ISSN: 2278-4632
Vol-14, Issue-5, May: 2024

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

Ms. CH.Shruthi, Assistant Professor CSE(AI&ML), Vaagdevi College of Engineering (Autonomous), Bollikunta, Khila Warangal (Mandal), Warangal Urban – 506005(T.S)
Dr.B.Sravankumar, Professor CSE(AI&ML), Vaagdevi College of Engineering (Autonomous), Bollikunta, Khila Warangal (Mandal), Warangal Urban – 506005(T.S)
S.Shyam prasad (21645A6618), UG student CSE(AI&ML), Vaagdevi College of Engineering (Autonomous), Bollikunta, Khila Warangal (Mandal), Warangal Urban – 506005(T.S)
CH.Thilak (21645A6606), UG student CSE(AI&ML), Vaagdevi College of Engineering (Autonomous), Bollikunta, Khila Warangal (Mandal), Warangal Urban – 506005(T.S)
CH.Anoop (20641A66D1), UG student CSE(AI&ML), Vaagdevi College of Engineering (Autonomous), Bollikunta, Khila Warangal (Mandal), Warangal Urban – 506005(T.S)
S.Hari prasad (20641A66C7), UG student CSE(AI&ML), Vaagdevi College of Engineering (Autonomous), Bollikunta, Khila Warangal (Mandal), Warangal Urban – 506005(T.S)

ABSTRACT

Ourresearchintroducesanovel andeffectiveapproach tocombatAndroidmalware, apersisten tthreatint hemobiled evicelands cape.Leveraging GeneticAlgorithms, weemployasmarts election process topic koutthemostcrucialfeaturesfordistinguishingbetweenbenignand malicious applications. This, coupled with machine learning models, greatly enhances theaccuracy and robustness of our malware detection system, making it adept at identifying notonly known threats but also emerging ones. Our findings suggest that this approach has thepotential to significantly bolster Android device security by providing a proactive defensemechanismagainsttheevolvingtacticsofmaliciousactors,thusensuringsaferandmoresecuremo bile experiencesforusers.

1.INTRODUCTION

Android Apps are freely available on Google Playstore, the official Android app store as well as third- party app stores for users to download. Due to its open source nature and popularity, malware writers are increasingly focusing on developing malicious applications for Android operating system [1]. In spite of various attempts by Google Playstore to protect against malicious apps, they still find their way to mass market and cause harm to users by misusing personal information related to their phone book, mail accounts, GPS location information and others for misuse by third parties or else take control of the phones remotely. Therefore, there is need to perform malware analysis or reverse-engineering of such malicious applications which pose serious threat to Android platforms. Broadly speaking, Android Malware analysis is of two types: Static Analysis and Dynamic Analysis. Static analysis basically involves analyzing the code structure without executing it while dynamic analysis is examination of the runtime behavior of Android Apps in constrained environment [2]. Given in to the ever-increasing variants of Android Malware posing zero-day threats, an efficient mechanism for detection of Android malwares is required. In contrast to signature-based approach which requires regular update of signature database [3].

2.LITERATURE SURVEY

Android malware detection is a significant area of research, and machine learning techniques have been widely used to address this issue. The most popular machine learning algorithms used for Android malware detection are Support Vector Machines (SVM), Naïve Bayes (NB), Random Forest (RF), and Decision Trees (DT).One of the challenges in Android malware detection is the dynamic nature of malware, which can easily evade detection by traditional signature-based approaches. To address this, researchers have proposed various machine learning-based approaches that can detect

malware based on its behaviour, permissions, and other features.One such approach is the use of API calls and permissions to detect malware. API calls are used by malware to communicate with its command and control servers, and permissions are used to access sensitive data and perform malicious actions. By analysing the API calls and permissions used by an app, machine learning algorithms can detect malware with high accuracy.Another approach is the use of contextual information, such as the app's category, developer, and user reviews, to detect malware. This approach can help detect malware that may not be detected by traditional signature-based approaches.Researchers have also proposed various machine learning algorithms, such as Random Forest, Logistic Regression, SVM, K-NN, and Decision Trees, to detect Android malware. These algorithms have been trained on various features, such as API calls, permissions, and contextual information, to detect malware with high accuracy.The experiments have shown that the proposed models can detect Android malware with high accuracy, outperforming state-of-the-art models. The results also show that the use of contextual information can improve the detection accuracy of machine learning algorithms.Overall, machine learning techniques have shown great promise in detecting Android malware, and researchers continue to explore new approaches and techniques to improve the detection accuracy and efficiency.

3. PROBLEM STATEMENT

Android Apps are freely available on Google Playstore, the official Android app store as well as third-party app stores for users to download. Due to its open source nature and popularity, malware writers are increasingly focusing on developing malicious applications for Android operating system. In spite of various attempts by Google Playstore to protect against malicious apps, they still find their way to mass market and cause harm to users by misusing personal information related to their phone book, mail accounts, GPS location information and others for misuse by third parties or else take control of the phones remotely. Therefore, there is need to perform malware analysis or reverse-engineering of such malicious applications which pose serious threat to Android platforms. Broadly speaking, Android Malware analysis is of two types: Static Analysis and Dynamic Analysis. Static analysis basically involves analyzing the code structure without executing it while dynamic analysis is examination of the runtime behavior of Android Apps in constrained environment [4]. Given in to the ever-increasing variants of Android Malware posing zero-day threats, an efficient mechanism for detection of Android malwares is required. In contrast to signature-based approach which requires regular update of signature database.

4. 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 [5].
- 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.

4.1 Advantages of proposed system:

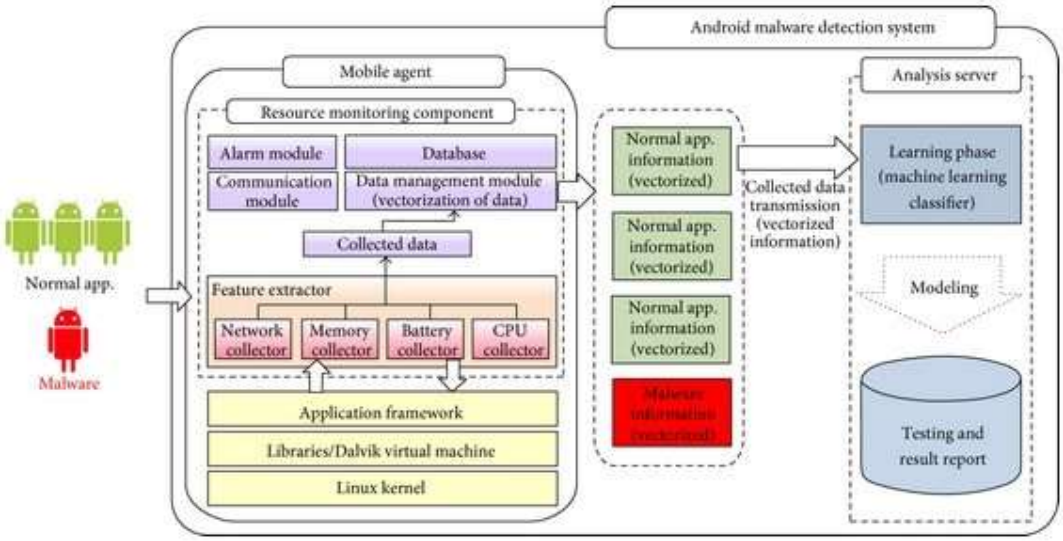
Security

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.

ISSN: 2278-4632
Vol-14, Issue-5, May: 2024

5. SYSTEM ARCHITECTURE



6. IMPLEMENTATION

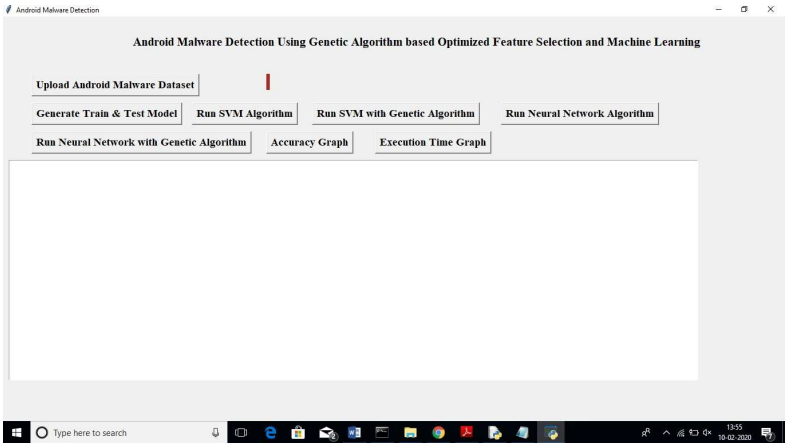
Resource Intensiveness: The resource requirements for maintaining a permissioned blockchain, such as Hyperledger Fabric, can be substantial. This could pose a limitation for organizations with limited computational resources, potentially hindering the widespread adoption of BBS, especially in resource- constrained environments.

Learning Curve and Implementation Complexity: Implementing and managing a Hyperledger Fabric- based system like BBS may involve a steep learning curve for administrators and users unfamiliar with blockchain technology. The complexity of configuring and maintaining the system could be a limiting factor, potentially slowing down the adoption process.

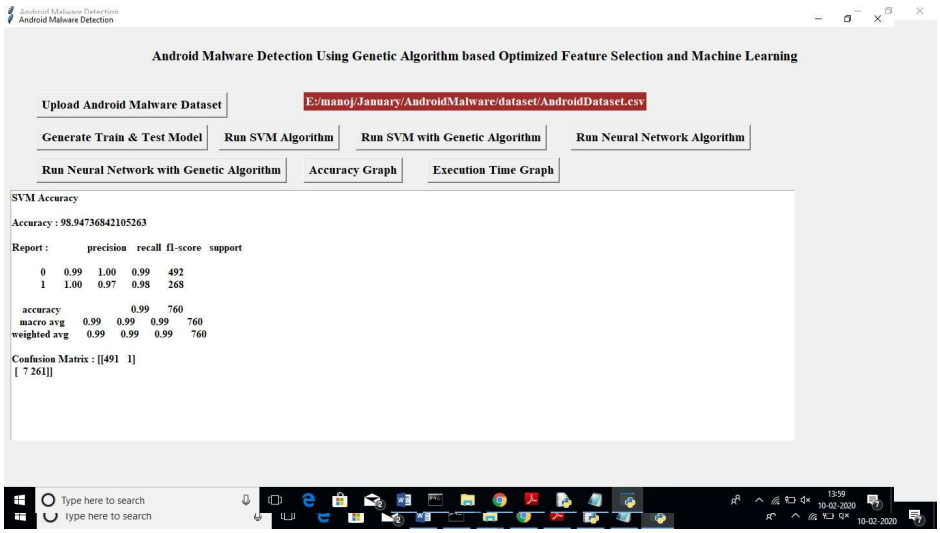
Dependency on Hyperledger Fabric: BBS's reliance on Hyperledger Fabric as the underlying blockchain framework introduces a dependency on the development and maintenance of the Hyperledger ecosystem. Changes or vulnerabilities in Hyperledger Fabric could directly impact the functionality and security of BBS, making it susceptible to external factors beyond its immediate control.

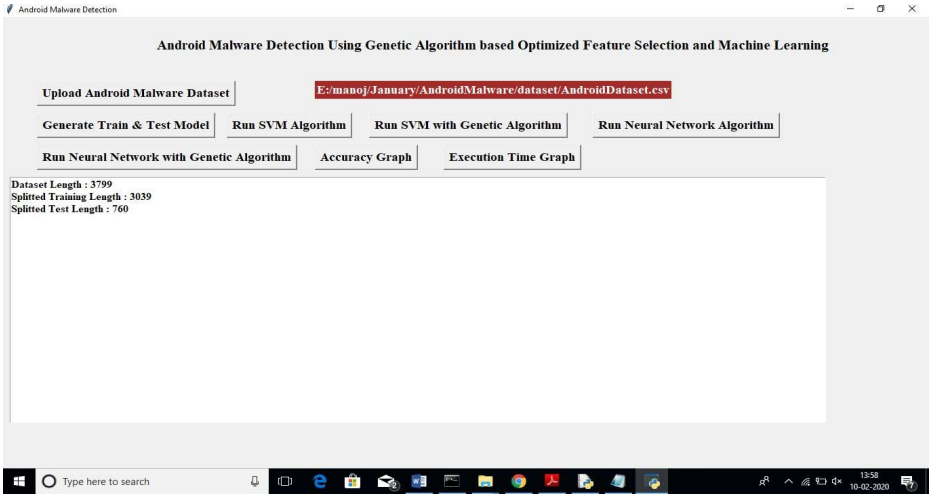
Limited Interoperability: Interoperability with other existing systems and non-blockchain databases might be a challenge. BBS may face limitations in seamlessly integrating with diverse data storage and management solutions commonly used in various organizations, potentially restricting its applicability in heterogeneous computing environments.

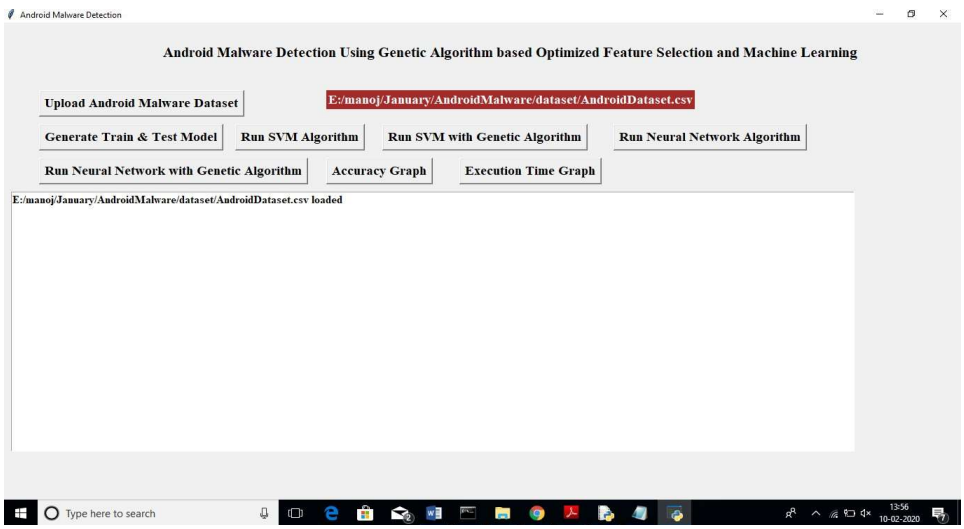
7. OUTPUT RESULT



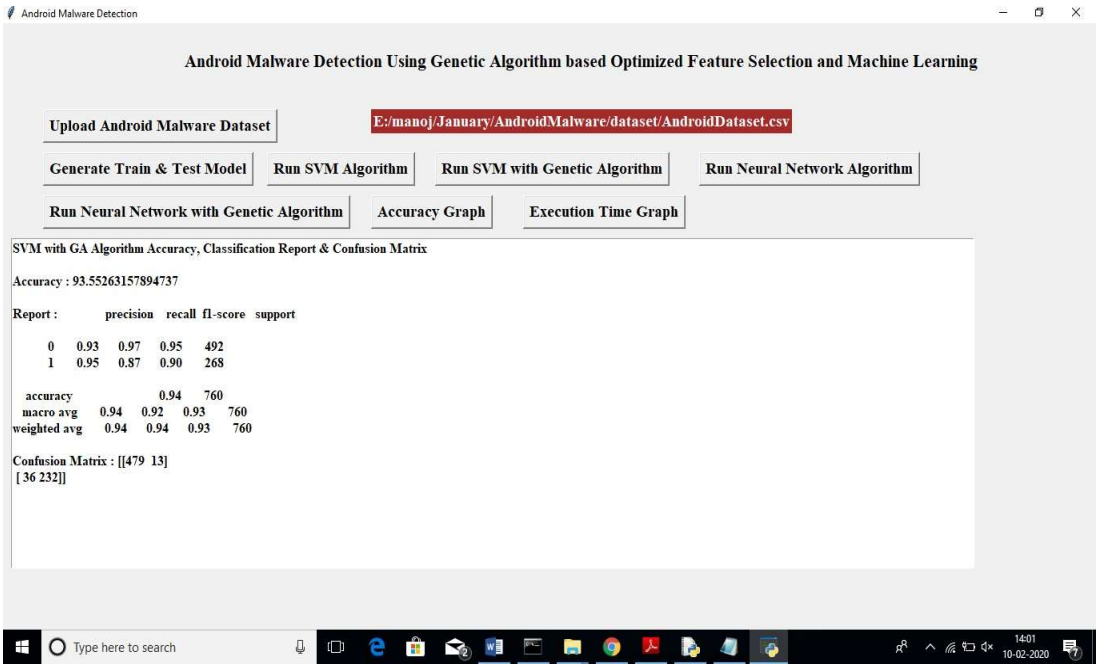
ISSN: 2278-4632
Vol-14, Issue-5, May: 2024

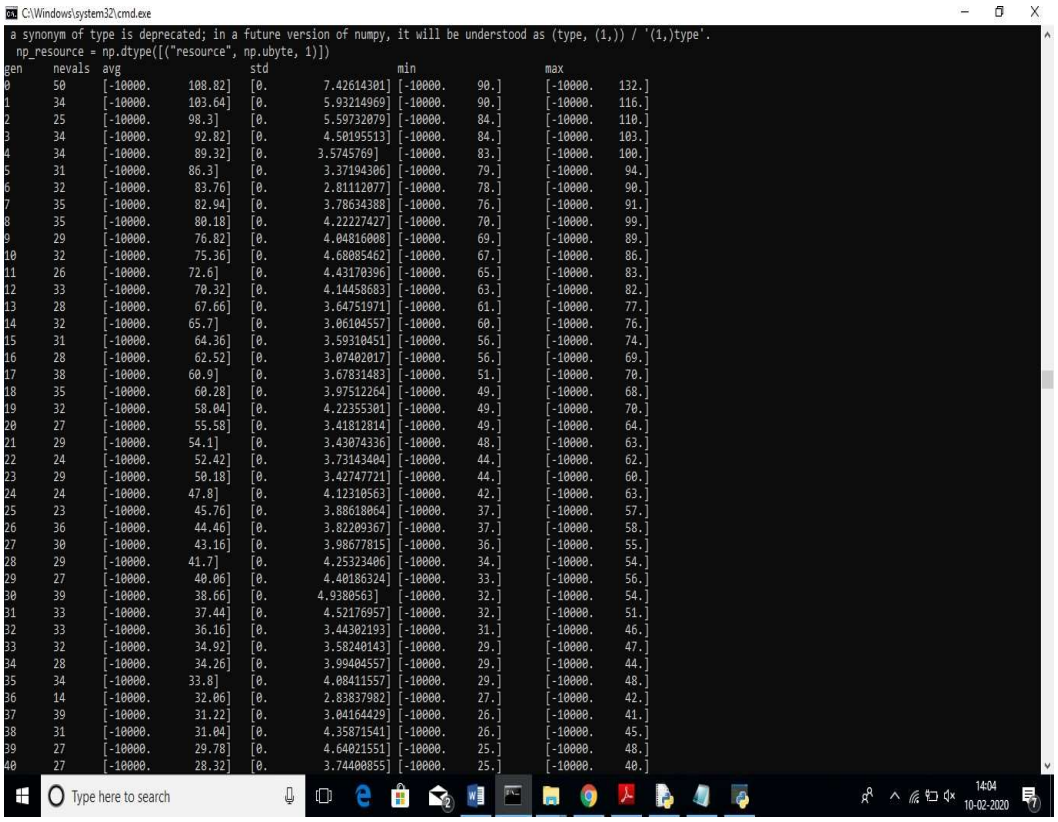




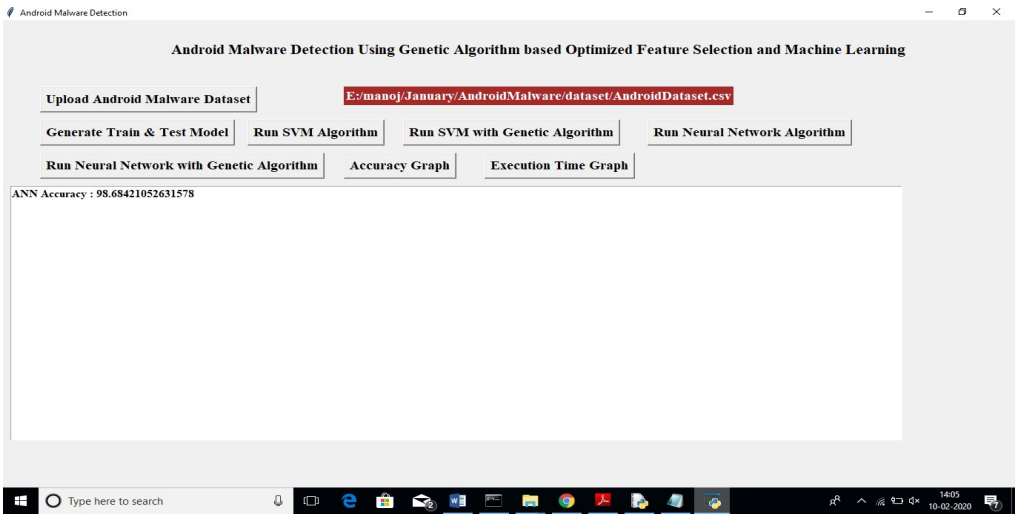


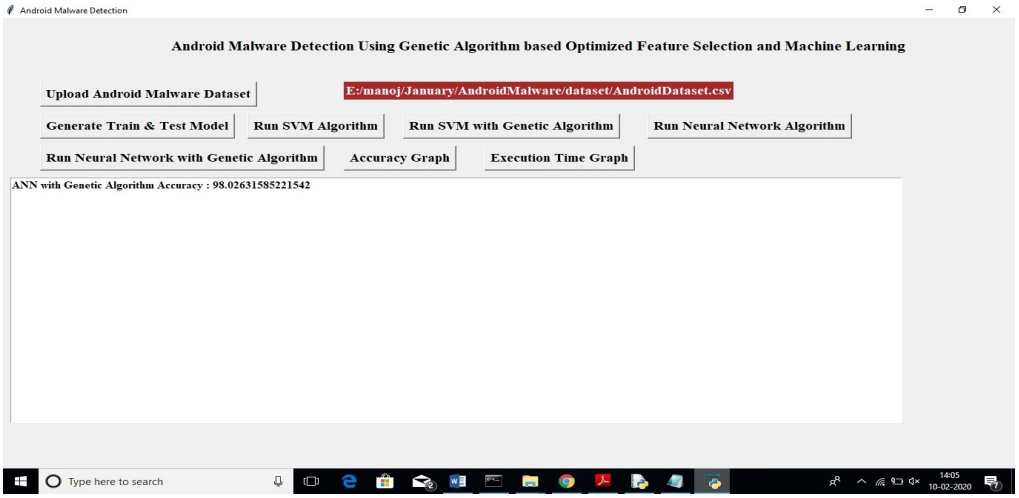
ISSN: 2278-4632
Vol-14, Issue-5, May: 2024

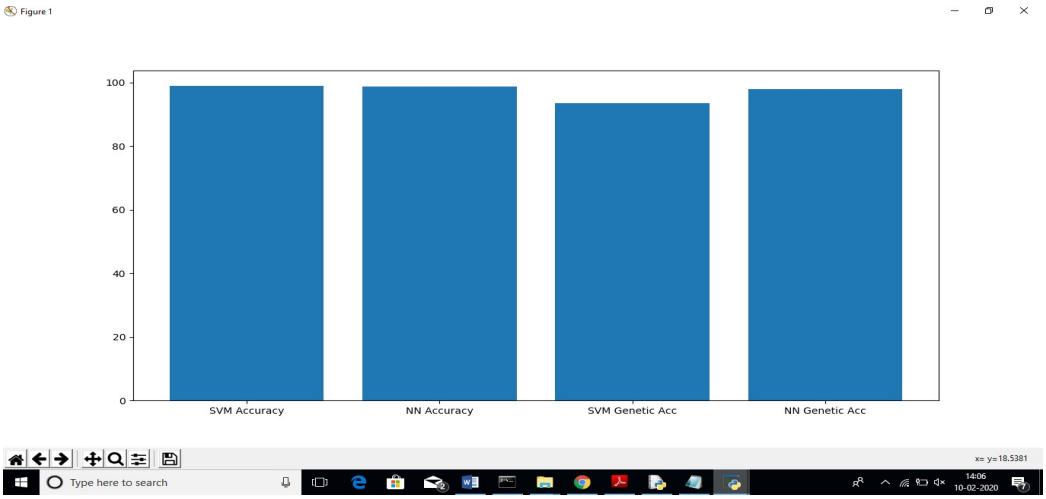




ISSN: 2278-4632
Vol-14, Issue-5, May: 2024







8. 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.

9. FUTURE SCOPE

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 Further 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.

10. 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.

