International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 08 | August - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

Malware Detection A Framework for Reverse Engineered Android Application through Machine Learning Algorithms

Mr. Shivakumara T[1], Pallavi M[2]

[1]*Assistant Professor, Department of Master of Computer Application, BMS Institute of Technology and Management, Bengaluru, Karnataka*

[2]*Student, Department of Master of Computer Application, BMS Institute of Technology and Management, Bengaluru, Karnataka*

Abstract - This smartphone operating system is rapidly gaining popularity. Consequently, Android has emerged as an attractive focus for malicious attackers. They are concealing harmful algorithms in complex ways within Android apps, making it challenging for security firms for the purpose of recognizing and categorizing these apps as malware. The evolution pertaining to Android malicious software has reached a point where it can avoid typical detection methods due to its uniqueness. Machine learning-based approaches have surfaced as a more effective solution to address the issue complexity of emerging Android threats. These approaches the actions exhibited by existing malware patterns and use this data for the purpose of differentiation between known dangers and new risks. This study focuses on identifying vulnerabilities in mobile apps by utilizing Backward Designed Android.

Key Words: SVM, AdaBoost, Ransomware, Android, Machine Learning

1. INTRODUCTION

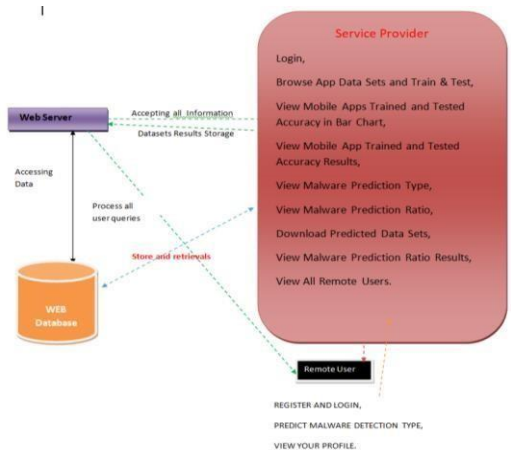
At this stage, smartphones and tablets are becoming an essential an integral aspect of the daily routines for many individuals. Additionally, the Android operating system currently holds a strong position in the mobile device market, accounting for about 80% of global sales on average in recent years.

As Android becomes more widely used on a diverse range of devices across the globe, the presence of. Malicious software designed for Android devices has also increased. This is due to the fact that Android functions as an opensource OS, which means the potential risk grows as hackers and developers insert undesired permissions, features, and app elements into Android apps. While the ability to enhance. its potential through the integration of external applications is appealing, it introduces the potential danger of malicious attacks. The security concern of unauthorized access to various sensitive information rises as the number of smartphone apps increases. Consequently, apps become
more vulnerable, leading to activities like collecting private data, engaging in SMS fraud, and distributing malware.

Unlike traditional assessment methods that involve examining smartphone Manifest.xml, source files, and the Dalvik bytecode Code, machine learning takes a different approach. It involves grasping the fundamental behaviors of both beneficial and harmful app scenarios subsequently employing data to enable learning. Static characteristics extracted from an application are widely applied in machine learning models methods, and this labor-intensive process can be simplified by obtaining and utilizing static characteristics from various smartphone configurations. These attributes aid in training a system to predict viruses utilizing machine learning techniques

strategies like Support Vector Machine (SVM), financial development, ensemble learning, and other techniques.

Machine learning harnesses diverse methods to categorize data. SVM stands out as a powerful learner, representing each data instance as a node within multi-dimensional space, where the vector's magnitude indicates feature values. Sorting is achieved by selecting the best hyperplane that separates the two categories thereby enhancing the recognition of variables. When combined with other machine learning algorithms, techniques like boosting or ensemble methods, such as Ada boost, assign greater importance to inaccurately categorized elements. Utilized alongside underperforming classifiers, these techniques enhance our initial model since they exhibit a high level of accuracy or categorization. more vulnerable, leading to activities like collecting private data, engaging in SMS fraud, and distributing malware.



Fig[1] : System Planning

Machine learning harnesses diverse methods to categorize data. SVM stands out as a powerful learner, representing each data instance as a node within multi-dimensional space, where the vector's magnitude indicates feature values. Sorting is achieved by selecting the best hyperplane that separates the two categories thereby enhancing the recognition of variables. When combined with other machine learning algorithms,
techniques like boosting or ensemble methods, such as Ada boost, assign greater importance to inaccurately categorized elements underperforming classifiers, these techniques enhance our initial model since they exhibit a high level of accuracy or categorization.

2. LITERATURE SURVEY

The approaches [1] outlined in this linked article make significant contributions to vital aspects in the context of identifying and detecting malware an improved rate of prediction. Numerous studies have concentrated on The notion of utilizing larger datasets has been widely explored, with various [2] studies incorporating multiple
sets of attributes to enhance the efficiency of detection rates. In the discussed article.

two datasets were employed, totaling 700 instances of malware and 160 attributes. Employing the Random Forest (RF) Algorithm, both datasets achieved an accuracy of about 91%. In a more extensive evaluation involving 5,560 malware samples, the model successfully identified 94% of malware instances while minimizing false alarms. Moreover, this approach outperformed static and dynamic solutions relying on system calls. Over nine years, researchers consistently demonstrated the model's exceptional classification performance, surpassing state of- the-art methods in both static and dynamic techniques. These evaluations encompassed interlinked experiments featuring robust infections from diverse sources. The model consistently achieved a 97% F1-measure validity for program recognition and malware classification.

[5] Authors introduced an innovative method for Android malware detection termed Systems for Detecting Malware Based on Permissions (PMDS). This approach involves assessing 2950 instances of legitimate and harmful Android apps. In the PMDS method outlined in another reference, app permissions are treated as behavioral indicators. Leveraging these indicators, a model based on machine learning developed to anticipate potential harmful behavior in unfamiliar applications built upon the combination of permissions required. Exhibiting a rate of false positives

UTILIZING MACHINE LEARNING ALGORITHMS AND ENSEMBLE LEARNING are not incorporated in the framework. The system has not been deployed. Features of Backward Designed Solutions.

3. PROPOSED SYSTEM

We introduce a fresh subset of attributes designed for statically detecting fraudulent Android applications. This subset encompasses seven meticulously chosen feature sets, incorporating approximately 56,000 attributes from these categories. Rigorous testing involving more than 500,000 benign and malicious Android apps, along with the most extensive malware sample set compared to any existing method, demonstrates a noteworthy 96.24%improvement in recognition accuracy with a mere 0.3% occurrence of false positives.

For model training, we employed the new attributes across six machine learning classifier models algorithms. Additionally, we applied a Boosted ensemble learning technique, AdaBoost, along incorporating a Decision Tree rooted in binary classification, to elevate. the accuracy of our predictions. Our model was trained using the latest and comprehensive malware samples collected from recent years.

The selection of characteristics in our proposed system is predicated on their capacity to encompass all data sets. This approach not only streamlines the data set but also reduces the time required for categorization, offering a more efficient function selection strategy.
It's worth noting that our study employs expanded attributes for categorization While this represents a common consideration in machine learning, the choice

of the framework for the detection or categorization model could wield a significant influence, particularly when dealing with vast datasets.

4. IMPLEMENTATION

MODULES:

- **Service Provider**
- **View and Authorize Users**
- **Remote User**

- **Service Provider**

To access this module, service providers need to use valid login credentials, which include a correct username and password. Once logged in successfully, they have the capability to engage within a variety of activities. These tasks encompass logging in, reviewing datasets associated with applications, conducting training and evaluation procedures. Additionally, they can visualize the precision of mobile app education and assessment outcomes using a bar chart, view the outcomes from accuracy assessments, explore the type of malware predictions made, analyze the ratio of malware predictions, and download projected datasets. Furthermore, the functionality enables the examination of malware prediction ratios for all remote users.

- **View and Authorize Users**

The admin has the capability to access a comprehensive list of all users registered within this module. The admin is authorized to review the user's details, including their username, email, and address. Furthermore, the admin holds the authority to grant approval to users.

- **Remote User**

Within this module, there exists a varying number of users. Prior to participating in any actions, users are required to complete a registration process. Upon registering, the details provided by the user are documented in the database. Subsequently, the user must employ their authorized username and password for successful login. Once logged in, the user gains the ability to engage in tasks such as registration and login procedures, predicting malware detection types, and exploring their own profile.

5. CONCLUSION

In this study, we developed a methodology for detecting harmful Android apps Our proposed approach integrates various machine learning attributes and achieves an impressive 96.24% precision in detecting malicious applications within the Android platform. Our initial steps involve devising strategies. to capture and examine the conduct of Android apps. Through techniques similar to reverse app engineering and Andro Guard, we. Paraphrasing content represent them as binary vectors. Subsequently, we utilize Python build tools and partitioning techniques. For model training using both non-harmful and harmful data.

By incorporating improved feature sets and expanding the scope of our experiments, we observe. that the model we propose maintains a low 0.3%. rate of incorrect positive identifications and achieves a 96% accuracy within the current context. Moreover, our research indicates that ensemble methods and robust student algorithms outperform alternative techniques when dealing with high dimensional classifications and data. However, it's important to acknowledge that our proposed approach does have limitations. It is constrained in static analysis terms, exhibits a deficient factors that ensure long-term viability, and encounters challenges related to collinearity issues In times ahead, we aim to enhance the model's robustness by incorporating factors related to growth and dynamics. Addressing concerns surrounding dependent variables and high inter-associations will be crucial before applying machine learning techniques.

6. REFERENCES

[1] A. O. Christiana, B. A. Gyunka, and A. Noah, "Android Malware Detection Through Machine Learning Techniques: A Review," doi: 10.3991/ijoe.v16i02.11549. The International Journal of Internet Medical Engineering, vol. 16, no. 02, p14, Feb. 2020.

[2] Ghimire, D., and J. Lee, "Geometric Feature-Based Expression of Face Detection in Image Arrangements Utilizing Multi-Class AdaBoost and Support Vector Machines," Sensors, vol. 13, no. 6, June 2013, pp. 7714-7734, doi:10.3390/s130607714.

[3] Authors A. Garg and K. Tai, "Comparison of statistical and machine learning methods in data modelling with multicollinearity," Int. J. Model. Identif. Control, vol. 18, no. 4, p. 295, 2013, doi:10.1504/IJMIC.2013.053535.

[4] Authors C. P. Obite, N. P. U. Ugwuanyim, and D. C. Bartholomew, "Multicollinearity Effect in Regression Analysis: A Feed Forward Artificial Neural Network Approach," Asian J. Probab. Stat., vol. 6, no. 1, January 2020, pp. 22-33, doi: 10.9734/ajpas/2020/v6i130151.