IJARST

International Journal For Advanced Research In Science & Technology

A peer reviewed international journal www.ijarst.in

ISSN: 2457-0362

ANDROID MALWARE DETECTION USING GENETIC ALGORITHM BASED OPTIMIZED FEATURE SELECTION AND MACHINE LEARNING

Ms.M.ANITHA 1, Mr.Y.NAGAMALLESWARA RAO 2,
Ms. K. MONIKA RAGAMAI 3

#1 Assistant professor in the Department of Master of Computer Applications in the SRK Institute of Technology, Enikepadu, Vijayawada, NTR District

#2 Assistant professor in the Department of Master of Computer Applications SRK Institute of Technology, Enikepadu, Vijayawada, NTR District

#3 MCA student in the Department of Master of Computer Applications at SRK Institute of Technology, Enikepadu, Vijayawada, NTR District

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 Play store, 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 Play store 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 malware's is required. Rather than signature-based methodology which requires ordinary update of mark information base.

2.LITERATURE SURVEY

Android is one of the most popular platforms for smartphones today. With several hundred

thousands of applications in different markets, it provides a wealth of functionality to its users. Unfortunately, smartphones running Android are increasingly targeted by attackers and infected with malicious software. In contrast to other platforms, Android allows for installing applications from unverified sources, such as third-party markets, which makes bundling and distributing applications with malware easy for attackers. According to a recent study over 55,000 malicious applications and 119 new malware families have been discovered in 2012 alone [18]. It is evident that there is a need for stopping the proliferation of malware on Android markets and smartphones.

The Android platform provides several security measures that harden the installation of malware, most notably the Android permission system. To perform certain tasks on the device, such as sending a SMS message, each application has to explicitly request permission from the user during the installation. However, many users tend to blindly grant permissions to unknown applications and thereby undermine the purpose of the permission system. As a consequence, malicious applications are hardly constrained by the Android permission system in practice.

A large body of research has thus studied methods for analyzing and detecting Android malware prior to their installation. These methods can be roughly categorized into approaches using static and dynamic analysis. For example, TaintDroid [11], DroidRanger [40] and DroidScope [37] are methods that can monitor the behavior of applications at run-time. Although very effective in identifying malicious activity, run-time monitoring suffers from a significant overhead and cannot be directly applied on mobile devices. By contrast, static analysis methods, such as Kirin [13], Stowaway [15] and RiskRanker [21], usually induce only a small run-time overhead. While these approaches are efficient and scalable, they mainly build on manually crafted detection patterns which are often not available for new malware instances. Moreover, most of these methods do not provide explanations for their decisions and are thus opaque to the practitioner.

Shabtai et al. [12] contributed a system that detects malicious behavior through network traffic analysis. This is done by logging user-specific network traffic patterns per examined app and subsequently identifying deviations that can be flagged as malicious. To evaluate their model, they employed the C4.5 algorithm, achieving an accuracy of up to 94%.

Canfora et al. [13] suggested an Android malware detection scheme that analyzes opcode frequency histograms; this is accomplished by observing the frequency of occurrences of each group of op-codes. Precisely, their detection model capitalizes on a vector of features obtained from eight Dalvik op-codes. These op-codes are usually used to alter the app's control flow. Six classification models were employed during the evaluation, namely LadTree, NBTree, RandomForest, RandomTree and RepTree. The proposed model was applied separately to the eight features and the three groups of features. The first group includes the move and the jump features, the second involves two well-known distance metrics, namely Manhattan and Euclidean distance, and the last embraces all the four features. The proposed method was evaluated on the Drebin dataset using several classifiers, namely J48, LadTree, NBTree, Random Forest, Random Tree and RepTree, and achieved an accuracy of 95%.

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.

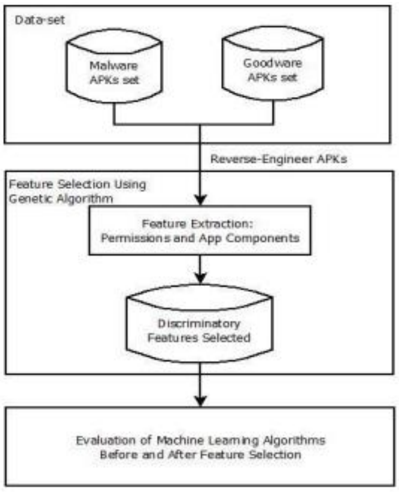

Data-set
Malware APKs set
Goodware APKs set
Reverse-Engineer APKs
Feature Selection Using Genetic Algorithm
Feature Extraction: Permissions and App Components
Discriminatory Features Selected
Evaluation of Machine Learning Algorithms Before and After Feature Selection

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.

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.

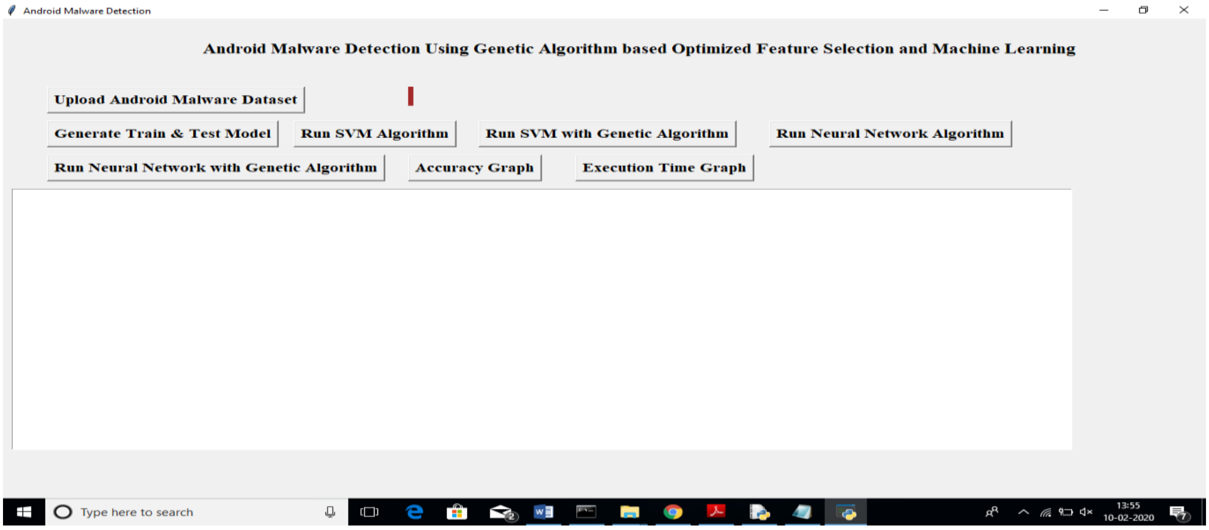
Non-Differentiable and Black-Box Optimization: GAs are particularly valuable in scenarios where the objective function is non-differentiable or the underlying system is a black box, meaning the gradient information is unavailable or unreliable. Since GAs only require the objective function evaluations, 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.

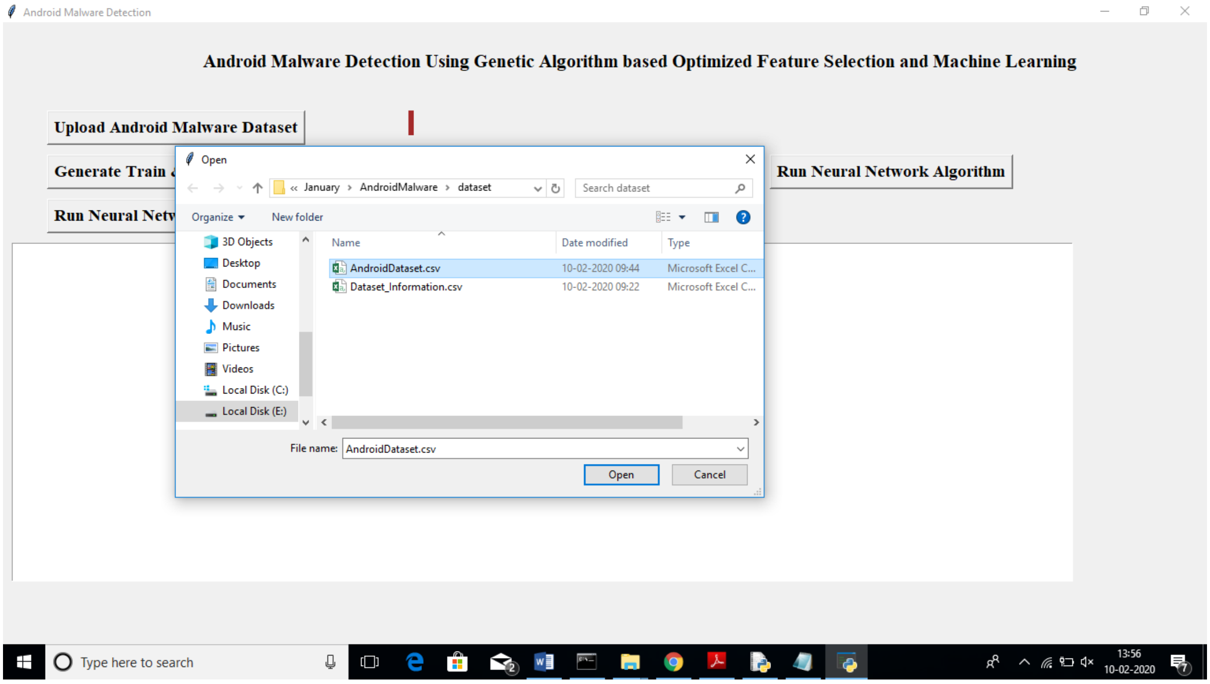
Exploration and Exploitation: GAs strike a balance between exploration (diversification) and exploitation (intensification). Initially, they explore a wide range of solutions to cover the search space. As the algorithm progresses, they exploit promising solutions by recombining and mutating them to refine 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
13:55
10-02-2020

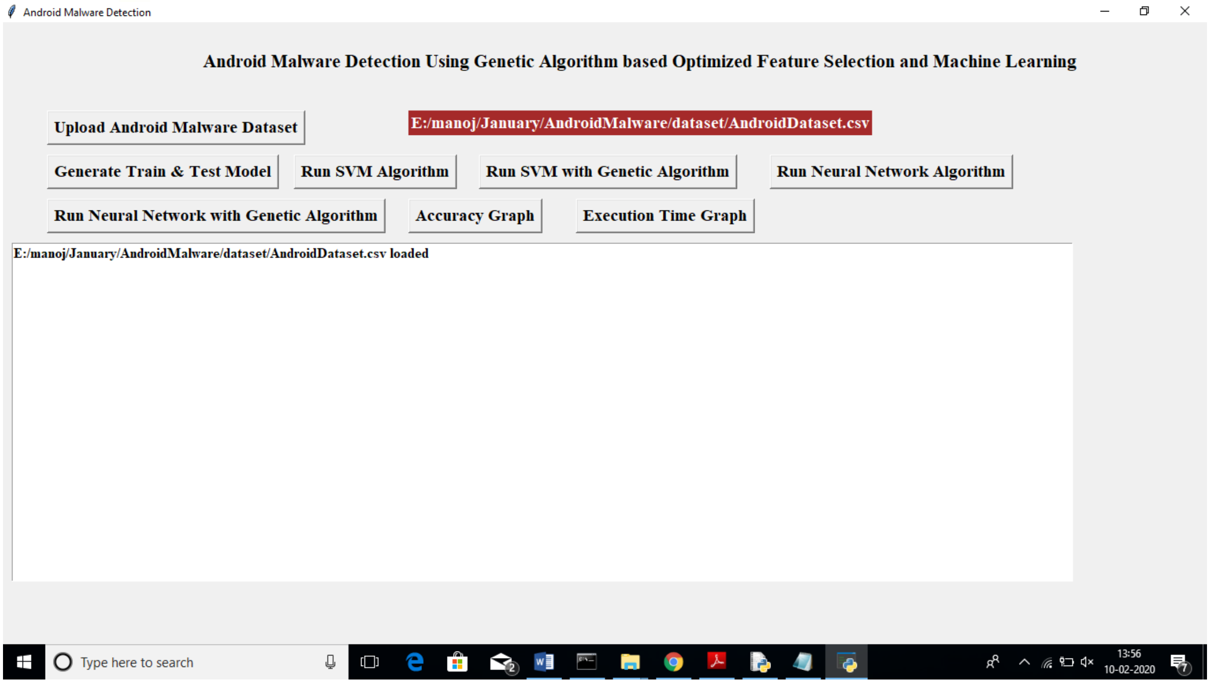
In above screen click on 'Upload Android Malware Dataset' button and upload dataset.


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
Search dataset
Organize
New folder
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
Local Disk (E:)
Open
Cancel
Type here to search
13:56
10-02-2020

Name	Date modified	Type
AndroidDataset.csv	10-02-2020 09:44	Microsoft Excel C...
Dataset_Information.csv	10-02-2020 09:22	Microsoft Excel C...

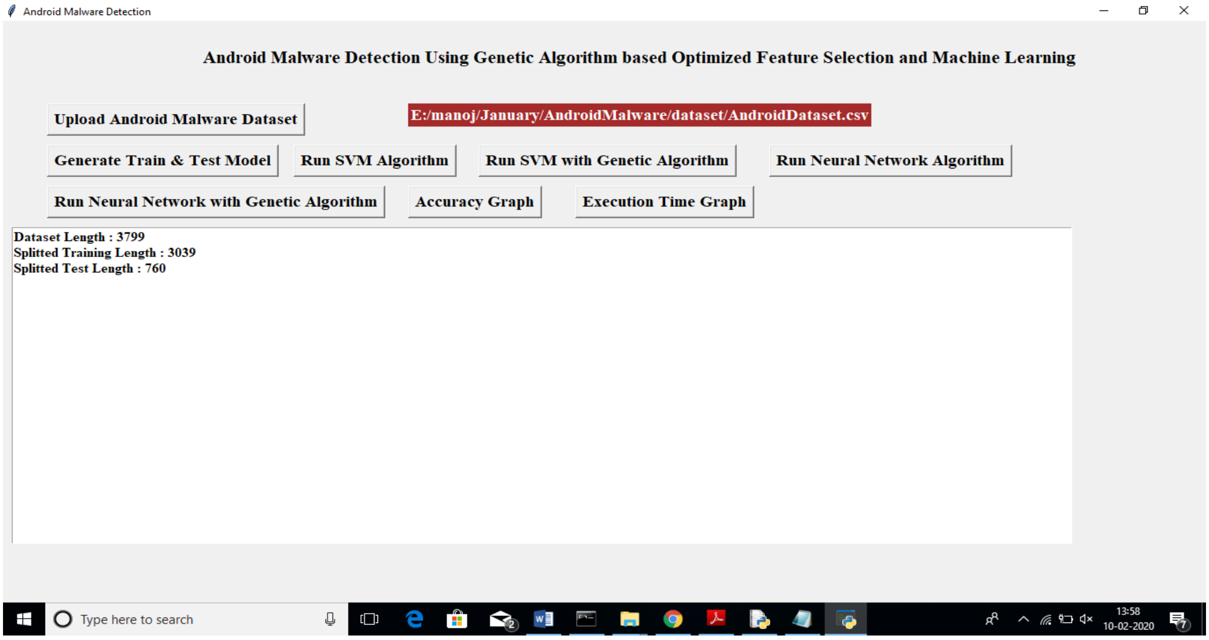
File name: AndroidDataset.csv

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
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
13:56
10-02-2020

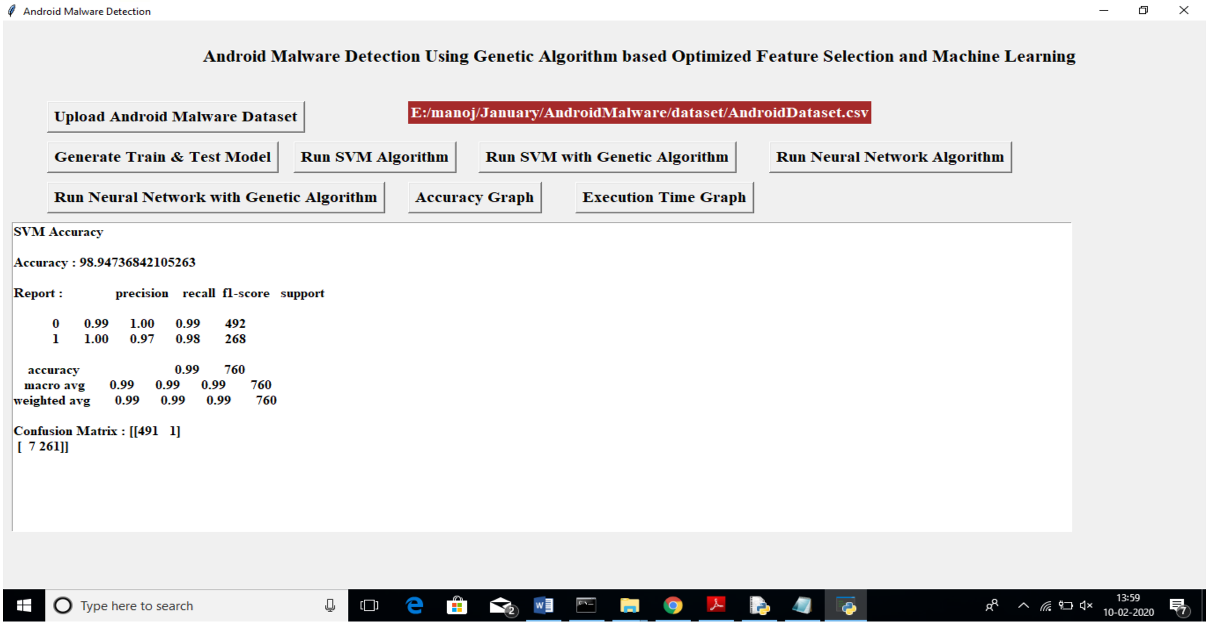
```
E:/manoj/January/AndroidMalware/dataset/AndroidDataset.csv loaded
```

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
13:58
10-02-2020

```
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

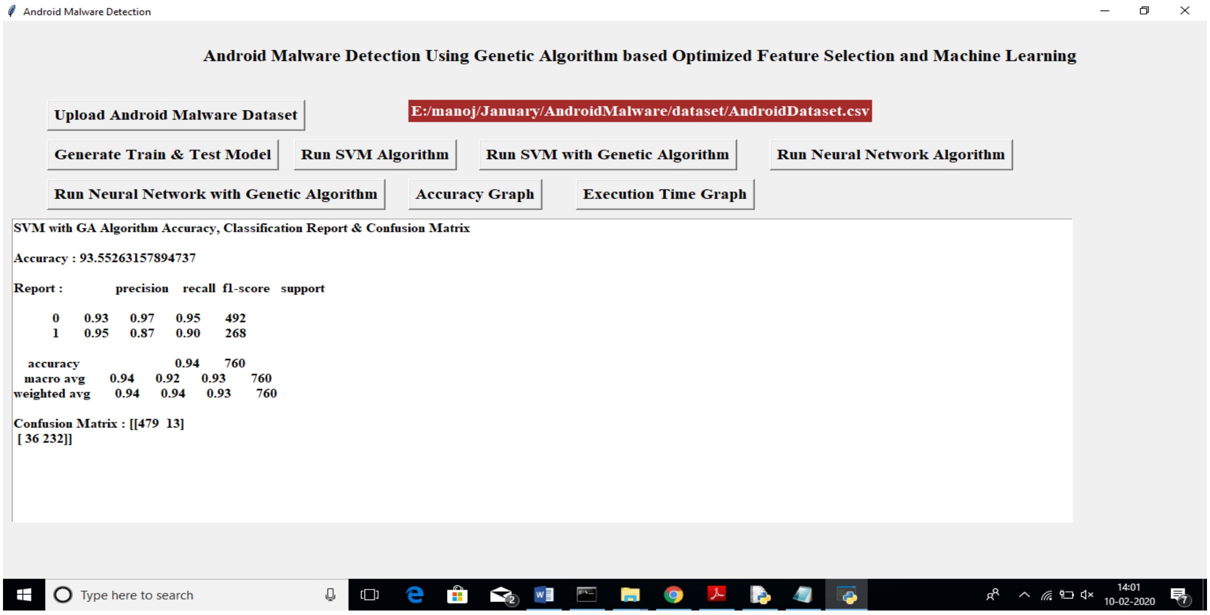

Android Malware Detection
Upload Android Malware Dataset
Generate Train & Test Model
Run SVM Algorithm
Run SVM with Genetic Algorithm
Run Neural Network Algorithm
Run Neural Network with Genetic Algorithm
Accuracy Graph
Execution Time Graph
Type here to search
13:59
10-02-2020

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

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

```
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
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
14:01
10-02-2020

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

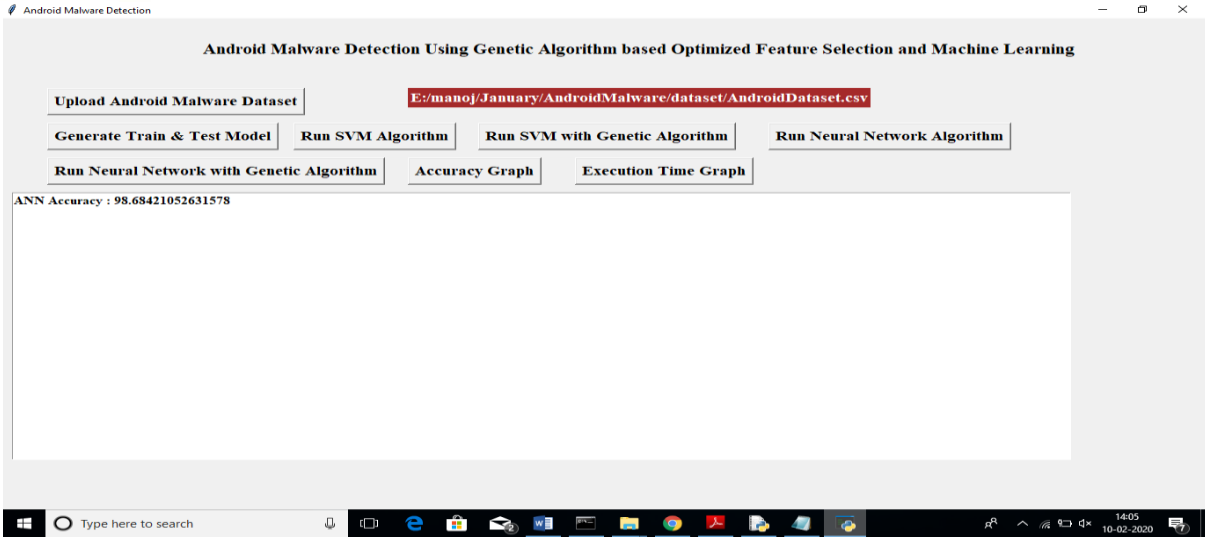
E:/manoj/January/AndroidMalware/dataset/AndroidDataset.csv

```
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.

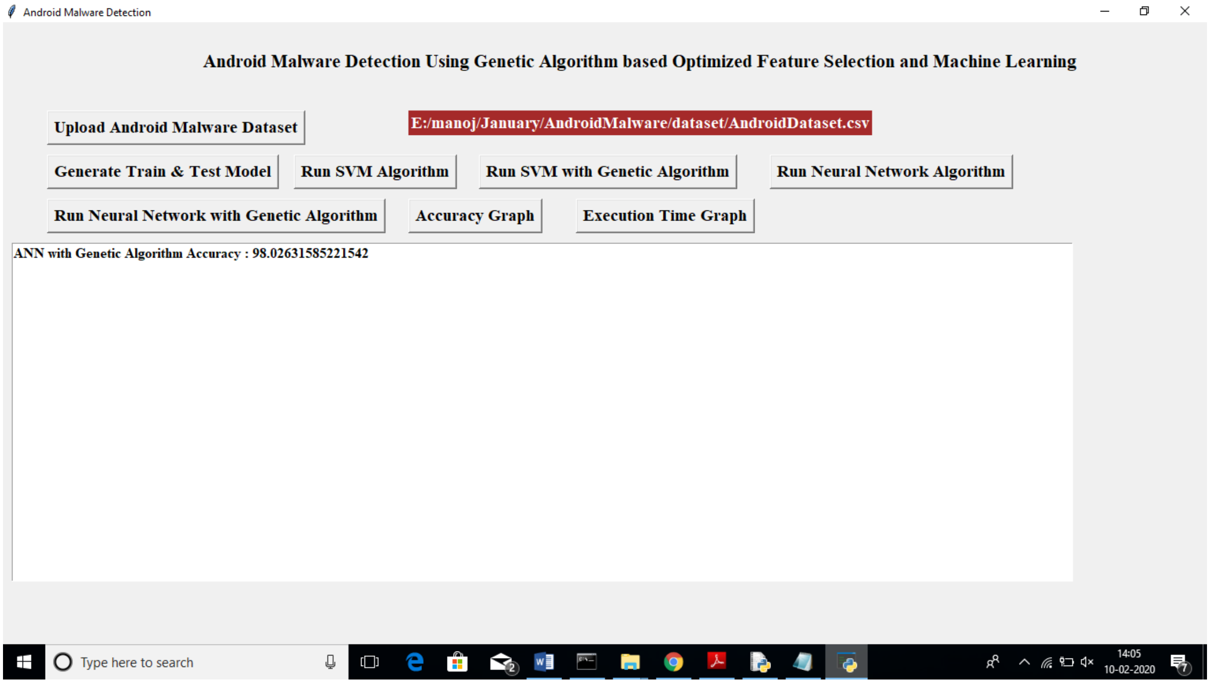
(Note: when u run genetic then 4 empty windows will open u just close all those 4 windows and let main window to run)

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
14:05
10-02-2020

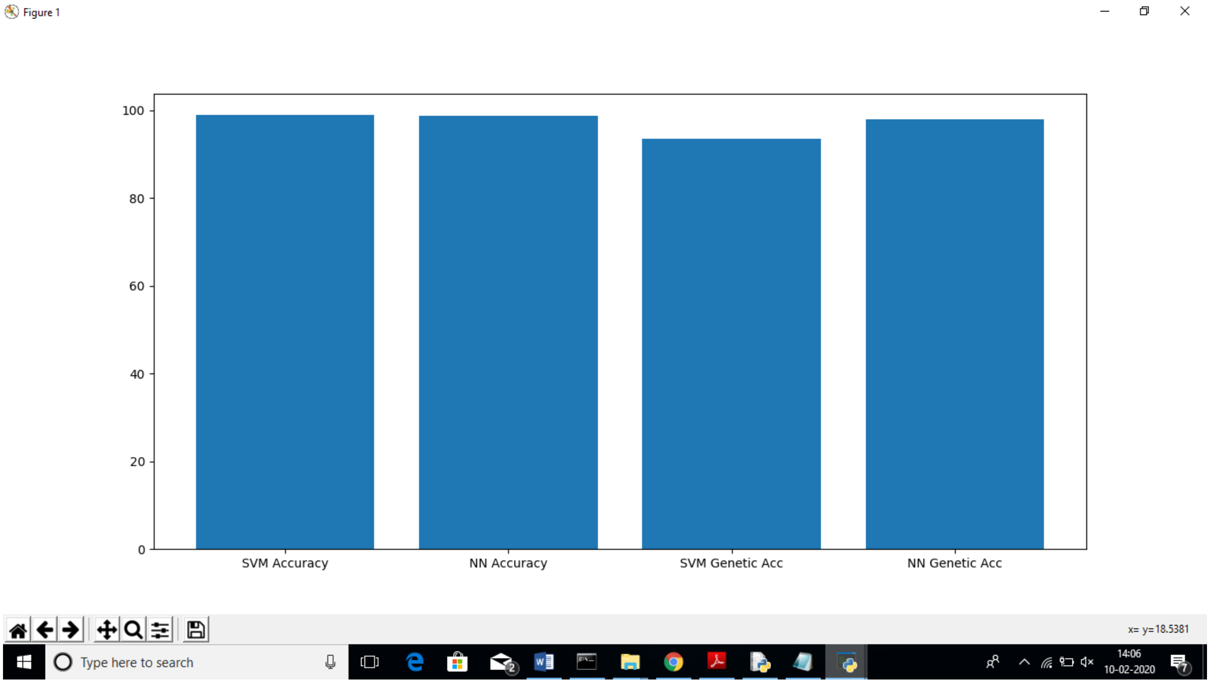
ANN Accuracy : 98.68421052631578

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

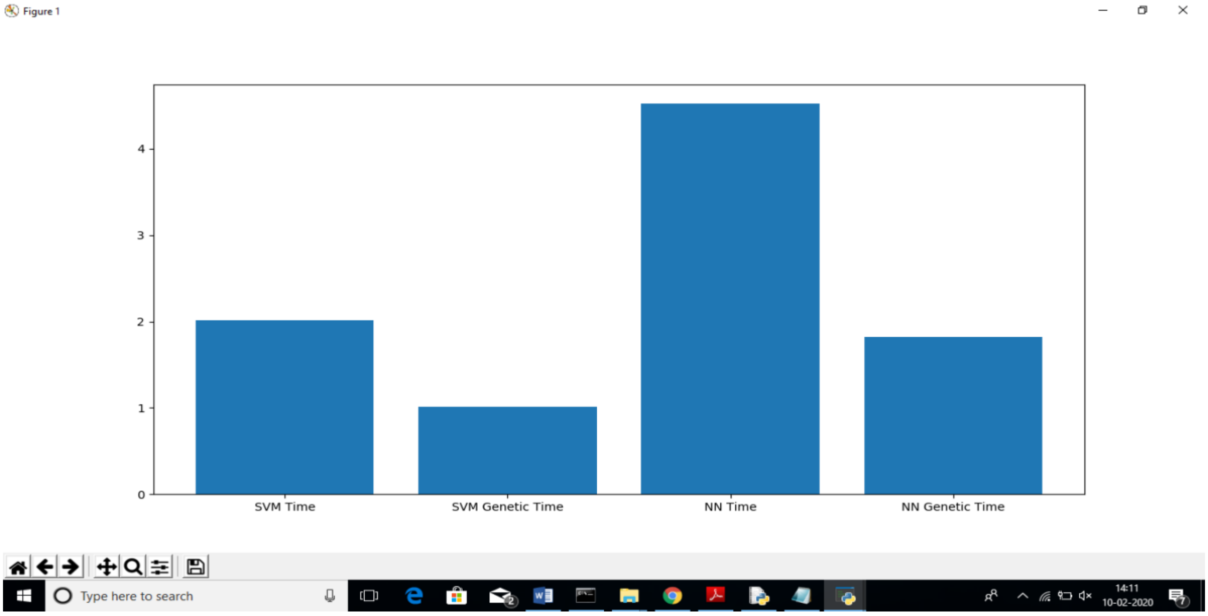

Android Malware Detection
Android Malware Detection Using Genetic Algorithm based Optimized Feature Selection and Machine Learning
Upload Android Malware Dataset
E:/manoj/January/AndroidMalware/dataset/AndroidDataset.csv
Generate Train & Test Model
Run SVM Algorithm
Run SVM with Genetic Algorithm
Run Neural Network Algorithm
Run Neural Network with Genetic Algorithm
Accuracy Graph
Execution Time Graph
Type here to search
14:05
10-02-2020

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.

Sno	Algorithm	Accuracy
1	**SVM with GA**	93%
2	**NN**	98.64%
3	**NN with genetic**	98.2%

4	**SVM**	98%

The Comparative Study show performance metrics before and after feature selection for Support Vector Machine and Neural Network classifiers respectively. As can be observed from ROC curves and performance metrics, both Support Vector Machine and Neural Network when used in conjunction with Genetic Algorithm for feature selection perform significantly well without compromising much in accuracy while working in much reduced feature vector space (less than half of original feature-set), thereby, reducing the classifier training time complexity.

5.CONCLUSION

As the number of threats posed to Android platforms is increasing day to day, spreading mainly through malicious applications or malwares, therefore it is very important to design a framework which can detect such malwares with accurate results. Where signature-based approach fails to detect new variants of malware posing zero-day threats, machine learning based approaches are being used.

The proposed methodology attempts to make use of evolutionary Genetic Algorithm to get most optimized feature subset which can be used to train machine learning algorithms in most efficient way.

From experimentations, it can be seen that a decent classification accuracy of more than 94% is maintained using Support Vector Machine and Neural Network classifiers while working on lower dimension feature-set, thereby reducing the training complexity of the classifiers

Future work can be enhanced using larger datasets for improved results and analyzing the effect on other machine learning algorithms when used in conjunction with Genetic Algorithm.

REFERENCES

[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.

[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.

[6] T. Kim, B. Kang, M. Rho, S. Sezer and E. G. Im, “A Multimodal Deep Learning Method for Android Malware Detection using Various Features”, vol.6013, no. c,2018.

[7] A. Martin, F. Fuentes-Hurtado, V. Naranjo and D. Camacho, “Evolving Deep Neural Networks architectures for Android malware classification”, 2017 IEEE Congr. Evol. Comput. CEC 2017-Proc., pp. 1659-1666, 2017.

[8] X. Su, D. Zhang, W. Li and K. Zhao, “A Deep Learning Approach to Android Malware Feature Learning and Detection”, 2016 IEEE Trust, pp. 244-251, 2016.

[9] K. Zhao, D. Zhang, X. Su and W. Li, “Fest: A Feature Extraction and Selection Tool for Android Malware Detection”, 2015 IEEE Symp. Comput. Commun, pp. 714-720, 4893.

[10] A. Feizollah, N. B. Anuar, R. Salleh and A. W. A. Wahab, “A review on feature selection in mobile malware detection”, Digit. Investig., vol.13, pp. 22-37, 2015.

[11] A. Firdaus, N. B. Anuar, A. Karim, M. Faizal and A. Razak, “Discovering optimal features using static analysis and a genetic search-based method for Android malware Detection”, vol. 19, no. 6, pp. 712-736, 2018.

[12] A. V. Phan, M. Le Nguyen and L.T. Bui, “Feature weighting and SVM parameters optimization based on genetic algorithms for classification problems”, Appl. Intel., vol. 46, no. 2, pp. 455-469, 2017.

AUTHOR PROFILES

Ms.M.ANITHA completed her Master of Computer Applications and Masters of Technology. Currently working as an Assistant professor in the Department of Masters of Computer Applications in the SRK Institute of Technology, Enikepadu, Vijayawada, NTR District. Her area of interest includes Machine Learning with Python and DBMS.

Mr.Y.NAGAMALLESWARA RAO completed his Masters of Technology from JNTUK ,MSC(IS) from ANU,BCA from ANU. He has a

System administrator, Networking administrator and Oracle administrator .He also a Web developer, PHP developer and python developer. Currently working has an Assistant professor in the department of MCA at SRK Institute of Technology, Enikepadu, NTR (DT). His areas of interest include Artificial Intelligence and Machine Learning .

Ms. K. MONIKA RAGAMAI is an MCA student in the Department of Master Of Computer Applications at SRK Institute of Technology, Enikepadu, Vijayawada, NTR District. She has Completed Degree in BSC(Computer Science) Triveni Mahila Degree College, Patamata. Her areas of interest are DBMS, Java Script, and Machine Learning with Python.