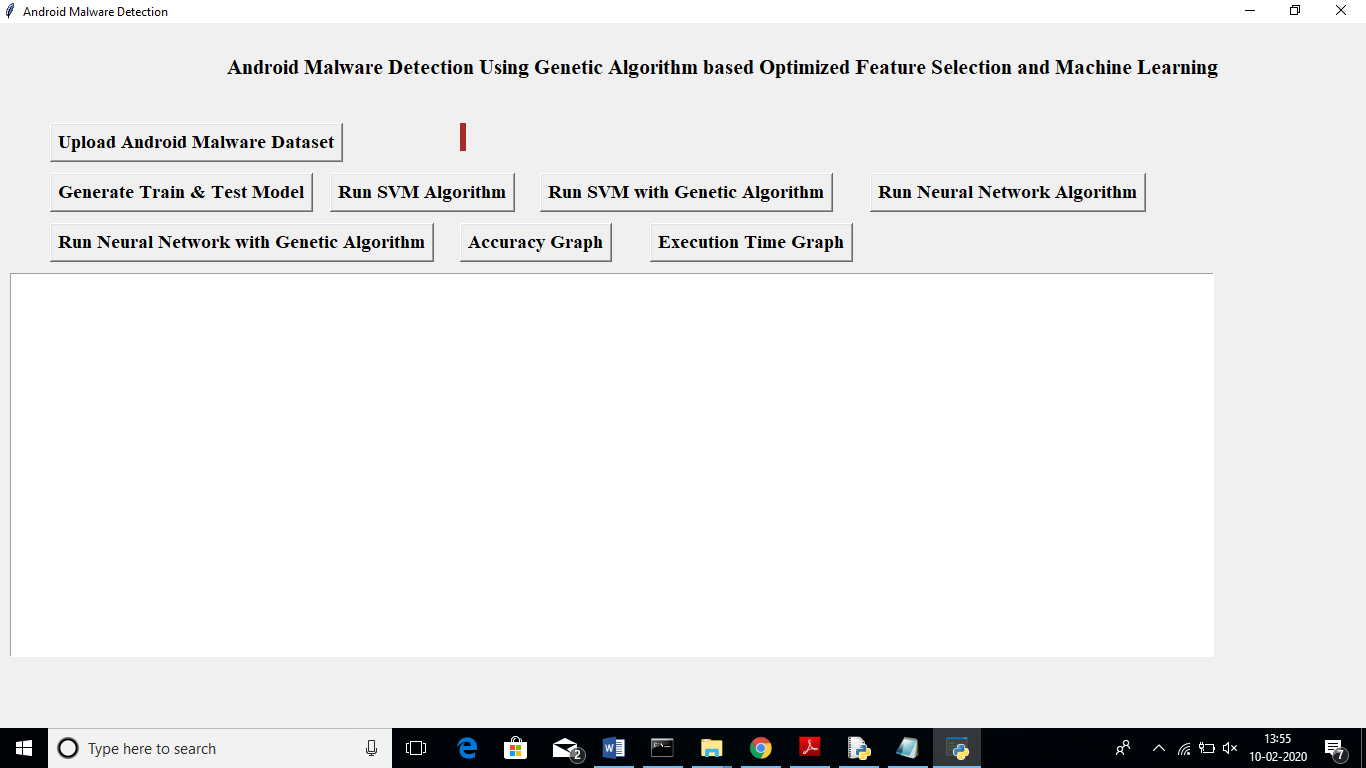
NNs are considered nonlinear statistical data modelling tools where the complex relationships between inputs and outputs are modelled or patterns are found.

NN is also known as a neural network.

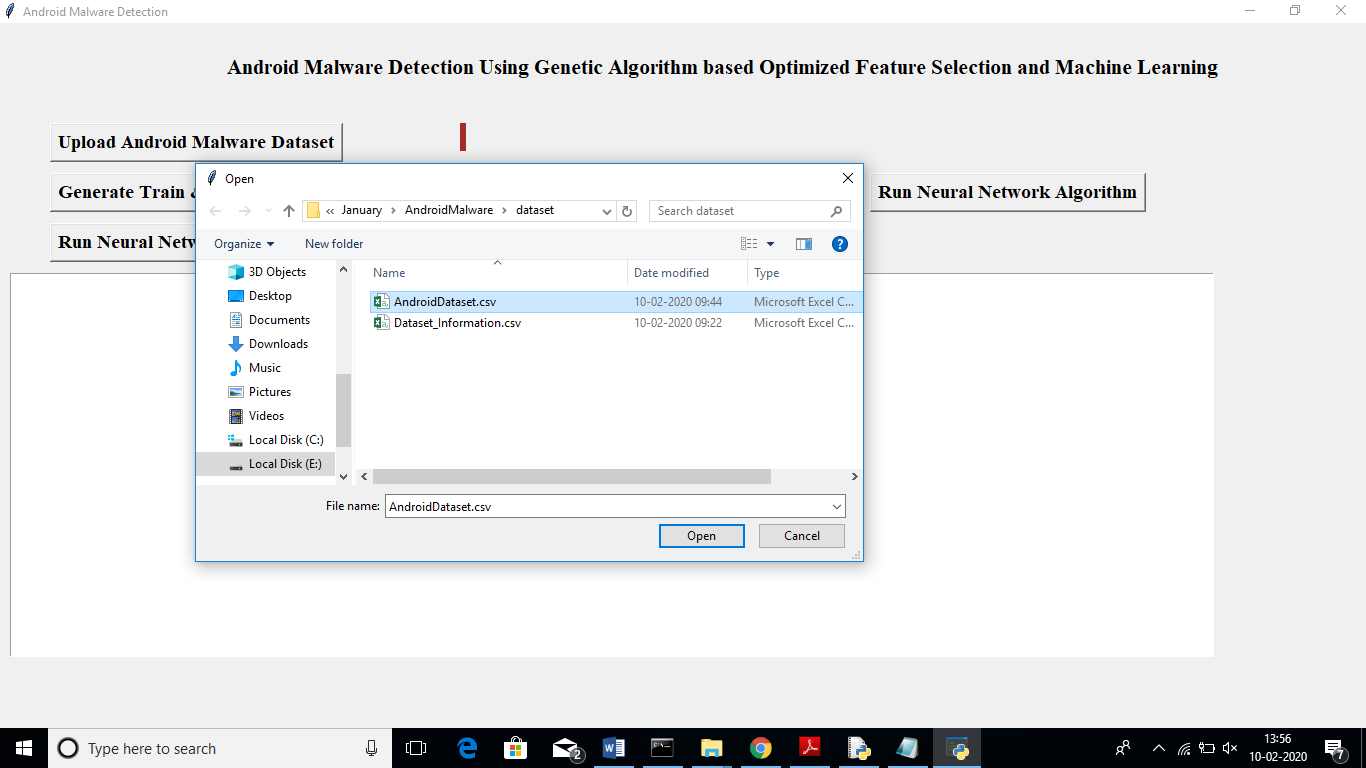
An NN has several advantages but one of the most recognized of these is the fact that it can actually learn from observing data sets. In this way, NN is used as a random function approximation tool. These types of tools help estimate the most cost-effective and ideal methods for arriving at solutions while defining computing functions or distributions. NN takes data samples rather than entire data sets to arrive at solutions, which saves both time and money. NNs are considered fairly simple mathematical models to enhance existing data analysis technologies.

NNs have three layers that are interconnected. The first layer consists of input neurons. Those neurons send data on to the second layer, which in turn sends the output neurons to the third layer.

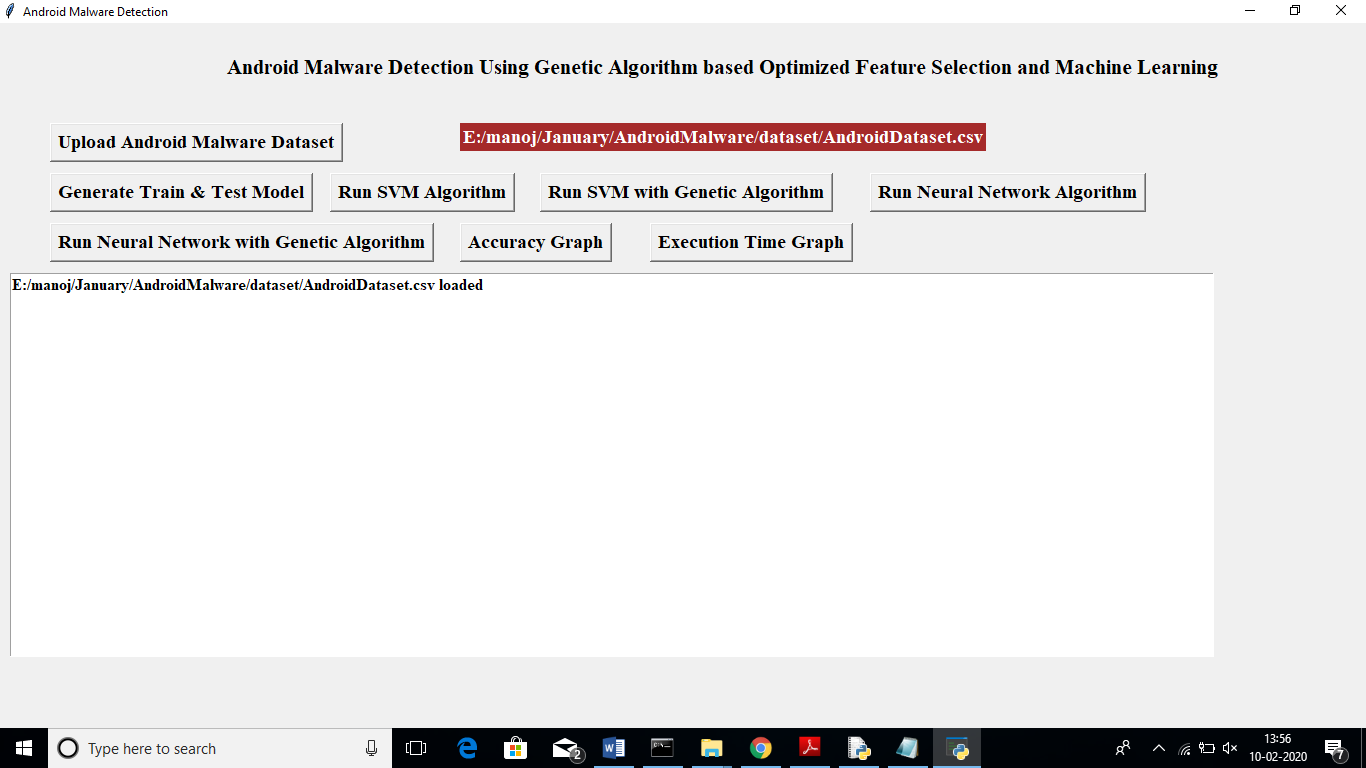
We downloaded android malware dataset from internet and it’s saved inside ‘dataset’ folder. To run this project double click on ‘run.bat’ file to get below screen



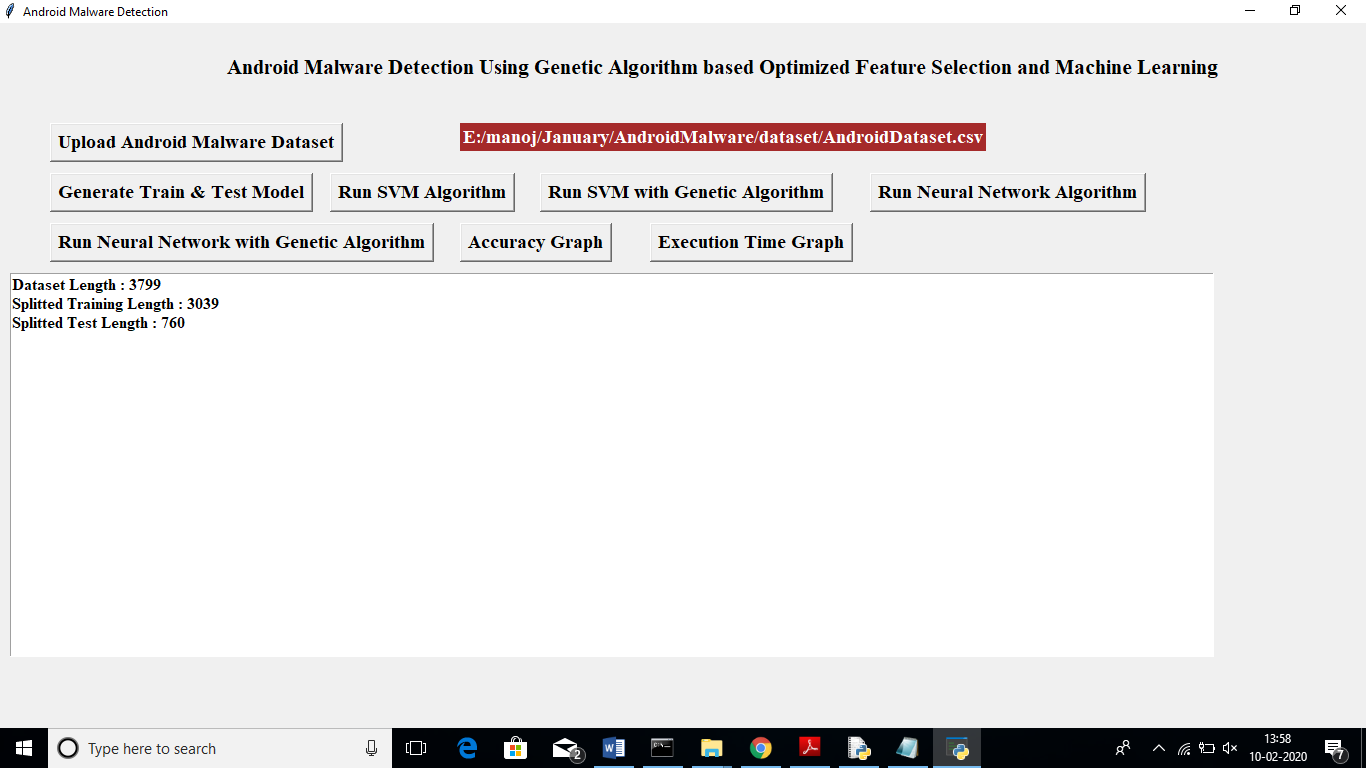
In above screen click on ‘Upload Android Malware Dataset’ button and upload dataset.



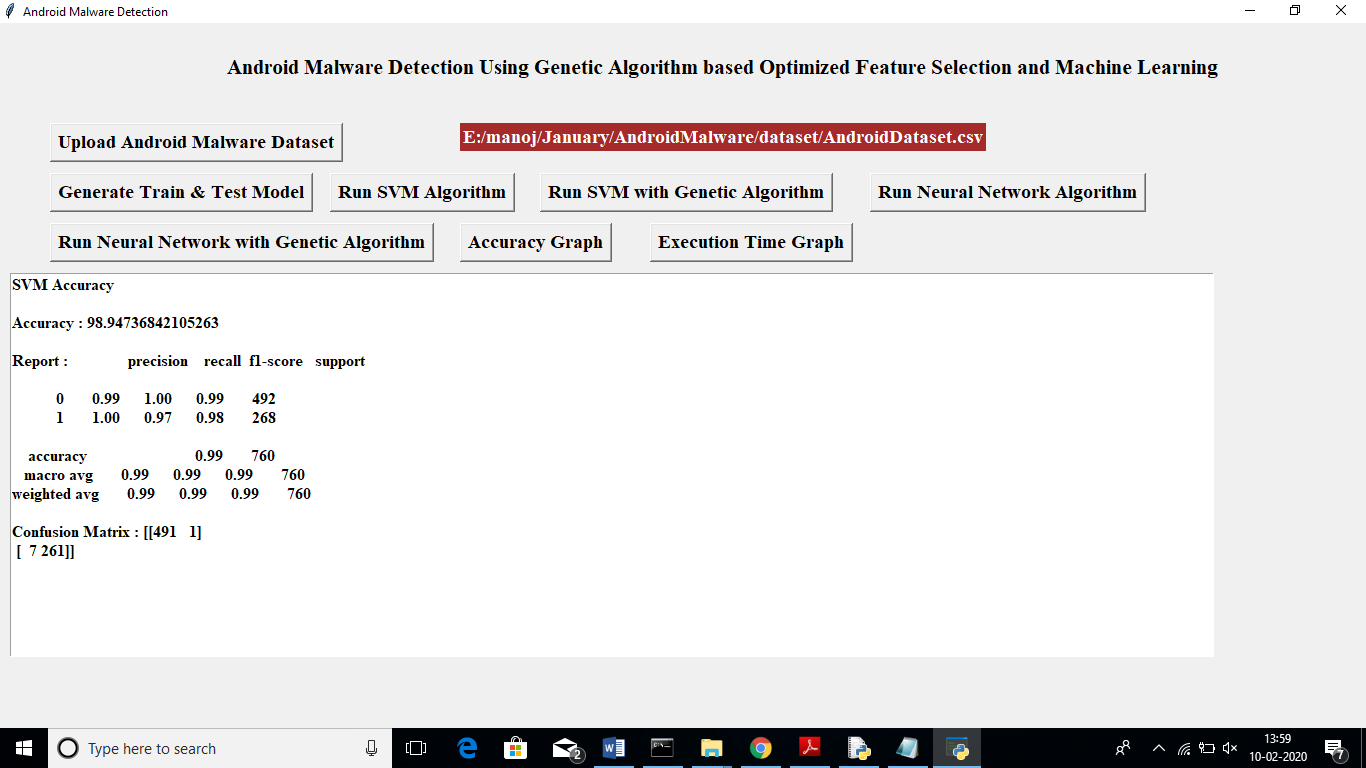
In above screen I am uploading ‘AndroidDataset.csv’ file and after upload will get below screen



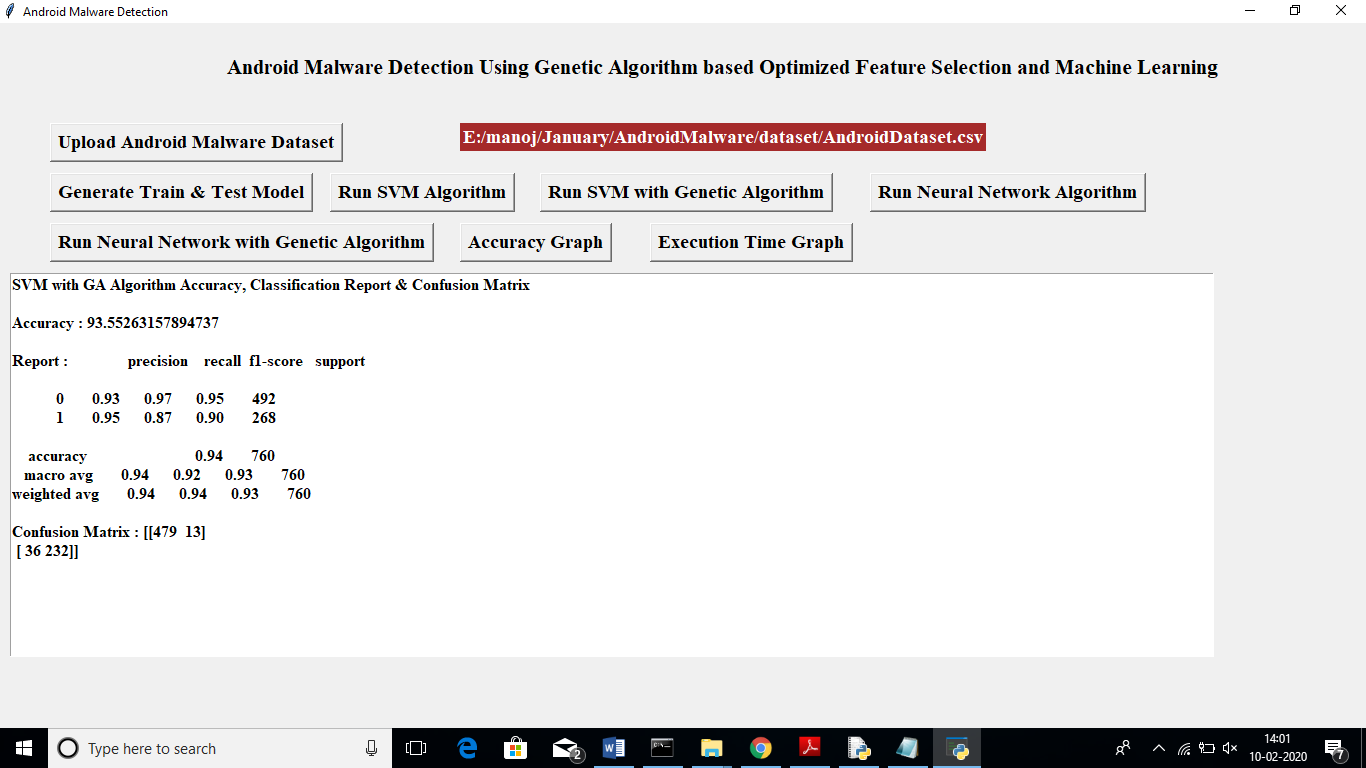
Now click on ‘Generate Train & Test Model’ button to split dataset into train and test part. All machine learning algorithms will take 80% dataset for training and 20% dataset to test accuracy of trained model. After clicking that button will get train and test model



In above screen we can see there are total 3799 android app records are there and application using 3039 records for training and 760 records for testing. Now we have both train and test model and now click on ‘Run SVM Algorithm’ button to generate SVM model on train and test and get its accuracy

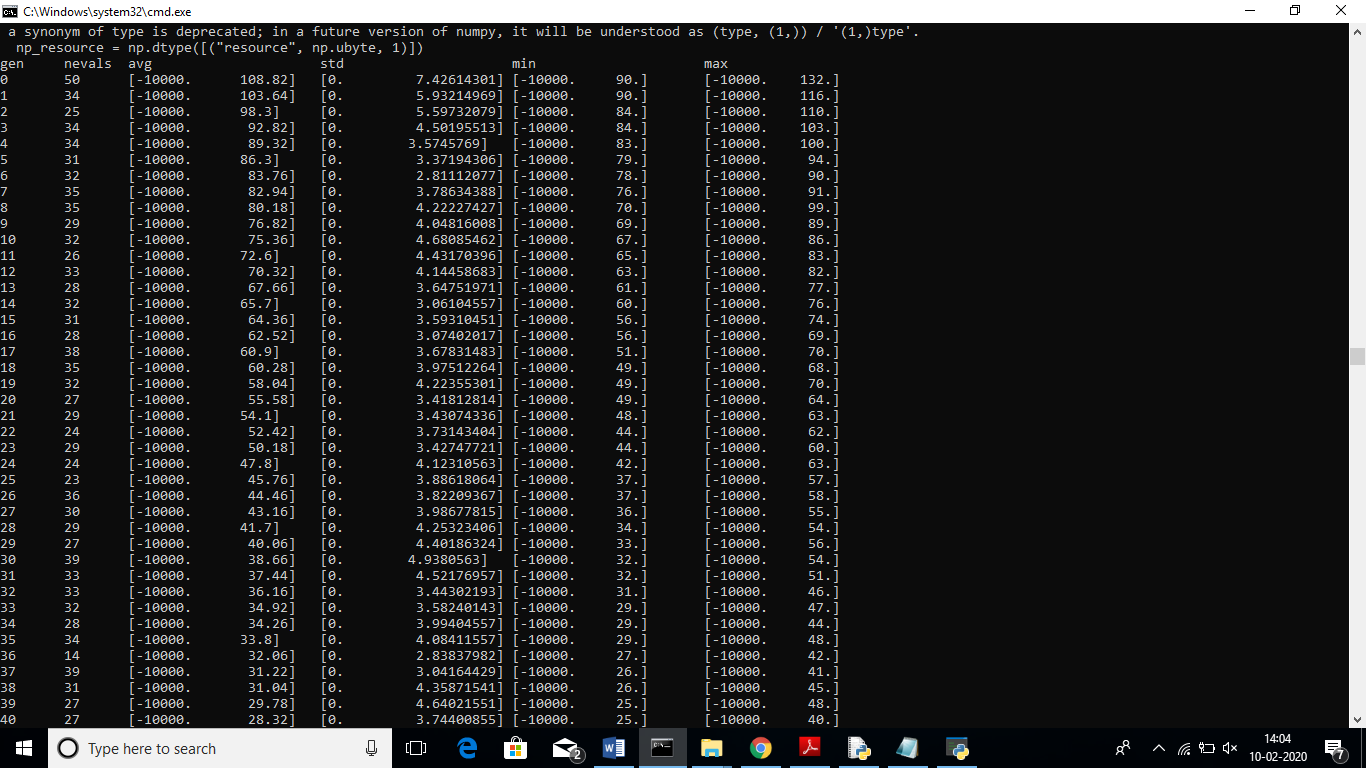


In above screen we got 98% accuracy for SVM and now click on ‘Run SVM with Genetic Algorithm’ button to choose optimize features and then run SVM on optimize features to get accuracy



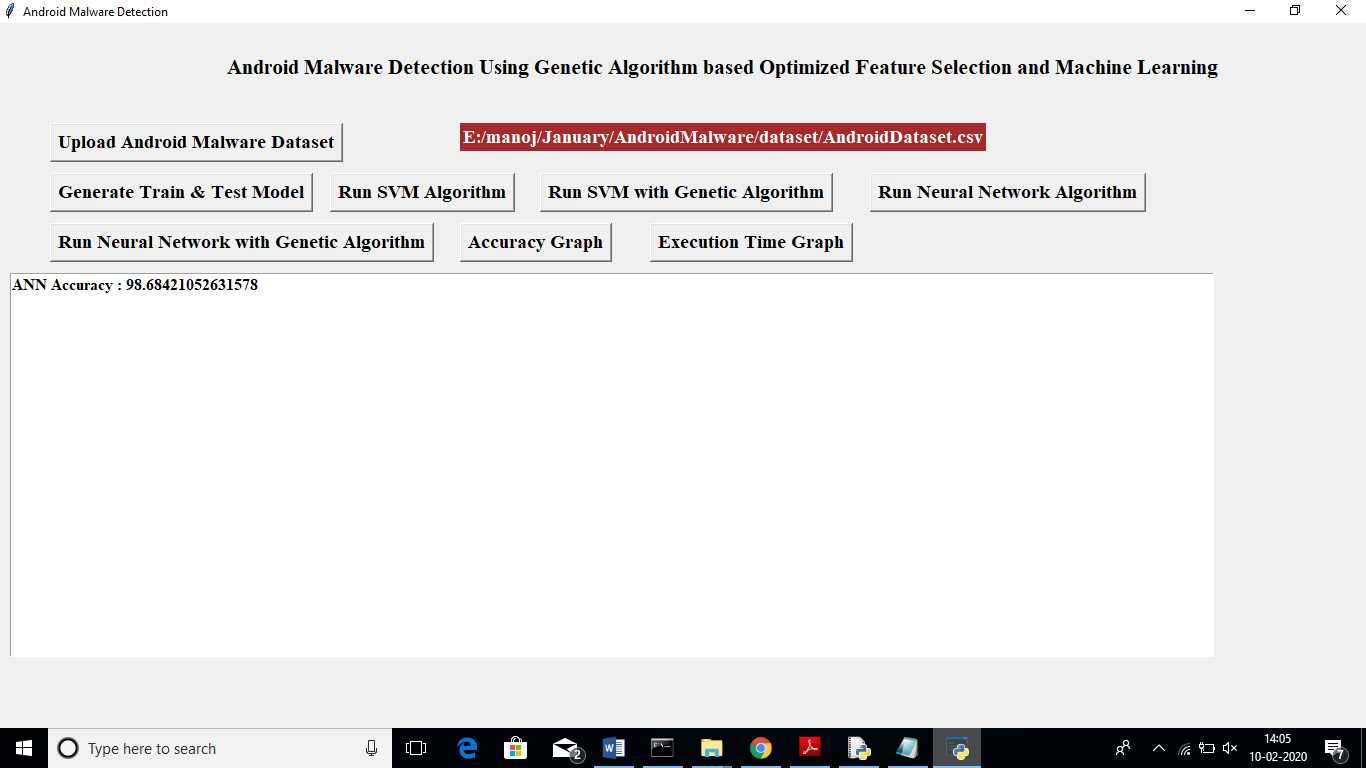
In above screen SVM with Genetic algorithm got 93% accuracy. Genetic with SVM accuracy is less but its execution time will be less which we can see at the time of comparison graph.

(Note: when u run genetic then 4 empty windows will open u just close all those 4 windows and let main window to run)

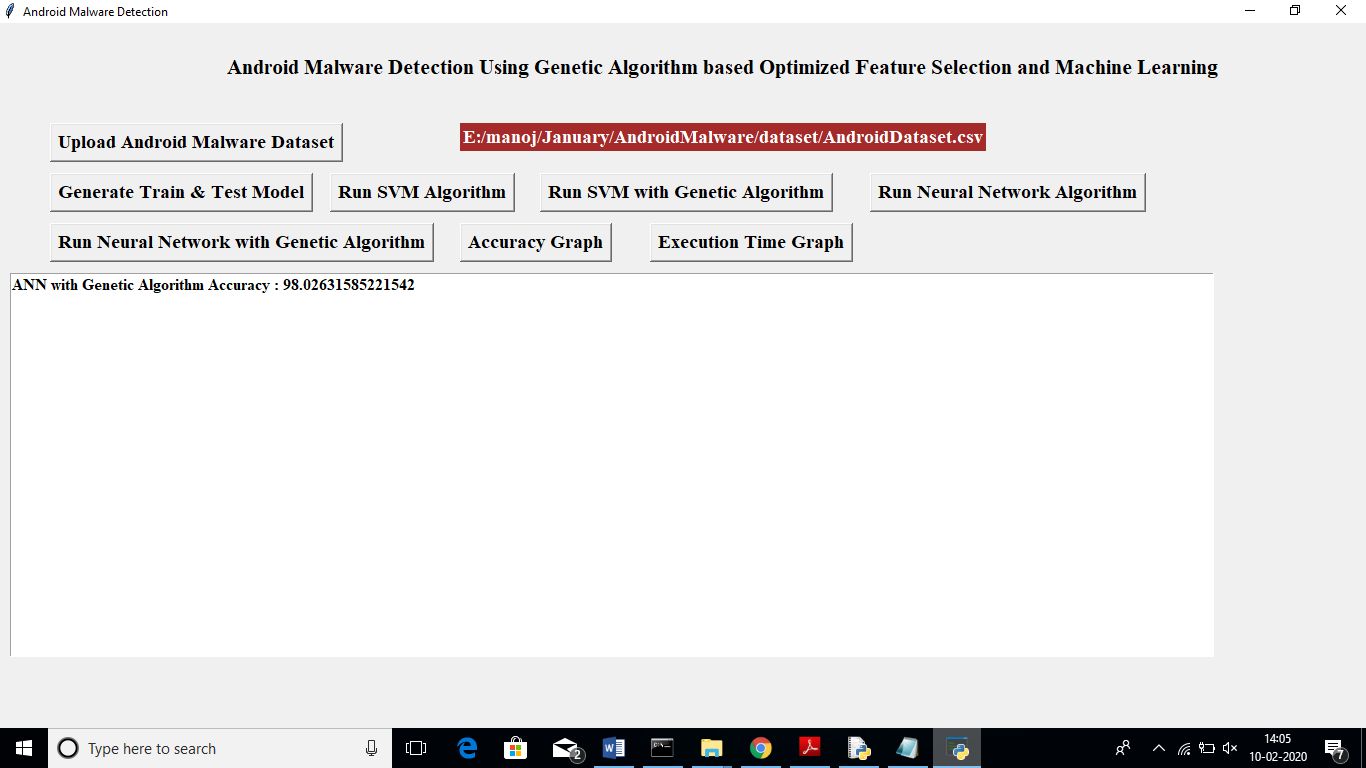


In above console we can see genetic algorithm chooses 40 features from all dataset features.

Now click on ‘Run Neural Network Algorithm’ button to test neural network accuracy.



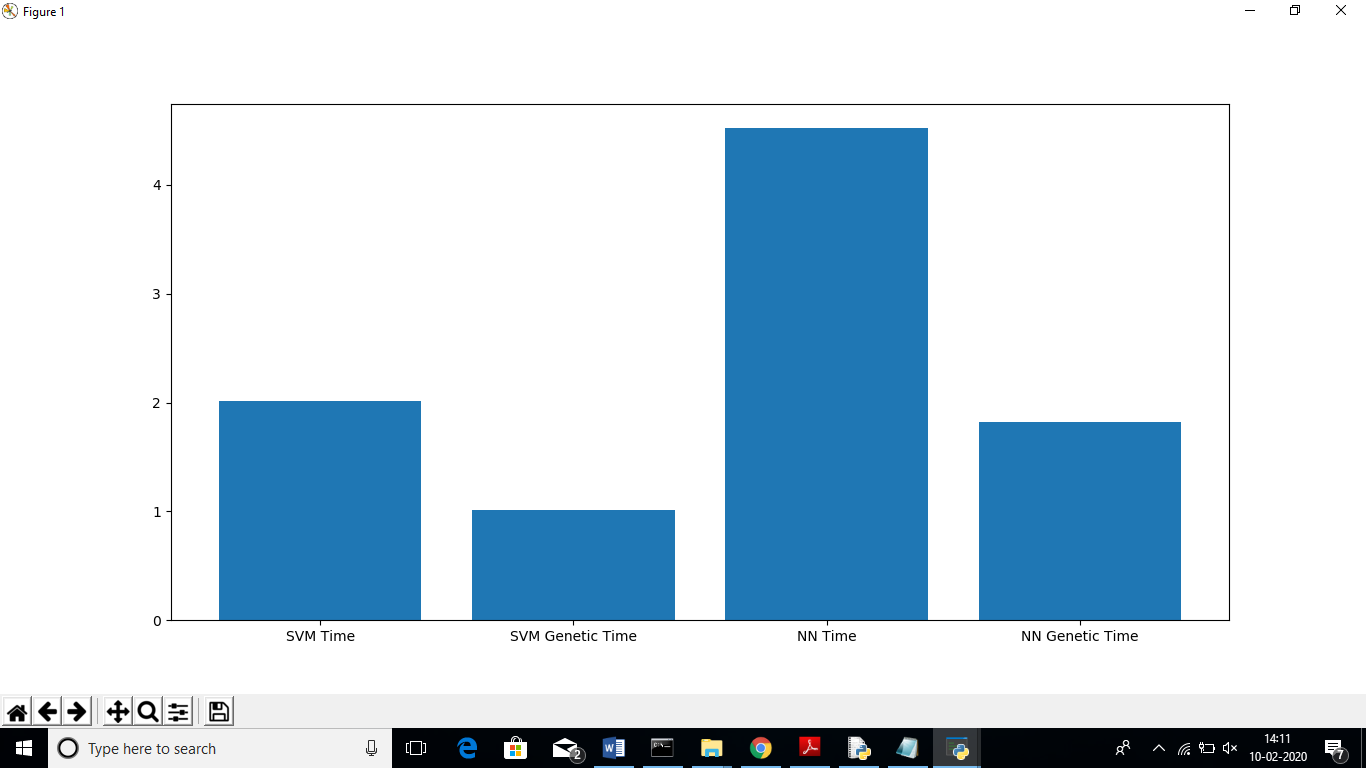
In above screen neural network also gave 98.64% accuracy. Now click on ‘Run Neural Network with Genetic Algorithm’ button to get NN accuracy with genetic algorithm



In above screen NN with genetic got 98.02% accuracy. Now click on ‘Accuracy Graph’ button to see all algorithms accuracy in graph



In above graph x-axis represents algorithm name and y-axis represents accuracy and in all SVM got high accuracy. Now click on ‘Execution Time Graph’ button to get execution time of all algorithm



In above graph x-axis represents algorithm name and y-axis represents execution time. From above graph we can conclude that with genetic algorithm machine learning algorithms taking less time to build model.