Detection of Android Malware Using Genetic Algorithm based Optimized Feature Selection

Mrs. S. MOUNIKA [1], Mr. Kadavakollu Siva Sankar [2]

#1 Assistant Professor in the department of AI&IT at DVR & DR HS MIC COLLEGE OF TECHNOLOGY (Autonomous), Kanchikacherla (NTR Dist, AP).

#2 MCA Student In The Department Of Computer Applications at DVR & Dr HS MIC College of Technology (Autonomous), Kanchikacherla, NTR District, A.P

ABSTRACT_ Android platform due to open source characteristic and Google backing has the largest global market share. Being the world's most popular operating system, it has drawn the attention of cyber criminals operating particularly through wide distribution of malicious applications. This paper proposes an effectual machine-learning based approach for Android Malware Detection making use of evolutionary Genetic algorithm for discriminatory feature selection. Selected features from Genetic algorithm are used to train machine learning classifiers and their capability in identification of Malware before and after feature selection is compared. The experimentation results validate that Genetic algorithm gives most optimized feature subset helping in reduction of feature dimension to less than half of the original feature-set. Classification accuracy of more than 94% is maintained post feature selection for the machine learning based classifiers, while working on much reduced feature dimension, thereby, having a positive impact on

computational complexity of learning classifiers.

1.INTRODUCTION

Android Apps are uninhibitedly accessible on Google Playstore, the official Android application store just as outsider application stores for clients to download. Because of its open source nature and fame, malware scholars are progressively zeroing in on creating malignant applications for Android working framework. Despite different endeavors by Google Playstore to ensure against pernicious applications, they actually discover their approach to mass market and cause mischief to clients by abusing individual data identified with their telephone directory, mail accounts, GPS area data and others for abuse by outsiders or, more than likely assume responsibility for the telephones distantly. Subsequently, there is have to perform malware examination or figuring out of such pernicious applications which present genuine danger to Android stages. Extensively, Android Malware investigation is of two sorts: Static Analysis and Dynamic Analysis. Static investigation essentially includes breaking down the code structure without executing it while dynamic examination will be assessment of the runtime conduct of Android Apps in obliged climate. Yielded to the ever-expanding variations of Android Malware presenting zero-day dangers, an effective system for recognition of Android malwares is required. Rather than signature-based methodology which requires ordinary update of mark information base.

2.LITERATURE SURVEY

2.1 D. Arp, M. Spreitzenbarth, M. Hübner, H. Gascon, and K. Rieck, "Drebin: Effective and Explainable Detection of Android Malware in Your Pocket," in Proceedings 2014 Network and Distributed System Security Symposium, 2014.

Pernicious applications represent a danger to the security of the Android stage. The developing sum and variety of these applications render traditional protections generally insufficient and consequently Android cell phones regularly stay unprotected from novel malware. In this paper, we propose DREBIN, a lightweight technique for recognition of Android malware that empowers recognizing pernicious applications legitimately on the cell phone. As the restricted assets obstruct checking applications at run-time, DREBIN plays out an expansive static examination, gathering whatever number highlights of an application as could be allowed. These highlights are installed in a joint vector space, with the end goal that commonplace examples characteristic for malware can be naturally distinguished and utilized for clarifying the choices of our strategy. In an assessment with 123,453 applications and 5,560 malware tests DREBIN outflanks a few related methodologies and identifies 94% of the malware with few bogus cautions, where the clarifications accommodated every recognition uncover pertinent properties of the recognized malware. On five famous cell phones, the strategy requires 10 seconds for an examination by and large, delivering it appropriate for checking downloaded applications legitimately on the gadget

2.2 N. Milosevic, A. Dehghantanha, and K. K. R. Choo, "AI supported Android malware characterization," Comput. Electr. Eng., vol. 61, pp. 266–274, 2017.

Malware have been utilized as a methods for leading digital assaults for quite a long time. Wide selection of cell phones, which store heaps of private and secret data, made them a significant objective for malware engineers. Android as the predominant versatile working framework has consistently been an intriguing stage for malware engineers and bunches of Android malware species are contaminating weak clients consistently which make manual malware examination an unthinkable mission. Utilizing AI strategies for malware criminology would help digital criminological agents in their battle against pernicious projects. In this paper, we present two AI helped approaches for static examination of the versatile applications: one dependent on consents , while the other dependent on source code investigation that uses a pack of words portrayal model. Our source code based characterization accomplished F-score of 95.1%, while the methodology that pre-owned consent names just performed with F-proportion of 89%. Our methodology gives a technique to computerized static code investigation and malware recognition with high precision and diminishes cell phone malware examination time.

2.3 J. Li, L. Sun, Q. Yan, Z. Li, W. Srisa-An, and H. Ye, "Critical Permission Identification for Machine-Learning-Based Android Malware Detection," IEEE Trans. Ind. Informatics, vol. 14, no. 7, pp. 3216–3225, 2018.

Android is the most generally utilized versatile working framework (OS). An enormous number of outsider Android (application) markets have arisen. The nonattendance of outsider market guideline has incited research organizations to propose distinctive malware recognition methods. Nonetheless, because of upgrades of malware itself and Android framework, it is hard to plan a recognition technique that can productively and adequately recognize vindictive applications for quite a while. Then, embracing more highlights will build the unpredictability of the model and the computational expense of the framework. Consents assume a crucial function in the security of the Android applications. Term Frequency—Inverse Document Frequency (TF-IDF) is utilized to survey the significance of a word for a record set in a corpus. The static examination technique doesn't have to run the application. It can proficiently and precisely extricate the authorizations from an application. In view of this perception and viewpoint, in this paper, another static location strategy dependent on TF-IDF and Machine Learning is proposed. The framework authorizations are extricated in Android application bundle's (Apk's) show record. TF-IDF calculation is utilized to compute the authorization esteem (PV) of every consent and the affectability estimation of apk (SVOA) of each application. The

SVOA and the quantity of the pre-owned authorizations are found out and tried by AI. 6070 kindhearted applications and 9419 malware are utilized to assess the proposed approach. The investigation results show that solitary utilize perilous authorizations or the quantity of utilized consents can't precisely recognize whether an application is malevolent or amiable. For malware discovery, the proposed approach accomplish up to 99.5% exactness and the learning and preparing time just requirements 0.05s. For malware families location, the precision is 99.6%. The outcomes demonstrate that the technique for obscure/new example's recognition precision is 92.71%. Analyzed against other best in class draws near, the proposed approach is more compelling by identifying malware and malware families.

3.PROPOSED SYSTEM

Two set of Android Apps or APKs: Malware/Goodware are reverse engineered to extract features such as permissions and count of App Components such as Activity, Services, Content Providers, etc. These features are used as featurevector with class labels as Malware and Goodware represented by 0 and 1 respectively in CSV format.

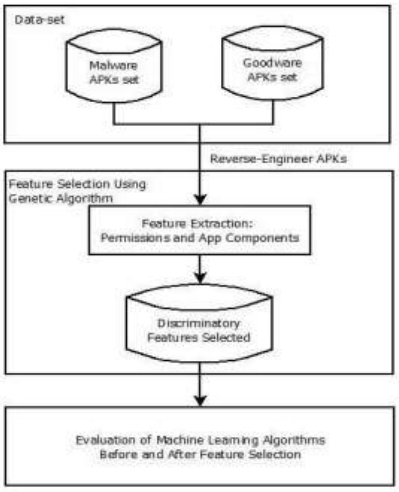

Data-set
Malware APKs set
Goodware APKs set
Reverse-Engineer APKs
Feature Selection Using Genetic Algorithm
Feature Extraction: Permissions and App Components
Discriminatory Features Selected
Evaluation of Machine Learning Algorithms Before and After Feature Selection

Fig. 1. Proposed Methodology

To reduce dimensionality of feature-set, the CSV is fed to Genetic Algorithm to select the most optimized set of features. The optimized set of features obtained is used for training two machine learning classifiers: Support Vector Machine and Neural Network.

In the proposed methodology, static features are obtained from AndroidManifest.xml which contains all the important information needed by any Android platform about the Apps. Androguard tool has been used for disassembling of the APKs and getting the static features.

Advantages :-

- Security
- Proposed a novel and efficient algorithm for feature selection to improve overall detection accuracy.
- Machine-learning based approach in combination with static and dynamic analysis can be used to detect new variants of Android Malware

posing zero-day threats.

3.1 GENETIC ALGORITHM

Genetic algorithms (GAs) are optimization techniques inspired by the process of natural selection and genetics. They are widely used in machine learning and various other fields to solve complex optimization problems. GAs operate on a population of candidate solutions, which undergo selection, crossover, and mutation operations to evolve and improve over successive generations.

The importance of genetic algorithms in machine learning can be understood through the following key points:

Optimization: GAs excel at finding near-optimal or optimal solutions in large, complex search spaces. They are particularly useful when traditional optimization techniques struggle due to the high dimensionality or non-linearity of the problem. By exploring a diverse set of solutions and leveraging evolutionary operators, GAs can converge towards a good solution.

Feature Selection: In machine learning, feature selection plays a crucial role in improving model performance and reducing overfitting. GAs can be used to identify the most informative subset of features from a large pool. By encoding different combinations of features as individuals in the population, GAs can effectively explore the feature space and select the most relevant features.

Hyperparameter Tuning: Machine learning models often involve various hyperparameters that need to be tuned to achieve optimal performance. GAs can be employed to search the hyperparameter space and find good combinations. By encoding different parameter configurations as individuals, GAs can efficiently explore the hyperparameter space and evolve towards better solutions.

Non-Differentiable and Black-Box Optimization: GAs are particularly valuable in scenarios where the objective function is non-differentiable or the underlying system is a black box, meaning the gradient information is unavailable or unreliable. Since GAs only require the objective function evaluations, they can handle complex optimization problems where the analytical gradients are not feasible.

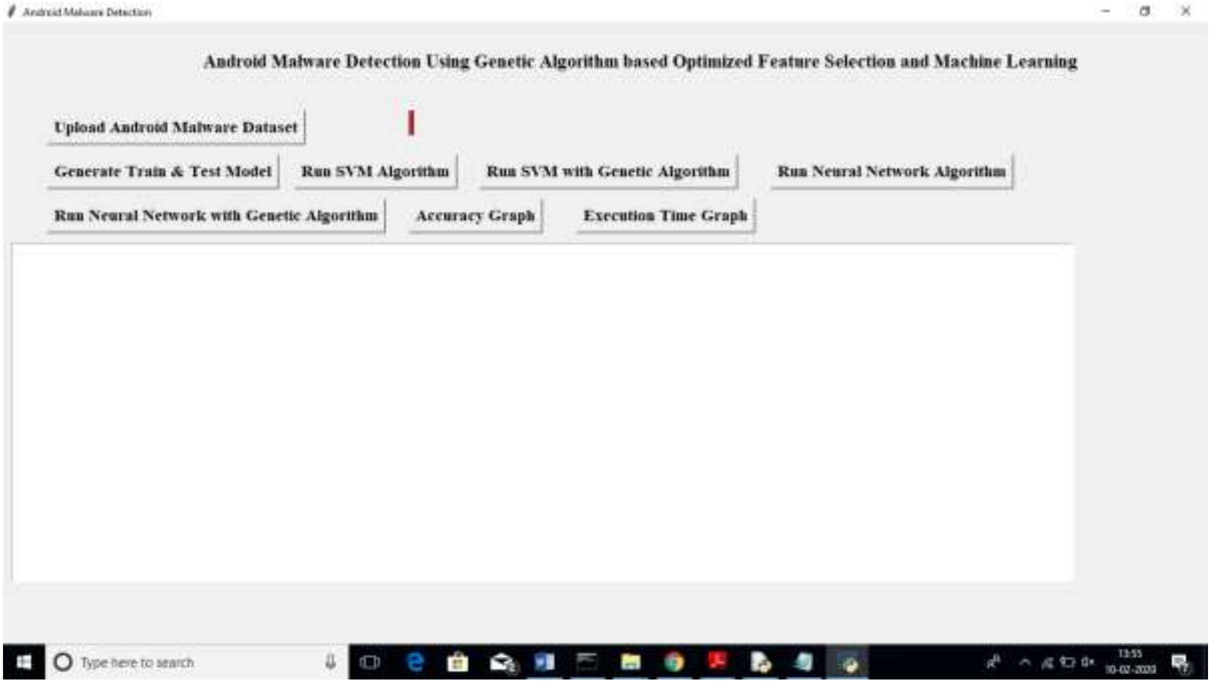
Global Search: GAs are designed to perform global search rather than being stuck in local optima. They maintain a diverse population, ensuring exploration of different regions of the search space. This ability makes GAs suitable for tasks where finding the global optimum is critical, such as neural network architecture search or solving complex combinatorial optimization problems.

Exploration and Exploitation: GAs strike a balance between exploration (diversification) and exploitation (intensification). Initially, they explore a wide range of solutions to cover the search space.

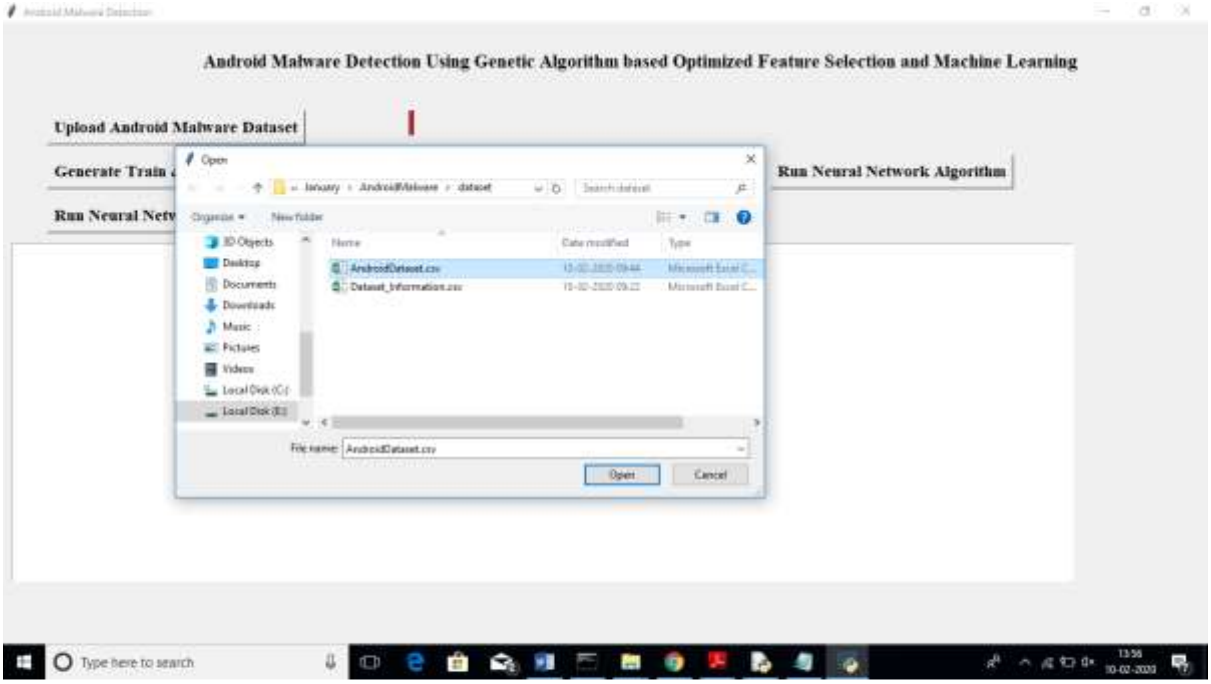
As the algorithm progresses, they exploit promising solutions by recombining and mutating them to refine the population and focus on regions with better fitness.

Overall, genetic algorithms are powerful optimization techniques that find applications in various machine learning tasks. They offer an efficient and effective approach to tackle complex problems, optimize model performance, and discover optimal solutions in diverse domains.

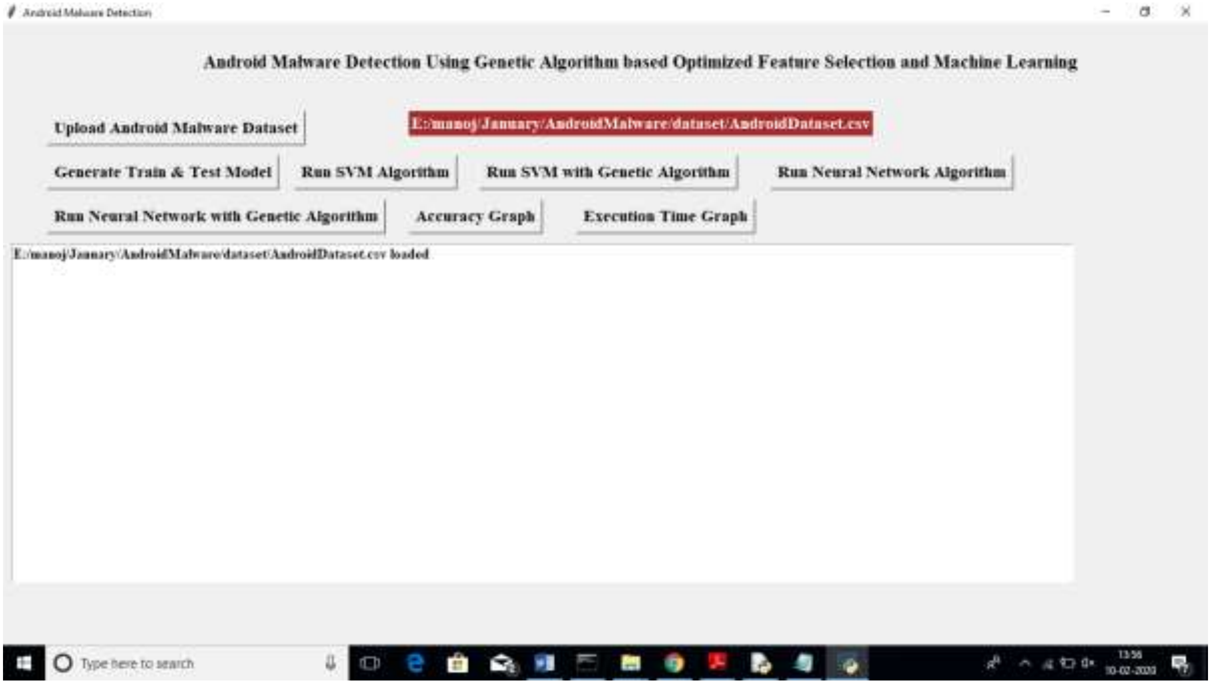
4.RESULTS AND DISCUSSIONS


Android Malware Detection
Android Malware Detection Using Genetic Algorithm based Optimized Feature Selection and Machine Learning
Upload Android Malware Dataset
Generate Train & Test Model
Run SVM Algorithm
Run SVM with Genetic Algorithm
Run Neural Network Algorithm
Run Neural Network with Genetic Algorithm
Accuracy Graph
Execution Time Graph
Type here to search

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


Android Malware Detection
Android Malware Detection Using Genetic Algorithm based Optimized Feature Selection and Machine Learning
Upload Android Malware Dataset
Generate Train
Run Neural Netw
Run Neural Network Algorithm
Open
January › AndroidMalware › dataset
Organize ▾
New folder
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
Local Disk (E:)
Name
Date modified
Type
AndroidDataset.csv
Dataset_Information.csv
Microsoft Excel C...
File name: AndroidDataset.csv
Open
Cancel
Type here to search

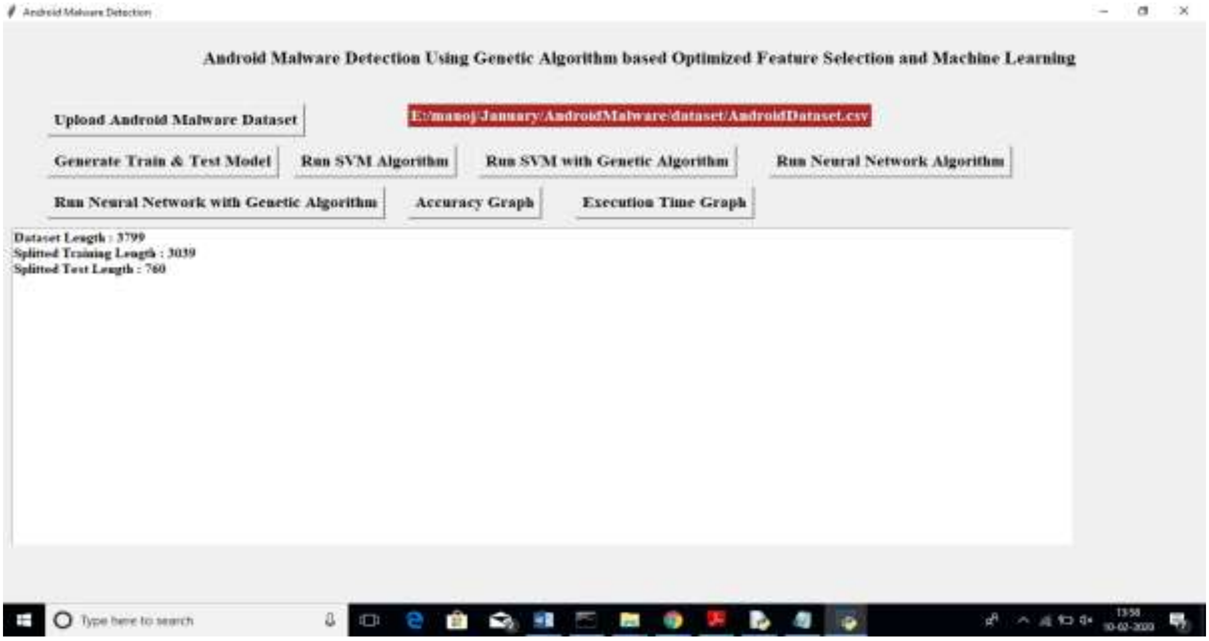
In above screen I am uploading 'AndroidDataset.csv' file and after upload will get below screen


Android Malware Detection
Android Malware Detection Using Genetic Algorithm based Optimized Feature Selection and Machine Learning
Upload Android Malware Dataset
Generate Train & Test Model
Run SVM Algorithm
Run SVM with Genetic Algorithm
Run Neural Network Algorithm
Run Neural Network with Genetic Algorithm
Accuracy Graph
Execution Time Graph
Type here to search

E:/manoj/January/AndroidMalware/dataset/AndroidDataset.csv

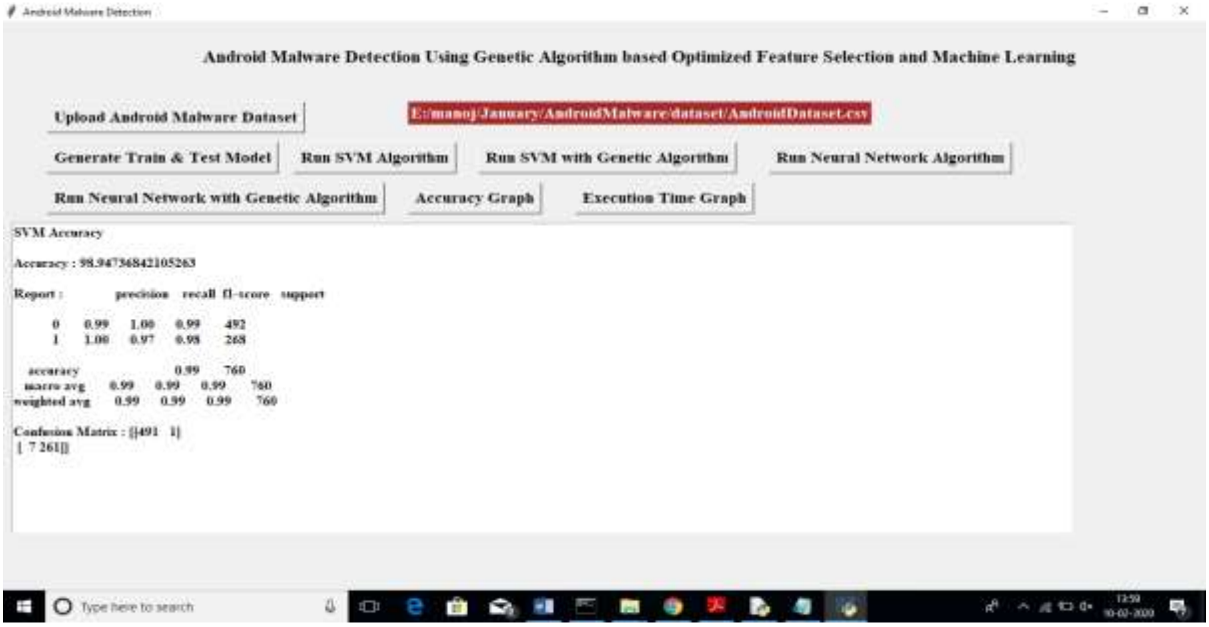
E:/manoj/January/AndroidMalware/dataset/AndroidDataset.csv loaded

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


Android Malware Detection
Android Malware Detection Using Genetic Algorithm based Optimized Feature Selection and Machine Learning
Upload Android Malware Dataset
E:/manoj/January/AndroidMalware/dataset/AndroidDataset.csv
Generate Train & Test Model
Run SVM Algorithm
Run SVM with Genetic Algorithm
Run Neural Network Algorithm
Run Neural Network with Genetic Algorithm
Accuracy Graph
Execution Time Graph
Type here to search

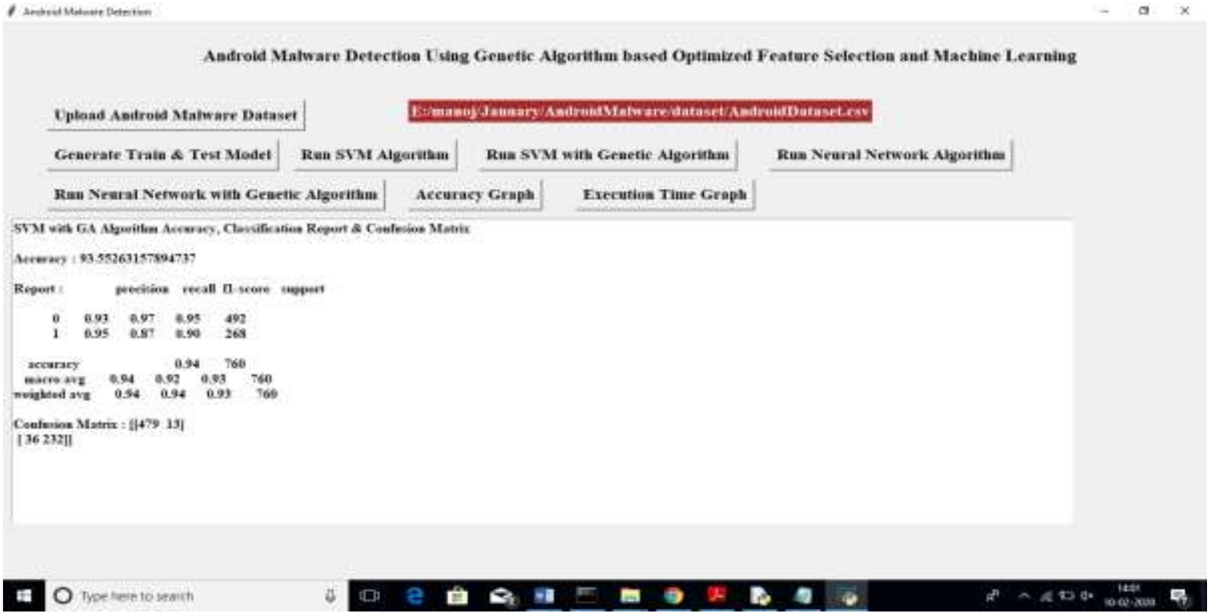
```
Dataset Length : 3799
Splitted Training Length : 3039
Splitted Test Length : 760
```

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


Android Malware Detection
Android Malware Detection Using Genetic Algorithm based Optimized Feature Selection and Machine Learning
Upload Android Malware Dataset
E:/manoj/January/AndroidMalware/dataset/AndroidDataset.csv
Generate Train & Test Model
Run SVM Algorithm
Run SVM with Genetic Algorithm
Run Neural Network Algorithm
Run Neural Network with Genetic Algorithm
Accuracy Graph
Execution Time Graph
Type here to search

```
SVM Accuracy

Accuracy : 98.94736842105263

Report :           precision    recall  f1-score   support

           0       0.99      1.00      0.99       492
           1       1.00      0.97      0.98       268

    accuracy                           0.99       760
   macro avg       0.99      0.99      0.99       760
weighted avg       0.99      0.99      0.99       760

Confusion Matrix : [[491   1]
 [  7 261]]
```

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

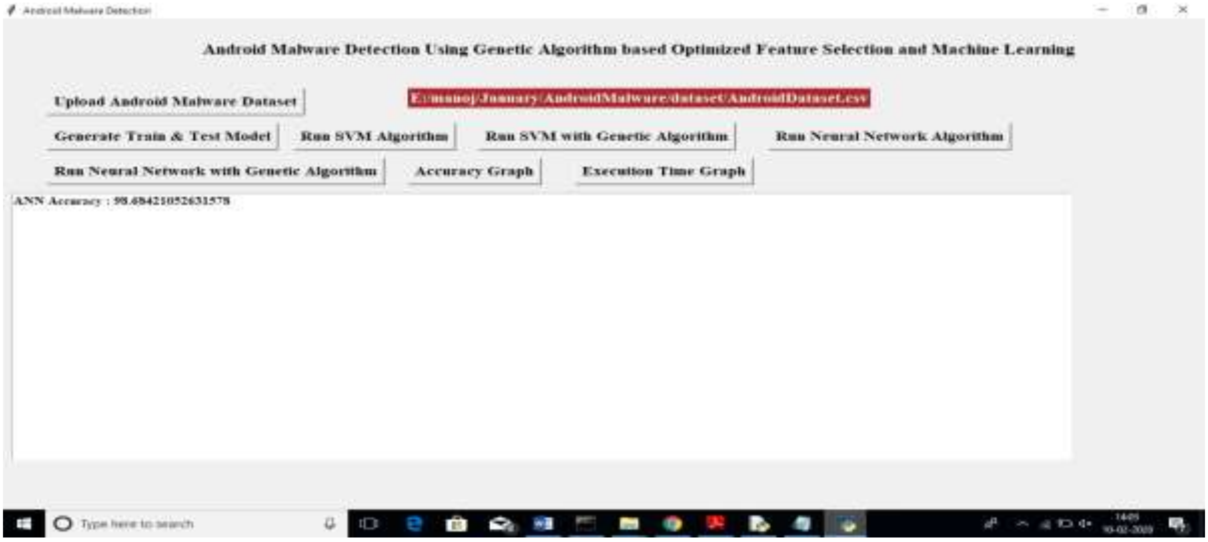

Android Malware Detection
Android Malware Detection Using Genetic Algorithm based Optimized Feature Selection and Machine Learning
Upload Android Malware Dataset
E:/manoj/January/AndroidMalware/dataset/AndroidDataset.csv
Generate Train & Test Model
Run SVM Algorithm
Run SVM with Genetic Algorithm
Run Neural Network Algorithm
Run Neural Network with Genetic Algorithm
Accuracy Graph
Execution Time Graph
Type here to search

```
SVM with GA Algorithm Accuracy, Classification Report & Confusion Matrix

Accuracy : 93.55263157894737

Report :           precision    recall  f1-score   support

       0       0.93      0.97      0.95       492
       1       0.95      0.87      0.90       268

    accuracy                           0.94       760
   macro avg       0.94      0.92      0.93       760
weighted avg       0.94      0.94      0.93       760

Confusion Matrix : [[479  13]
 [ 36 232]]
```

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.

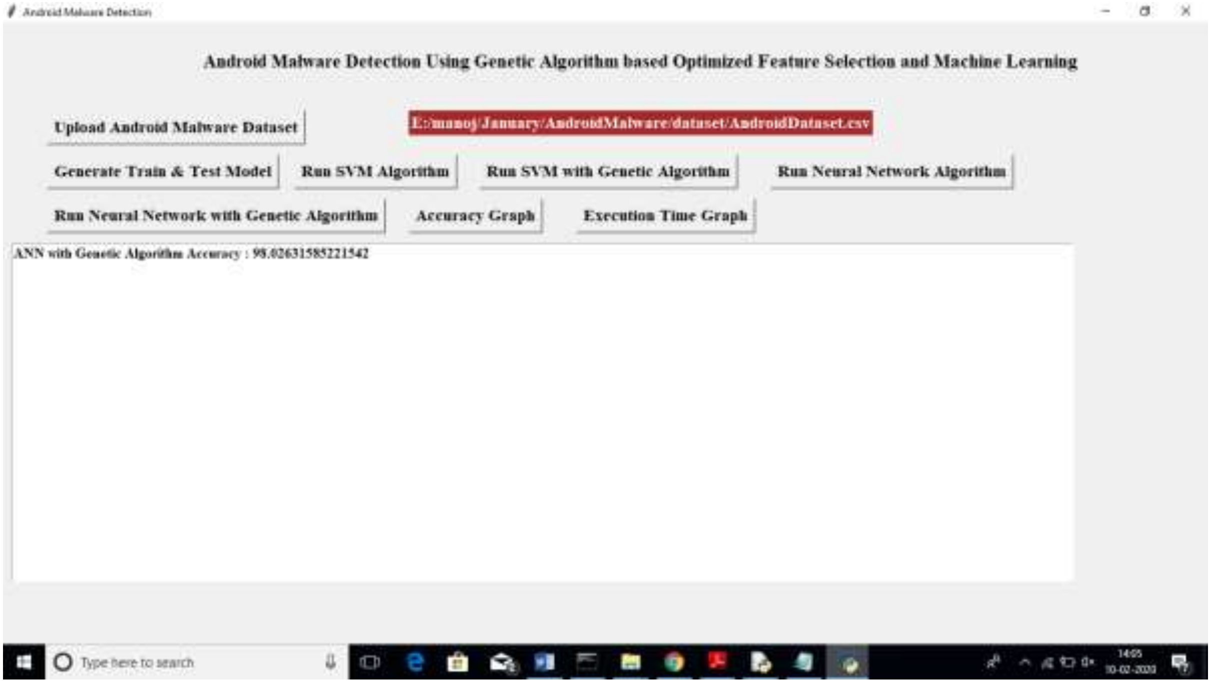
(Note: when u run genetic then 4 empty windows will open u just close all those 4 windows and let main window to run)

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


Android Malware Detection
Android Malware Detection Using Genetic Algorithm based Optimized Feature Selection and Machine Learning
Upload Android Malware Dataset
E:/manoj/January/AndroidMalware/dataset/AndroidDataset.csv
Generate Train & Test Model
Run SVM Algorithm
Run SVM with Genetic Algorithm
Run Neural Network Algorithm
Run Neural Network with Genetic Algorithm
Accuracy Graph
Execution Time Graph
Type here to search

```
ANN Accuracy : 98.68421052631578
```

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

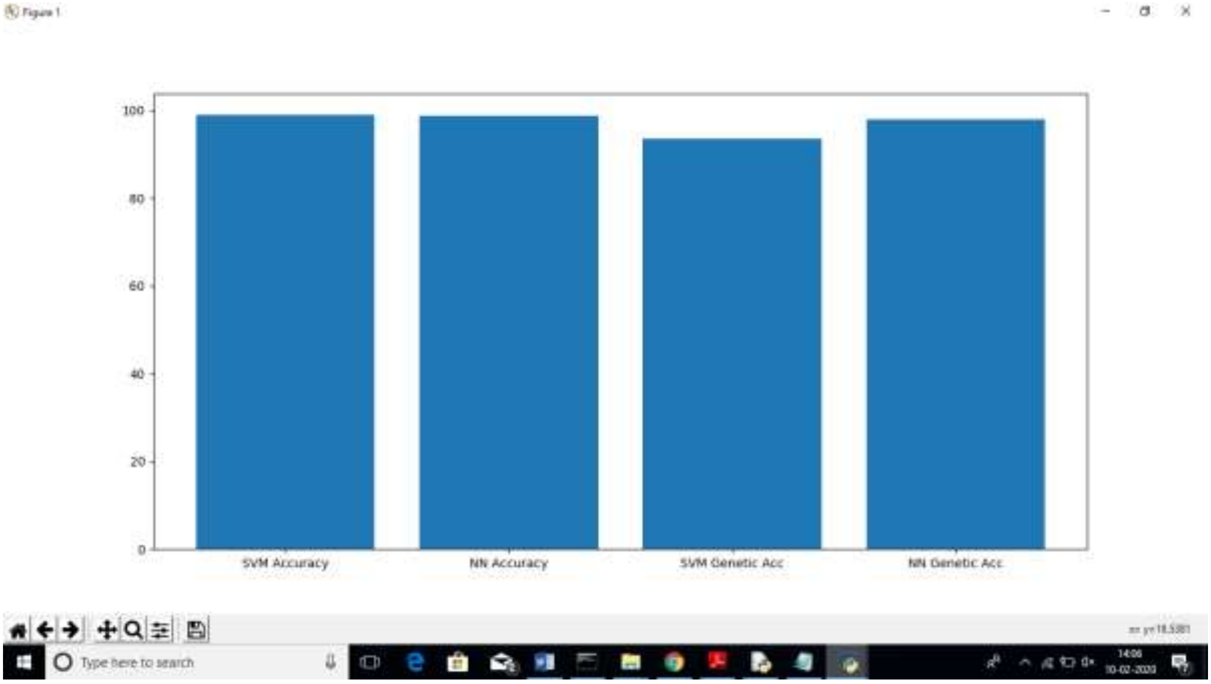

Android Malware Detection
Upload Android Malware Dataset
Generate Train & Test Model
Run SVM Algorithm
Run SVM with Genetic Algorithm
Run Neural Network Algorithm
Run Neural Network with Genetic Algorithm
Accuracy Graph
Execution Time Graph
Type here to search

Android Malware Detection Using Genetic Algorithm based Optimized Feature Selection and Machine Learning

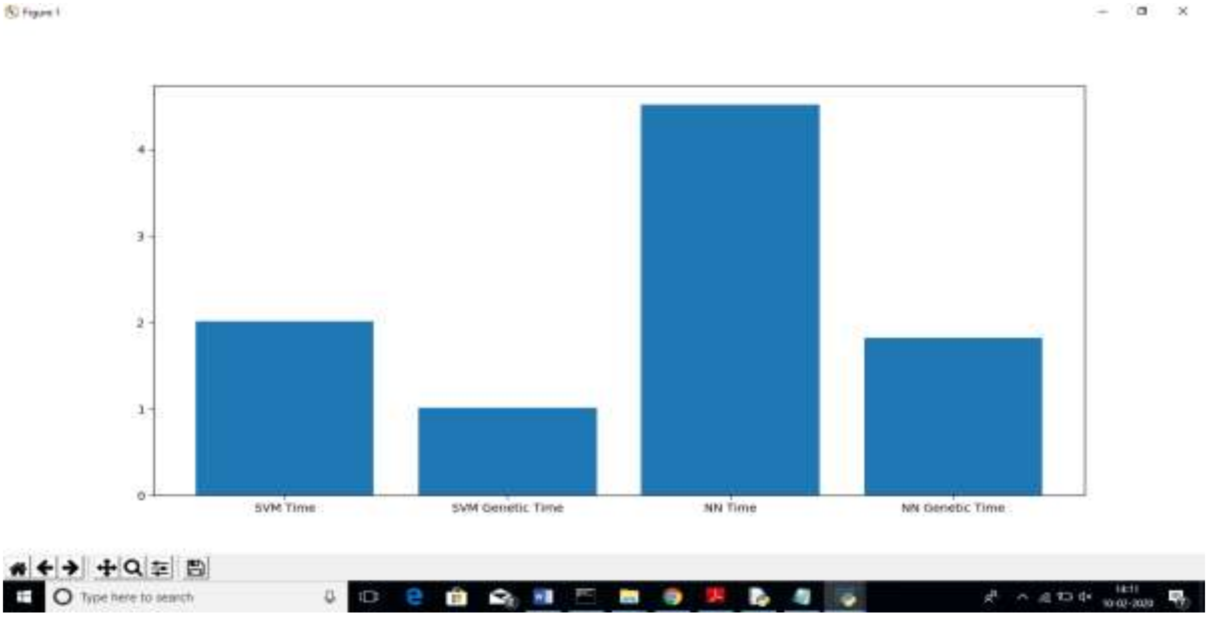
E:/manoj/January/AndroidMalware/dataset/AndroidDataset.csv

```
ANN with Genetic Algorithm Accuracy : 98.02631585221542
```

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.

5.CONCLUSION

As the quantity of dangers presented to Android stages is expanding everyday, spreading chiefly through malignant applications or malwares, consequently it is essential to plan a system which can identify such malwares with precise outcomes. Where signature-based methodology neglects to recognize new variations of malware presenting zero-day dangers, AI based methodologies are being utilized. The proposed system endeavors to utilize transformative Genetic Algorithm to get most enhanced element subset which can be utilized to prepare AI calculations in most proficient manner. From experimentations, it tends to be seen that a fair arrangement exactness of over 94% is kept up utilizing Support Vector Machine and Neural Network classifiers while chipping away at lower measurement include set, in this manner decreasing the preparation multifaceted nature of the classifiers Further work can be upgraded utilizing bigger datasets for improved outcomes and dissecting the impact on other AI calculations when utilized related to Genetic Algorithm.

REFERENCES

D. Arp, M. Spreitzenbarth, M. Hübner, H. Gascon, and K. Rieck, “Drebin: Effective and Explainable Detection of Android Malware in Your Pocket,” in Proceedings 2014 Network and Distributed System Security Symposium, 2014.

[2] N. Milosevic, A. Dehghantanha, and K. K. R. Choo, “Machine learning aided Android malware classification,” Comput. Electr. Eng., vol. 61, pp. 266–274, 2017.

[3] J. Li, L. Sun, Q. Yan, Z. Li, W. Srisa-An, and H. Ye, “Significant Permission Identification for Machine-Learning-Based Android Malware Detection,” IEEE Trans. Ind. Informatics, vol. 14, no. 7, pp. 3216–3225, 2018.

[4] A. Saracino, D. Sgandurra, G. Dini, and F. Martinelli, “MADAM: Effective and Efficient Behavior-based Android Malware Detection and Prevention,” IEEE Trans. Dependable Secur. Comput., vol. 15, no. 1, pp. 83–97, 2018.

[5] S. Arshad, M. A. Shah, A. Wahid, A. Mehmood, H. Song, and H. Yu, “SAMADroid: A Novel 3-Level Hybrid Malware Detection Model for Android Operating System,” IEEE Access, vol. 6, pp. 4321–4339, 2018.

Author’s Profile

Mrs.S.MOUNIKA completed her Bachelor of Technology in Computer Science and Engineering. She completed her Masters of Technology in Computer Science and Engineering from JNTU KAKINADA UNIVERSITY. Currently working as an Assistant Professor in the department of AI&IT at DVR & DR HS MIC COLLEGE OF TECHNOLOGY (Autonomous), Kanchikacherla (NTR Dist, AP). Her areas of interest are Data Mining, Cloud Computing and Machine Learning & Networks

Mr. Kadavakollu Siva Sankar, as MCA student in the department of DCA at DVR & DR. HS MIC COLLEGE OF TECHNOLOGY, Kanchikacherla, NTR District. He has completed BSC in MARUTHI DEGREE COLLEGE From KRISHNA UNIVERSITY. His areas of interests are Networks, Machine Learning and Cloud Computing.