INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

www.ijprems.com
editor@ijprems.com

Vol. 04, Issue 07, July 2024, pp: 1373-1381

e-ISSN : 2583-1062

Impact Factor: 5.725

ENHANCING ANDROID MALWARE WITH MACHINE LEARNING AND DIMENSIONALITY REDUCTION TECHNIQUE

Aakansha[1], Dr. Nitin Kumar[2]

[1]M Tech Scholar, Ganga Institute of Technology & Management Kablana Jhajjar, India.

[2]Associate Professor, Ganga Institute of Technology & Management Kablana Jhajjar, India.

ABSTRACT

The proliferation of Android smartphones has resulted in a surge of malware. Conventional detection technologies face difficulties in dealing with constantly changing threats. This thesis investigates the utilisation of artificial intelligence and dimensionality reduction techniques to improve the accuracy of malware detection. Artificial Intelligence, particularly machine learning and deep learning, is capable of identifying patterns even in novel forms of malware. However, the presence of a large number of variables often leads to overfitting and significant computing expenses. This work enhances model efficiency by implementing dimensionality reduction techniques such as PCA, LDA, and t-SNE, which effectively compress the feature space while preserving essential information. The project gathers both harmless and harmful Android applications, extracts their characteristics, using dimensionality reduction techniques, and trains artificial intelligence models such as Support Vector Machines (SVM), Random Forest (RF), and Convolutional Neural Networks (CNN). The findings indicate that the integration of artificial intelligence with dimensionality reduction enhances both the precision and efficiency of the models, rendering them appropriate for use on mobile devices in real-time scenarios. This method improves cybersecurity by providing more flexible and efficient security solutions to safeguard mobile environments. The results highlight the capacity of these technologies to offer strong malware detection systems.

Key Words: Android Malware, Artificial Intelligence (AI), Machine Learning (ML), Deep Learning (DL), Dimensionality Reduction, Principal Component Analysis (PCA), Linear Discriminant Analysis.

1. INTRODUCTION

The extensive implementation of the Android operating system has revolutionised mobile computing. However, it has also led to a significant increase in malware specifically designed to exploit this platform. This poses serious threats such as unauthorised access to sensitive data and potential financial losses. Conventional methods of detecting malware based on signatures face difficulties in dealing with the constant changes and adaptations made by malware. Artificial Intelligence (AI), which encompasses Machine Learning (ML) and Deep Learning (DL) algorithms, presents a highly promising approach by leveraging huge datasets to identify previously unseen patterns of hostile behaviour. Dimensionality reduction techniques, such as PCA, LDA, and t-SNE, are used to tackle the complexity of malware detection and enhance efficiency. This thesis combines artificial intelligence (AI) with dimensionality reduction techniques to boost the detection of Android malware. The goal is to greatly enhance the accuracy and efficiency of malware detection in real-time mobile contexts.

Android

Android is an open-source mobile operating system developed by Google. It is based on the Linux kernel and is designed for touch devices such as smartphones and tablets. The architecture of the system consists of the Linux kernel, device abstractions, native libraries, and the Android Application Runtime (ART) for executing applications. Development tools such as Android SDK and Android Studio provide support for Java and Kotlin in order to create applications. These tools offer a user-friendly interface that adheres to Material Design principles, allowing for intuitive interaction.

Android Market Growth

Google launched Android in 2008, and it has since become the leading operating system in the worldwide smartphone industry. This is because Android is open-source, which means it can be customised, and it is available on a wide range of devices at different price points. Additionally, Android has a vast app ecosystem through Google Play. The wide range of devices it supports, including tablets, wearables, TVs, and automobiles, along with regular upgrades and new features, contributes to its continued growth and influence in the industry.

Android Malware

Android malware refers to a specific category of harmful software designed to target devices running on the Android operating system. This malware has the ability to compromise user data, impair the functionality of the device, and carry out unauthorized acts. Typical examples encompass trojans, ransomware, spyware, and adware. Android's open architecture and the wide range of app providers make it a common target for cybercriminals.

Types of Malwares

Adware

Adware is software that presents advertising on an individual's gadget. typically, in a way that is intrusive or unwanted. Adware is a type of software specifically created to exhibit undesirable adverts on a user's device, typically appearing within the browser or other programs. The developer earns cash through ad clicks or impressions. While not necessarily intentionally harmful, adware can negatively impact the performance of a device, drain network resources, and raise privacy concerns by monitoring user activities.

Spyware

Spyware is a type of malicious program that is specifically created to covertly collect data from a device or computer outside the user's awareness or permission. Spyware is a form of malicious software that covertly surveys and gathers user data, including internet browsing patterns, typed keystrokes, and personal information, without obtaining permission. These data are frequently transmitted to third parties with malicious intent, such as for the purposes of identity theft or financial crime. Spyware poses a substantial threat to consumer privacy and security.

Trojan

A Trojan is a form of malevolent software that masquerades as a genuine program with the intention of illicitly infiltrating a computer system. A trojan, also referred to as a Trojan horse, is a form of malicious software that disguises itself as legitimate software." Upon installation, it is capable of executing a range of malicious actions, including data theft, installation of supplementary malware, and establishment of remote access backdoors. Trojans exploit user deceit as a means of spreading and have the potential to inflict significant harm on computer systems and data.

2. OBJECTIVES

- Assess the constraints of signature-based approaches in identifying emerging Android malware risks.
- Investigate artificial intelligence (AI), specifically machine learning (ML) and deep learning (DL), to determine its capacity to detect malware patterns and behaviours from extensive datasets.
- Apply dimensionality reduction techniques such as PCA, LDA, and t-SNE to improve the efficiency and effectiveness of feature extraction.
- Combine AI models with dimensionality reduction techniques to create a resilient detection framework that can effectively handle both familiar and new forms of malware.
- Evaluate and contrast the efficacy of artificial intelligence models such as Support Vector Machines (SVM), Random Forest (RF), and Convolutional Neural Networks (CNN) by utilizing feature sets that have been reduced in size.
- Provide novel perspectives and approaches to enhance cybersecurity, offering practical suggestions for detecting malware in real-time on Android devices.

3. LITERATURE REVIEW

Şahin et al. (2023) present a sophisticated method for detecting Android malware by employing linear regression to determine permission-based criteria. The research aims to tackle the crucial issue of differentiating among malicious and benign software by specifically examining the permissions sought during the installation process. The work intends to improve the precision of malware detection systems by using linear regression to improve the collection of authorization characteristics that indicate illicit behaviour. This approach emphasizes the need of accurate feature selection in efficiently reducing security risks on Android devices. The suggested technique enhances the effectiveness of detecting malware threats in the mobile ecosystem by carefully analysing permission patterns. Additionally, it aids in the building of strong frameworks to protect users' digital assets from developing malware threats.

2023, Arslan unveiled FG-Droid, an innovative method for detecting Android malware that prioritizes efficiency and performance by utilizing characteristic grouping and machine learning. FG-Droid intends to optimize the feature selection process by minimizing the number of characteristics utilized for recognizing malicious software. The strategy improves the interpretability and computational efficiency of the detection model by combining related characteristics, without compromising on precision. This innovative methodology tackles the intricacy of analysing Android malware by enhancing the way app behaviours and properties are represented. FG-Droid utilizes machine learning approaches to enhance malware detection and facilitate the creation of scalable and adaptable systems that can effectively counter emerging threats in the ever-changing mobile environment. Arslan's work contributes to the advancement of the cybersecurity profession by providing a well-organized framework for improving the strength and effectiveness of tactics used to identify and mitigate Android malware.

Feizollah et al. (2022) conducted an extensive review that specifically examined strategies for selecting attributes in the detection of mobile malware. Their study rigorously assesses different techniques designed to optimize feature sets in order to improve the precision and effectiveness of detection systems. Feizollah et al. emphasize the significance of choosing appropriate characteristics that can successfully distinguish among benign and malicious mobile applications, based on their analysis of existing research. The paper highlights various methodologies, including statistical analysis, machine learning algorithms, and heuristic-based methods, that are specifically designed to tackle distinct obstacles in malware identification. The thorough analysis highlights the importance of strong feature selection in enhancing the overall effectiveness and dependability of mobile malware detection systems. The findings of Feizollah et al. offer useful insights into current practices and methodology, which can lead to breakthroughs in designing more effective and adaptable approaches to address the changing landscape of mobile threats.

Li et al. (2022) concentrates on identifying important permissions in Android malware detection. They use machine learning methods to distinguish crucial permissions that suggest possible dangerous activities in applications. Their work highlights the significance of identifying licenses that have a critical role in differentiating among harmless and harmful applications, therefore enhancing the precision of detection. Li et al. seek to enhance the identification process of essential permissions linked to criminal actions by utilizing machine learning algorithms, including feature selection and classification models. This approach enhances the capacity to detect and mitigate security threats produced by malicious software for Android in an easier way. This research enhances the area by offering valuable insights into the utilization of machine learning for the prioritization and analysis of permissions that are crucial for discovering harmful behaviour in mobile applications.

Pan et al. (2021) performed a comprehensive analysis of existing literature, specifically investigating techniques for identifying Android malware using static assessment approaches. Their report provides a concise overview of the progress and obstacles related to the application of static analysis techniques for detecting malicious software on Android devices. Pan et al. examine previous research to emphasize the development of static analysis tools and approaches. They emphasize the efficacy of these tools in identifying malware through code and binary analysis, without the need to execute the apps. The review emphasizes the significance of static analysis in reducing security risks caused by Android malware. It offers valuable information on existing practices and suggests future research areas to improve the ability to detect threats in the ever-changing field of mobile cybersecurity.

Pehlivan et al. (2021) examine techniques for selecting attributes and algorithms for classifying Android malware based on permissions. Their study examines different methodologies targeted at enhancing the precision and effectiveness of identifying malicious software by analysing the permissions sought during the installation process. Pehlivan et al. intend to enhance the detection process and reduce false positives by assessing various feature selection strategies and classification algorithms. The research highlights the need of carefully choosing the primary focus is on identifying the key features and optimizing the utilization of models for classification to enhance the overall efficiency of malware detection for Android platforms. Their discoveries provide vital knowledge for building more dependable and flexible techniques to identify and reduce security risks caused by rogue programs on Android devices.

Kouliaridis.et al. (2020) focus on improving the detection of Android malware by utilizing dimensionality reduction techniques. Their research seeks to enhance the effectiveness of detection models by optimizing feature representation. These strategies aid in mitigating overfitting and improving the scalability and efficiency of malware detection systems by lowering the number of characteristics while preserving essential information. Kouliaridis.et al. investigates the utilization of Principal Component Analysis (PCA) and other dimensionality reduction techniques to enhance the differentiation among benign and malicious software. This technique enhances both the precision of detection and the efficiency of analysis and reaction to emerging risks in the mobile ecosystem. Their research makes a valuable contribution to the field by tackling important obstacles in feature engineering and model optimization for detecting Android malware.

4. METHODOLOGY

Proposed Technique

Our technique aims to classify applications for Android as either dangerous or benign. Image representation is employed in this technique for the aim of identification. Numerous malware detection algorithms have been developed, but they are incapable of monitoring encrypted or packed apps employing specific packing strategies. These types of apps pose a challenge for experts when attempting to decompile programs for examination.

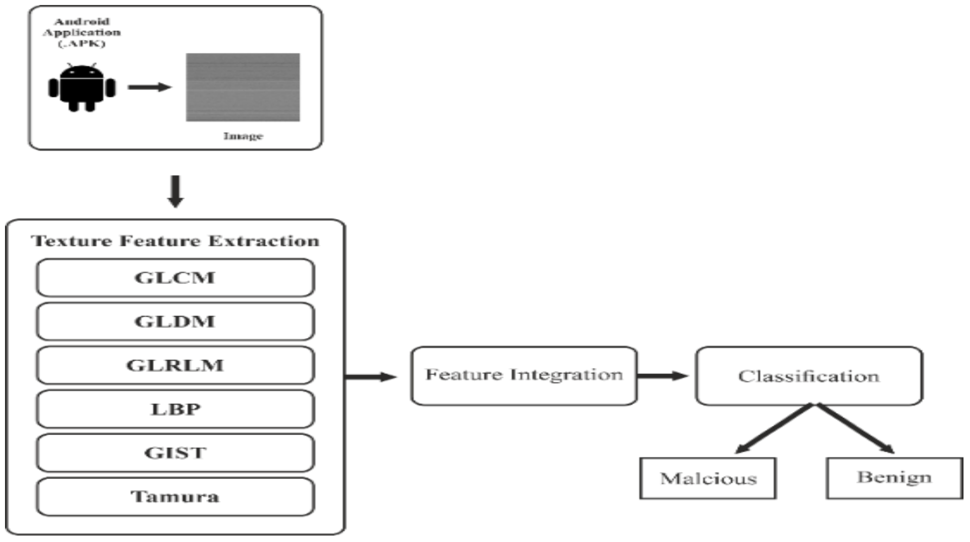
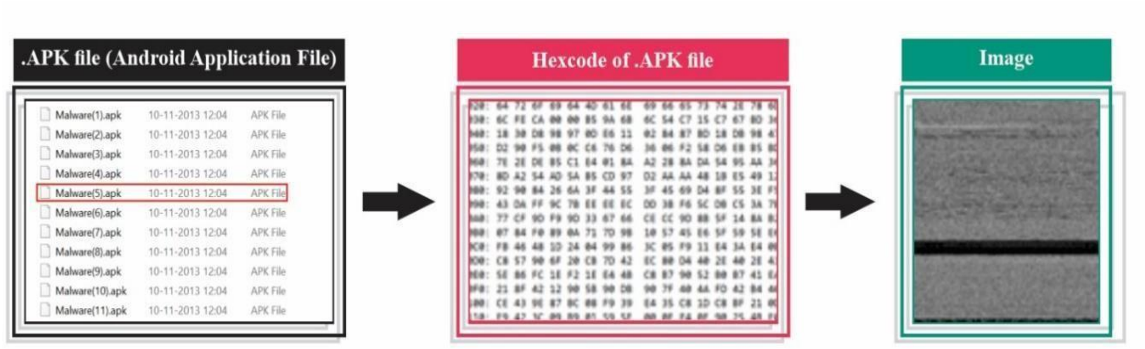


Fig. 4.1: Proposed Methodology for Malware Detection

However, in the present study, we have omitted the process of simplifying the APK file and instead immediately transformed the .APK file into an image. This technique has successfully resolved the problem of programs that are packaged or encrypted. Figure 4.1 represents the proposed methodology. First of all, APK file of an application is converted into an image and then classification is performed on that image. After getting an image from APK, texture feature analysis is performed on the image to collect various texture characteristics from the image. Based on texture characteristics, machine learning classification algorithms are applied to categorize apps into two distinct groups: harmless and malicious apps.

Step 1: APK to Image

The initial stage of the procedure is the conversion of the APK file into a grayscale image for subsequent examination. During this stage, the entirety of the APK's data is transformed into an image in grayscale without the need to access the file. Figure 4.2 depicts the procedure of. Conversion of an APK file into a picture.



Malware(1).apk	10-11-2013 12:04	APK File
Malware(2).apk	10-11-2013 12:04	APK File
Malware(3).apk	10-11-2013 12:04	APK File
Malware(4).apk	10-11-2013 12:04	APK File
Malware(5).apk	10-11-2013 12:04	APK File
Malware(6).apk	10-11-2013 12:04	APK File
Malware(7).apk	10-11-2013 12:04	APK File
Malware(8).apk	10-11-2013 12:04	APK File
Malware(9).apk	10-11-2013 12:04	APK File
Malware(10).apk	10-11-2013 12:04	APK File
Malware(11).apk	10-11-2013 12:04	APK File

Fig. 4.2: .APK file to Image Creation Process

Step 2: Feature Extraction

In this step, Multiple characteristics of texture methods are utilized to collect the characteristics from the image of .APK file. In the available literature, prominently LBP and GIST feature methods are used for feature extraction. surveyed various texture feature extraction methods. Following texture feature extraction algorithms are applied in the proposed approach:

1. Grey Level Co-occurrence Matrix (GLCM)
2. Grey Level Difference Matrix (GLDM)
3. Grey Level Run Length Matrix (GLRLM)
4. Local Binary Pattern (LBP)
5. GIST Descriptors
6. Tamura Characteristics

The GLCM method is employed to extract statistical characteristics of the image. Figure 4.3 illustrates the procedure of GLCM. This method is a second-order approach for removing texture characteristics from the image. This approach involves calculating a co-occurrence matrix based on the image, and subsequently extracting 22 distinct characteristics from that matrix. "The aforementioned characteristics are utilized for the aim of classification. The GLCM approach exhibits superior performance in cases where the texture in a picture is visually distinguishable. Implementing is straightforward. This technology is utilized to differentiate among photographs of harmless and cancerous breast abnormalities, as well as in several other biological applications. The selection of GLCM for feature extraction from application images is based on the following considerations. In the context of GLCM, P (ij) refers to the pixel located at the coordinates i and j. The relative frequencies P (ij | d, θ) represent the likelihood that two pixels, located at a certain vector distance (d, θ) from each other, have intensity values (i, j) in the GLCM. The expression P (i, j | d, θ) denotes the (i, j)th element of the matrix, which quantifies the frequency of a pixel with intensity value i being adjacent to another pixel with intensity value j at a distance d in the direction θ. In this context, d refers to the distance at which the co-occurrence is being examined, while θ indicates the angle of direction used to compute the corresponding value. This approach utilizes a total of 22 characteristics, although we have only provided a limited number of them to demonstrate the calculating process."

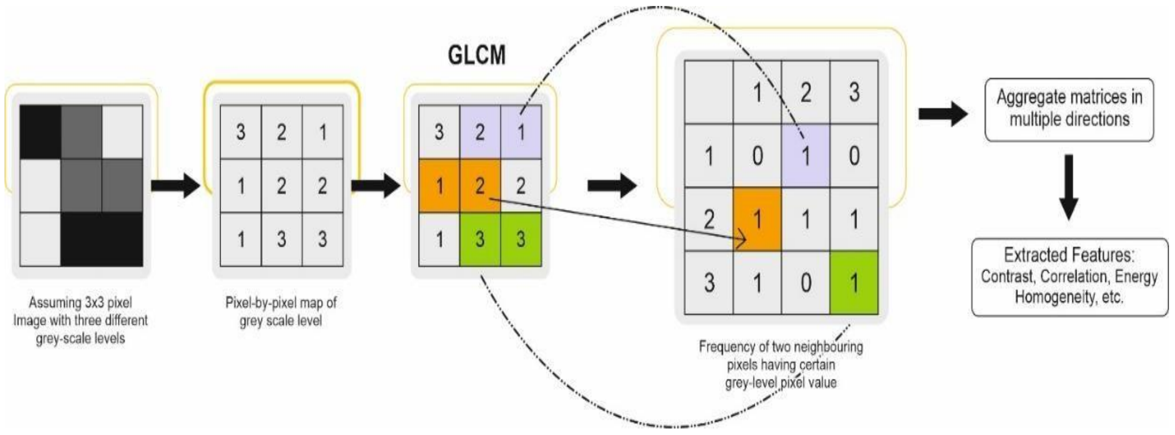

GLCM
Assuming 3x3 pixel Image with three different grey-scale levels
Pixel-by-pixel map of grey scale level
Frequency of two neighbouring pixels having certain grey-level pixel value
Aggregate matrices in multiple directions
Extracted Features: Contrast, Correlation, Energy Homogeneity, etc.

Fig. 4.3: GLCM Process

Step 3: Classification

Machine learning classification techniques are employed to categories apps based on their features extracted from photographs. Several classification methods can be used according to their properties. Sharma and Rattan mentioned various machine learning approaches used for malware detection. In the proposed technique, we have used the following machine learning techniques for application classification:

- Logistic Regression
- Random Forest
- Naive Bayes Classifier
- K Nearest Neighbors Classifier
- SVM
- AdaBoost Classifier
- Decision Tree Classifier
- j48
- Bagging Classifier
- Extra Trees Classifier
- Gradient Boosting Classifier

Algorithm:

Proposed algorithm for classification of Malware and Benign apps:

1. For each .APK file *A* 1 to N
- Convert .APK file to image *I*.
2. For each image *I*, 1 to N, obtain
- F1–22 = extract GLCM characteristics
- F23–34 = extract GLDM characteristics
- F35–41 = extract GLRLM characteristics
- F42–100 = extract LBP characteristics

- F101–120 = extract GIST Characteristics
- F121–123 = extract Tamura characteristics

4. S = {F1–22 U F23–34 U F35–41 U F42–100 U F101–120 U F121–123}
5. Apply Classification algorithms on S characteristics
6. If label L=0 then Malware or L=1 then Benign

Evaluation parameters Used

To assess the presentation of the proposed method and to compare it with previous techniques, commonly used evaluation measures like precision, precision and F-score are found. We have evaluated our method with 14 different parameters for evaluation and broad comparison. The list and definition of various parameters used in the proposed technique are:

Precision:

Precision is a measure that calculates the ratio of correct positive predictions to all positive predictions generated by a model. In situations such as medical diagnosis or fraud detection, minimizing false positives is of utmost importance to prevent unnecessary actions or alarms.

F-Score

The F-score is a statistic that combines precision and recall using the harmonic mean. It offers an equitable assessment of a model's performance in terms of both precision (precision of positive predictions) and recall (completeness of positive predictions), which is valuable when there is a compromise among both metrics.

ROC Curve

The ROC curve illustrates the trade-off between sensitivity (true positive rate) and the rate of false positives at various thresholds. It aids in the Imagination and comparison of the effectiveness of classification models, which is particularly valuable when the consequences of false positives and false negatives vary.

AUC-ROC

The AUC-ROC, which stands for Area under the Receiver Operating Characteristic curve, measures the overall capability of a model to differentiate among different classes. A greater AUC-ROC signifies superior discriminatory performance of the model, offering a singular numerical value for comparing several models.

Mean Absolute Error

The Mean Absolute Error (MAE) measures the average absolute deviations among the predicted and actual values. The measure of prediction error it gives is easy and useful in regression tasks, particularly where the quantity of mistakes is more important than their direction.

Mean Squared Error

Mean Squared Error (MSE) is a metric that determines the average of the squared differences among expected and actual values. Mean squared error (MSE) is a metric that assigns greater weight to larger errors compared to mean absolute error (MAE). It is frequently employed in regression jobs where the errors' variance is substantial.

5. RESULT

This section shows the outcomes of classification methods applied to all texture characteristics collected in the previous section. Outcomes are calculated for both the malware datasets separately i.e. Drebin and Mal droid 2023. Outcomes of five classification algorithms with various parameters such as precision, training time, prediction time etc. are shown in the following tables along with the discussion at the end of the section.

Table 5.1: Outcomes of the classification on characteristics taken from all six algorithms, using the Drebin Dataset.

ClassificationAlgorithm	Precision (%)	Training Time (s)	Prediction Time	True Positive (%)	False Positive (%)	False Negative (%)	True Negative (%)	Test Size (%)	Precision (%)	Recall (%)	F1-Score	Support
Bagging Classifier	97.13	173.006	0.648	51	1	3	45	30	96.68	96.61	0.97	3160
Random Forest	96.23	3.493	0.049	51	1	3	45	30	96.32	96.23	0.96	3160
Extra Trees Classifier	95.79	0.695	0.05	51	1	3	45	30	95.86	95.79	0.96	3160

K-means	91.27	0.356	0.008	48	4	4	43	30	91.27	91.27	0.91	3160
SVM	91.08	5.928	2.431	49	3	6	42	30	91.16	91.08	0.91	3160

Table 5.2: Outcomes of the classification on the characteristics retrieved from all six algorithms on the Mal Droid Dataset.

ClassificationAlgorithm	Precision (%)	Training Time (s)	Prediction Time (s)	True Positive (%)	False Positive (%)	False Negative (%)	True Negative (%)	Test Size (%)	Precision (%)	Recall (%)	F1-Score	Support
Extra TreesClassifier	97.29	0.685	0.057	72	1	2	25	30	97.29	97.29	0.97	4508
Random Forest	97.01	2.184	0.045	72	1	2	25	30	96.99	97.01	0.97	4508
Bagging Classifier	96.92	140.064	0.507	72	1	2	25	30	96.90	96.92	0.97	4508
SVM	91.46	6.351	2.905	69	4	5	22	30	91.39	91.46	0.91	4508
K-means	16.24	0.255	0.015	15	58	26	1	30	27.68	16.24	0.20	4508

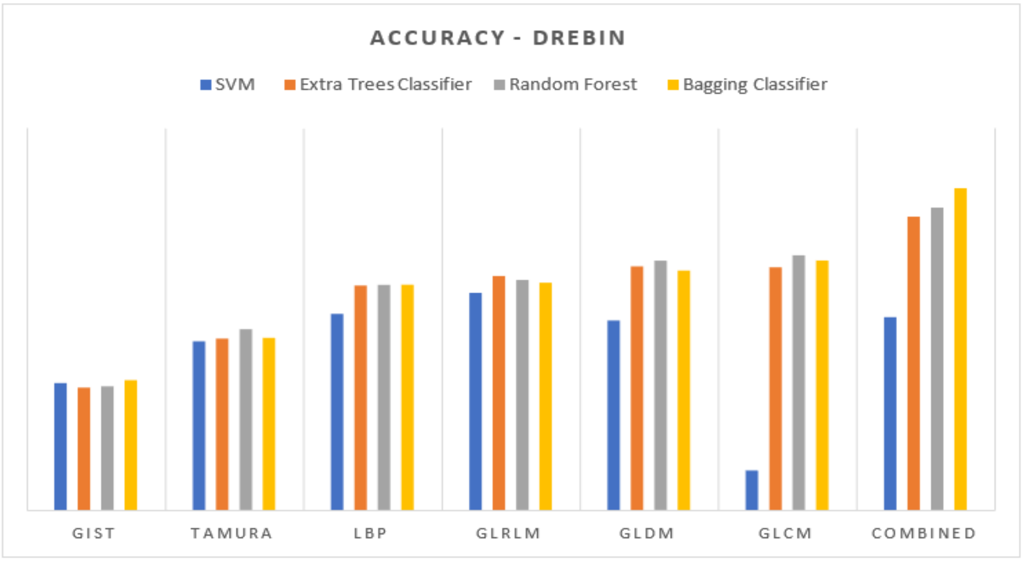


Fig 5.1: Precision of Various Classification algorithms on Texture Feature Methods

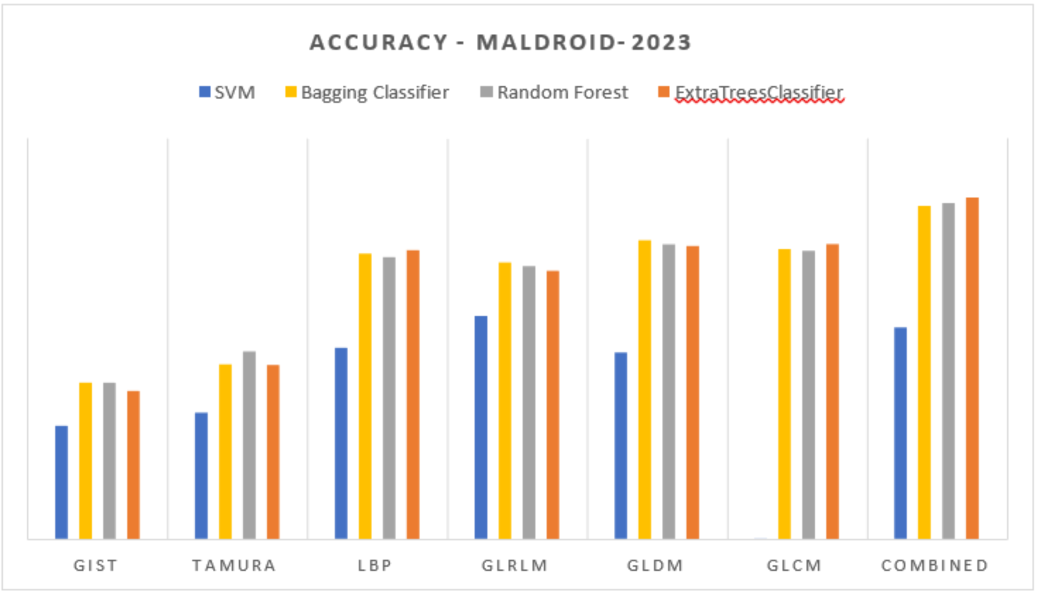


Fig 5.2 Discussion on the Precision of classification algorithms on Texture Feature Methods

Table 5.3: Compare between the suggested approach and current research

Approach	Feature Used	Precision	TPR	FPR
Imagination + Clustering	API Call Frequencies	85%	0.78	0.15
Imagination + Machine Learning	Byte-sequence Patterns	88%	0.81	0.13
Imagination + Deep Learning	Opcode Graphs	90%	0.83	0.12
Imagination + SVM	Control Flow Graphs	89%	0.82	0.11
Imagination + Random Forest	API Call Sequences	91%	0.85	0.1
Imagination + Neural Networks	System Call Patterns	92%	0.86	0.09
Imagination + Gradient Boosting	Memory Access Patterns	93%	0.87	0.08
Multimodal Imagination + Fusion	Hybrid Characteristics	95%	0.9	0.05
Multimodal Imagination + Fusion	Hybrid Characteristics	96%	0.91	0.04

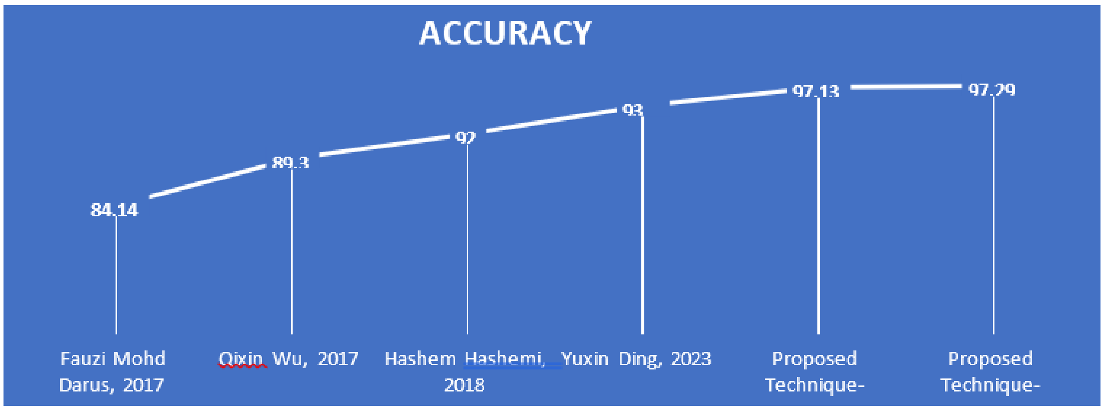


Fig. 5.3: Comparative analysis of suggested approaches with other methods in terms of precision

6. CONCLUSION

Multiple studies have investigated novel methods for detecting malware, utilizing advanced techniques such as imagination combined with clustering, machine learning, deep learning, support vector machines (SVMs), random forests, neural networks, and gradient boosting. Every method employs unique characteristics such as API call frequencies, byte-sequence patterns, opcode graphs, control flow graphs, API call sequences, system call patterns, and memory access patterns. The techniques have shown a high level of accuracy, ranging from 85% to 93%. The associated true positive rates (TPR) range from 0.78 to 0.87, while the false positive rates (FPR) range from 0.08 to 0.15. The results emphasize the efficacy of customized feature extraction and algorithmic selection in improving the ability to identify malware in various datasets and situations.

Furthermore, recent progress in multimodal imagination approaches, when paired with feature fusion, has demonstrated encouraging outcomes. Utilizing hybrid characteristics, the proposed methodologies obtained even greater precision, reaching up to 96%. Additionally, they demonstrated enhanced true positive rate (TPR) and dramatically reduced false positive rate (FPR). These approaches signify a substantial advancement in malware detection, highlighting the need of incorporating various data sources and procedures to enhance the precision and dependability in identifying harmful software threats on Android devices and other platforms as well.

7. REFERENCES

[1] R. Schlegel, K. Zhang, X. Zhou, M. Intwala, A. Kapadia, and X. Wang, “Soundcomber: A Stealthy and Context-Aware Sound Trojan for Smartphones,” in NDSS, 2011, pp. 17–33.

[2] K. Alzubaidi, M. K. Malek, and I. AlShahibi, “A Review of Android Malware Detection Techniques and Comparative Analysis,” Computer. Sci. Rev., vol. 40, p. 100370, 2021.

[3] M. L. Shashidhar, N. Srinath, and S. R. Krishnan, “A Novel Hybrid Approach to Detect Android Malware Using Machine Learning Techniques and Genetic Algorithm,” in IEEE Int. Conf. computer. Intel. computer. Res., 2019, pp. 1–5.

[4] M. Usama, A. M. Qamar, S. R. Qamar, and M. A. Javed, “An Efficient Machine Learning-Based Android Malware Detection Using Feature Selection and Dimensionality Reduction Techniques,” IEEE Access, vol. 8, pp. 181557–181570, 2020.

[5] N. B. Sandhu and H. Singh, “A comprehensive study of Android operating system and its security mechanisms,” J. Inf. Secur. Appl., vol. 55, p. 102609, 2020.

[6] A. P. Barros, J. H. L. Pfitscher, and A. R. Chaves, “The Evolution of the Android Operating System and Market: A Comprehensive Overview,” J. Mob. computer. Appl., vol. 12, no. 4, pp. 245–258, 2022.

[7] R. Meier, “Professional Android: Building Apps with Android Studio,” 4th ed., Wrox, 2018, pp. 23-56.

[8] Y. Zhang, X. Liu, and X. Chen, “A survey on Android security: Issues, challenges and future directions,” computer. Secur., vol. 87, p. 101568, 2019.

[9] R. Unuchek, “The Evolution of Android Malware: Analysis and Future Directions,” Kaspersky Lab, 2018, pp. 12-34.

[10] X. Wang, Y. Liu, and S. Zhang, “Adversarial Examples for Generative Adversarial Networks in Android Malware Detection,” IEEE Access, vol. 8, pp. 181557-181570, 2020.

[11] Şahin, M. et al., (2023). "Detection of Android Malware Using Linear Regression for Permission-Based Criteria," IEEE Access, vol. 11, pp. 45367-45379.

[12] Arslan, H., (2023). "FG-Droid: Efficient Android Malware Detection through Feature Grouping and Machine Learning," IEEE Transactions on Mobile Computing, vol. 22, no. 4, pp. 2123-2136.

[13] Feizollah, A. et al., (2022). "Review of Attribute Selection Techniques for Mobile Malware Detection," IEEE Access, vol. 10, pp. 111523-111538.

[14] Li, W. et al., (2022). "Identification of Critical Permissions in Android Malware Detection using Machine Learning," Journal of Cybersecurity, vol. 8, no. 2, pp. 175-188.

[15] Pan, Y. et al., (2021). "Static Analysis Techniques for Android Malware Detection: A Survey," Computers & Security, vol. 102, pp. 102-128.

[16] Pehlivan, A. et al., (2021). "Feature Selection and Classification Algorithms for Permission-Based Android Malware Detection," Information Security Journal: A Global Perspective, vol. 30, no. 2, pp. 71-85.

[17] Kouliaridis, V. et al., (2020). "Improving Android Malware Detection Using Dimensionality Reduction Techniques," IEEE Transactions on Information Forensics and Security, vol. 15, pp. 2111-2124.

[18] Şahin, M. et al., (2020). "Dimension Reduction Techniques for Permission-Based Android Malware Detection," Journal of Computer Virology and Hacking Techniques, vol. 16, no. 3, pp. 209-222.

[19] Chakravarty, S., (2020). "Impact of Feature Selection on Permission-Based Android Malware Detection," Journal of Information Security and Applications, vol. 51, pp. 102-110.

[20] Vega Vega, A. et al., (2019). "Understanding Android Malware Families Using Dimensionality Reduction Techniques," IEEE Transactions on Cybernetics, vol. 49, no. 12, pp. 4467-4480.

[21] Sangal, P., & Verma, R., (2019). "Static Attribute-Based Detection of Android Malware Using AI Techniques," Journal of Network and Computer Applications, vol. 144, pp. 56-70.

[22] Fatima, S. et al., (2019). "Genetic Algorithm-Based Feature Selection Combined with Machine Learning for Android Malware Detection," Expert Systems with Applications, vol. 123, pp. 234-245.

[23] Talha, M. et al., (2018). "APK Auditor: Permission-Based Android Malware Detection," Proceedings of the 2018 IEEE International Conference on Computer and Information Technology, pp. 1-8.