Sule Lamido University Journal of Science & Technology Vol. 6 No. 1&2 [March, 2023], pp. 235-246
https://doi.org/10.56471/slujst.v6i.371

Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

A Comparative Analysis of Android Malware Detection with and without Feature Selection Techniques using Machine Learning

Mohammed Ibrahim[1]*, Abdullahi Abdullahi[2], Muhammad Aminu Ahmad[3] and Rabi Mustapha[4]
Department of Computer Science, Kaduna State University, Kaduna, Nigeria
mohammed.ibrahim@kasu.edu.ng[1*], abdul4light@yahoo.com[2], muhdaminu@kasu.edu.ng[3], rabichubu@kasu.edu.ng[4]

*Corresponding Author

Abstract

Android is an open-source operating system mainly built for smart devices to make them easy to use and user-friendly. Thus, it has immensely engulfed other operating systems in mobile devices, which have become not only a major stakeholder in the market but have also become attractive targets for cyber criminals to lure many Androids *malware with the intention of stealing or destroying the user's information without the user knowing. Many traditional signature-based anti-malware efforts have been made to combat malicious apps, but these efforts have been insufficient due to the lack of ability to detect unknown malware. This insufficient effort by traditional signature-based has led to the intervention of researchers to embark upon combating unknown malware using machine learning techniques. This study looks into many existing research papers on malware detection using machine learning in order to determine the significance of feature selection techniques. The comparative analysis examines the importance of feature selection and unselected feature techniques.*

Keywords: *Android, Detection, Features, Machine Learning, Malware*

1. Introduction

Since its debut in 2008, Android has overtaken all other mobile operating systems in terms of popularity. The majority of the malicious code created by malware developers is used for nefarious purposes (Robiah et al., 2009). Its behavior and signature have rapidly evolved, making it challenging to halt. Malware including viruses, trojan horses, worms, and botnets were formerly associated with desktop environments and were infrequently detected on mobile devices. However, as mobile device technology advances to a high-end level and is able to support complicated operating systems, it has become the next target for malware.

Android malware is on the rise as a result of the market's strong adoption of the Android operating system. In the era of greater mobile and smart connection, malware is an ever-evolving threat to people, organizations, and governments. The mobile malware represents a security threat to mobile devices, there are different types of mobile malware compromising the security platform (Rashid and Al-Oqaily, 2015). The first line of defense for mobile users is frequently anti-malware firms. The amount and variety of Android malware are increasing due to financial gains, making detection more challenging. Static approaches are inadequate to identify new dangerous codes since their signatures are continually changing. Due to their ability in obfuscation, machine learning algorithms have significantly improved malware detection during the previous two decades. Training and prediction are the two steps on which machine learning models are based. A training dataset with samples of both good-ware and malware apps is used in the training step. There are three stages to it. The extraction of a huge number of features from the many training dataset files makes up the first stage. Using the proper feature selection strategies, unwanted features are rejected while those without feature selection techniques accommodate all the extracted features in the second step. The use of one or more classification models that can differentiate between dangerous and benign app is the third phase. As a result, these models can now make precise predictions when dealing with fresh executable files in the prediction step. In this paper, various existing papers were analyzed to evaluate the role of feature selection techniques in malware detection using machine learning and comparative

Sule Lamido University Journal of Science & Technology Vol. 6 No. 1&2 [March, 2023], pp. 235-246
https://doi.org/10.56471/slujst.v6i.371
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

analysis. The paper is organized as follows: section 2 overview the machine learning techniques in malware detection, section 3 related work, 4 comparison and performance evaluation of existing techniques, section 5 provide summary of the paper and 6 conclude the review of the existing work.

2. Machine Learning

To safeguard Android from malware, machine learning processes as demonstrated in Figure 1 with classifiers are used in malware identification. The traits of each class are automatically learned from input data or samples to determine whether the application is harmful or benign. The machine learning method enables the model to be trained for effective malware detection. Three approaches are used in machine learning to learn a system: supervised, unsupervised, semi-supervised and reinforcement learning.

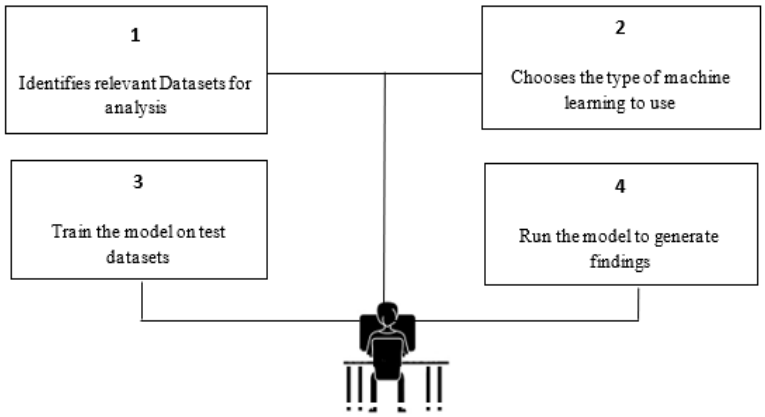

Figure 1. Machine learning process in malware identification.

In any case, the data must pass through detection, gathering, combining, structuring and organization so it can be used in identifying malware. These features can either be raw information contained in the files that will be examined with feature selection techniques or the raw information is processed as shown in Figure 2. Both the benign and malicious files are considered for the training of the chosen machine learning model.

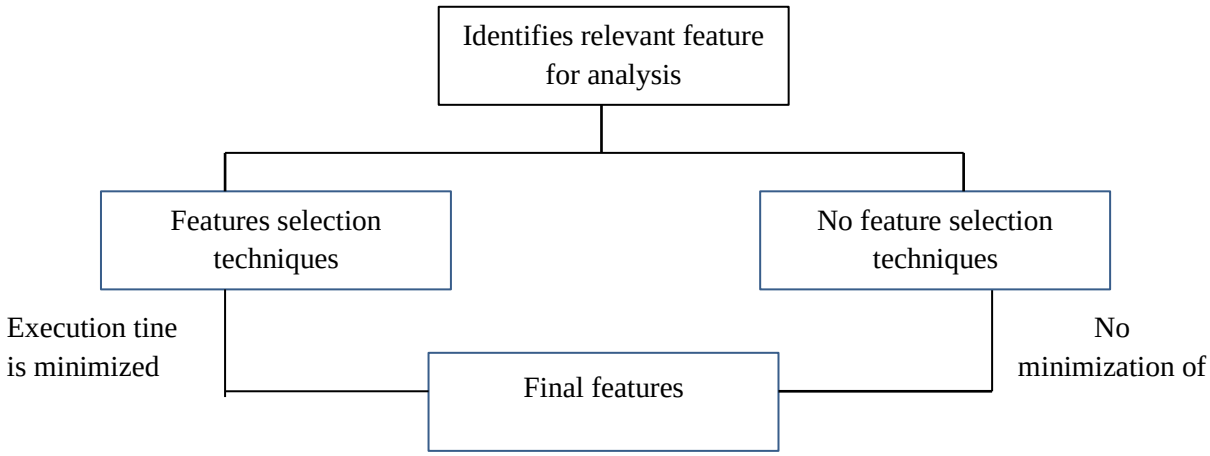

Figure 2. Data preparation for malware detection

3. Related Works

A survey of the literature on Android malware detection using machine learning approaches is presented in this section. There are several indicators to measure the performance of a given classifier, but in this paper, we were interested

Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

in the accuracy rate of each of the research classification studies. Accuracy is defined as the number of malicious files classified as malicious plus the number of benign files classified as benign, divided by the total number of files. However, a brief summary of the techniques and typical features utilized for machine learning-based Android malware detection was elucidated as follow.

Harry et al. (2015) Proposed Android Anomaly Detection System Using Machine Learning Classification. This research work analyzing anomalies on power consumption, battery temperature and network traffic data as features for machine learning classification algorithm (Support Vector Machine (SVM), Random Forest (RF) and Logistic Model Tree (LMT)). Feature combinations were used in seven folds to identify malware, with the highest accuracy achieved by combining three features (power consumption, batter temperature, and network data traffic) for classifiers.

Hsin-Yu and Sheng-De (2015) Proposed Machine learning based hybrid behavior models for Android malware analysis. This study employed a method that emphasizes static analysis, obtaining data from the contagiodump dataset, performing a frequency analysis, and obtaining the features of the data in a comprehensive view. The number of Android API calls that are made is calculated and sorted according to their usage by benign and malicious applications, respectively, by the frequency analysis. Using the statistics, two separate feature sets are produced. The first is a list of APIs that are more frequently utilized by legitimate applications than by malicious ones. The second is a list of APIs that are more frequently utilized by malicious than by good applications. They took into account the preferred APIs by good applications as well as the often-used APIs by Android malware when analyzing the two distinct behavior elements, building decision models for each of the two feature sets using the SVM method, a popular machine learning technique. The hybrid model classifier is created by combining the two decision models using fusion logic. SVM scores serve as the foundation for the fusion logics. Four different feature combinations were utilized to identify malware, with the hybrid model, which fuses the normal and harmful behavior models using fusion logic, achieving the greatest accuracy (96.69%) in the process.

Westyarian et al., (2015) proposed Malware Detection on Android Smartphones using API Class and Machine Learning. In order to distinguish between malicious and benign applications in an android environment, this research effort uses API calls as characteristics to the three classifiers (SVM,J48, and RF). Also, using 16 API classes and 51 packages, there are 412 sample Android applications, 205 of which are benign, and 207 of which are malicious

Songyang et al (2016) proposed Effective Detection of Android Malware Based on the Usage of Data Flow APIs and Machine Learning. This research work uses a machine learning to detect Android Malware with the aid of dataflow Application Program Interfaces (APIs) as classification features. To improve the k-nearest neighbor classification model and collect API-level dataflow information, a thorough analysis was conducted. Also, 1,160 benign and 1,050 malicious samples totaling 2210 APK files were utilized to test the suggested system. The results show that the system has an accuracy rate of up to 97.66% for detecting unidentified Android malware. According to our static dataflow analysis experiment, the new API subset enables the discovery of over 85% of sensitive data transfer channels while cutting down on analysis time by roughly 40%.

Long and Haiyang (2017) present an Android Malware Detection System Based on Machine Learning, in their system, features extraction from the APK files were based on the static and dynamic analysis. These features were reduced using Principal Component Analysis (PCA) and relief. The client and server sides of the model are separated into two main parts. Client-side software largely serves as the users' user interface and sends out alerts when predictions come true. Due to resource constraints, a straightforward check was nevertheless carried out at the client side by extracting the MD5 value whenever a new application is installed and comparing its value with the malicious MD5 value kept at the server side. If matches are found, as in this case, the model triggers a malicious alert to the client with an option to delete the files. The APK file is sent to the server for feature extraction using static and dynamic analysis using a marching learning classifier (SVM) and evaluation of the unknown Android application by classifying it as malware or benign. The model was trained on these three features independently PCA, RELIEF and PCA- RELIEF. The PCA-RELIEF has highest accuracy.

Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

Mohsen et al., (2018) proposed Application of Machine Learning Algorithms for Android Malware Detection. In order to improve malware detection, this research involves static app analysis, which requires examining the presence and frequency of keywords in the Android application manifest file and creating static feature sets from a dataset of 400 apps. To ascertain which technique is more suited for Android malware detection, the classification performance of the ML algorithms' accuracy and true positive rate are assessed and analyzed. Using the two most promising ML classifier methods identified from prior research, the Support Vector Machine (SVM) and K-Nearest Neighbor (KNN) algorithms, In an Android app's manifest file, they look at both the user permissions and the intent filters requested. The main objective of this study is to determine whether using the two machine learning algorithms and looking for keywords in the permissions requested in an app's manifest file and system call logs may improve our ability to identify malicious apps. The experimental results for a dataset of real malware and benign apps show average accuracy rates of 79.08 percent and 80.50 percent, respectively, with an average true positive rate of over 67.00 percent and 80.00 percent.

Oktay and Ibrahim (2019) Present Permission-based Android Malware Detection System Using Feature Selection with Genetic Algorithm (GA). This research proposes a feature selection using genetic algorithm method for detecting Android malware (GA). However, to detecting and analyzing Android malware, three alternative classifier algorithms (Decision Tree (DT), Naïve Bayes (NB) and Support Vector Machine (SVM)) with various feature subsets were constructed and compared using GA. With the 16 specified permissions and a dataset of 1740 samples containing 1119 malwares and 621 benign samples, this classifier SVM produces best accuracy result of 98.45% and, with 152 permissions, the accuracy drops to 96.92% for both features produced by GA.

Zhuo et al. (2019) brought a Combination Method for Android Malware Detection Based on Control Flow Graphs and Machine Learning Algorithms. This paper presents a machine learning-based combination method for detecting Android malware. To obtain API information, decompile the Android application and construct the control flow graph (CFG) from the source code. Then extract Application Program Interface (API) calls from the CFG and build three different types of API data sets: Boolean data sets, frequency data sets and chronological data sets. Three detection models for Android malware detection based on API calls, API frequency, and API sequence are built using these three data sets. Finally, a machine learning ensemble meta-algorithm for the experiments were carried out on 10010 benign and 10683 malicious applications. Our detection model achieves 98.98 percent detection precision, as well as high accuracy and stability, according to the results.

Hyoil et al., (2020) proposed Enhanced Android Malware Detection: An SVM-based Machine Learning Approach. This research paper uses a machine learning classifier called Support Vector Machine (SVM) along with API calls as features extracted from the Android application files, called APK files. The static analysis was used in extracting 133,227 features from the 58,602 Android applications. However, the Drebin dataset for malware detection was used for the experiment, with 28,489 benign apps and 30,113 malicious apps. Finally, noises were removed from 133,227 to 41,545 API features, and the experimental result shows 99.75% overall accuracy.

Todd McDonald et al., (2021) proposed a Machine Learning-Based Android Malware Detection Using Manifest Permissions. The effectiveness of four different machine learning algorithms in conjunction with features selected from Android manifest file permissions to classify applications as malicious or benign is investigated in this study. Case study results on a test set of 5,243 samples yield accuracy, recall, and precision rates of more than 80%. With 82.5 percent precision and 81.5 percent accuracy, Random Forest outperformed the other algorithms (Support Vector Machine, Gaussian Nave Bayes, and K-Means).

Abijah et al. (2021) Presented Android Malware Detection and Classification using LOFO Feature Selection and Tree-based Models. This paper describes an Android malware detection system that uses tree-based learning models to classify malware applications based on the top selected features. The experimental evaluation is carried out on the DREBIN data set, which contains 15,036 samples, 5560 of which are malware applications and 9476 of which are benign applications, and each sample contains 215 features obtained from static code analysis to demonstrate the efficacy of the proposed method. The XGBoost classifier outperforms other tree-based models in prediction accuracy of 95.59% with a limited set of features.

Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

Durmus et al. (2021) Proposed a novel Android malware detection system: adaption of filter-based feature selection methods. This method uses static Android malware detection based on machine learning. Permissions extracted from application files are used as features in the developed system. Dimension reduction is carried out with eight different feature selection methods to enhance the running time and efficiency of machine learning algorithms. While four of these, Information Gain (IG), Odds ratio, Chi-square and Inverse document frequency, are used in Android malware detection systems, the remaining document frequencies (threshold, Relevance frequency, feature selection etc.) are adapted from text classification studies. The adapted methods are compared in terms of both extracted features and classification results. When the results are examined, it is shown that the adapted methods improve the efficiency of the classification algorithms and can be used in this field.

Juliza et al (2022) proposed a static analysis approach for Android permission-based malware detection systems. This study used the Androzoo dataset for benign applications (5000 benign) and the Drebin dataset for malware applications (5000 malware), and these were combined into a database for training and testing sets. After pre-processing, the datasets were filtered as unsupervised and randomized. Then, for optimization, particle swarm optimization (PSO), information gain, and evolutionary computation were used to provide the best permission features using static analysis. The feature selection techniques were evaluated with five machine learning classifiers (Random Forest, MLP, kNN, J48, and Adaboost) to detect malware and The performance of each classifier is measured by five metrics such as TPR, FPR, precision, recall, f-measure, and accuracy.

Ahmed et al., (2022) developed an Android Malware Detection Leveraging Machine Learning. This research work studied the analysis of static, dynamic, and hybrid applications to categorize the malicious and benign applications in the android environment. They propose an Android malware detection model based on static, dynamic, and hybrid analysis along with the machine learning classifiers and use the feature rank approach, as this method leverages certain critical elements in feature arrangement due to its ability to choose the appropriate features needed to build the malware detection models. Following that, they apply various machine learning algorithms (such as gradient boosting, XGBoost, Decision Tree and Random Forest) and compare their performance to determine the most accurate algorithm. Finally, the results showed that the accuracy achieved from static, dynamic, and hybrid analyses was above 94%, so using static analyses alone should be efficient and lower cost for classification in these cases.

Beenish et al. (2022) Malware Detection: A Framework for Reverse Engineered Android Applications through Machine Learning Algorithms. This study employs reverse-engineered Android application features as well as machine learning algorithms to identify vulnerabilities in smartphone applications. This work's success is based on two factors. To begin, a model that combines more innovative static feature sets with the largest current datasets of malware samples than traditional methods. Second, ensemble learning was combined with machine learning algorithms such as AdaBoost, SVM, and others to improve the performance of our model. The experimental results and findings show that detecting malware from Android applications is 96.24% accurate, with a 0.3 false positive rate (FPR).

Eslam et al.(2022) used a Machine Learning to Identify Android Malware that rely on API calling sequences and Permissions. This research work presents an Android malware detection technique based on APIs and permissions. The goal is to assess and investigate the integration of machine learning classifiers with prominent Android features such as APIs and permissions. They investigated several methods for identifying Android malware based on the feature used. They examined all machine learning classifiers on Android malware detection and discovered varying performance. Furthermore, two processes are involved in this research work: preprocessing and processing. Preprocessing comprises the gathering of features from Android apps by obtaining their most frequently used API calls and permissions. The input consists of 3,800 unique Android applications that were gathered from the Malgenome data set. The Malgenome dataset contains permissions and API calls for benign and malicious apps; likewise, the Maldroid dataset contains innocuous apps, adware, banking malware, and mobile riskware. The data set was split into three subgroups throughout the processing stage: training, validation, and testing. The training set was used to train and test multiple models; the models were trained and evaluated on the data set using a variety of methods, including K-nearest neighbor, Naive Bayes, Support Vector Machine, and Decision Tree.

Sule Lamido University Journal of Science & Technology Vol. 6 No. 1&2 [March, 2023], pp. 235-246
https://doi.org/10.56471/slujst.v6i.371
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

Ahmed et al. (2022) proposed an Android Malware Detection Approach Based on Static Feature Analysis Using Machine Learning Algorithms. This research work proposed a static base classification method for Android malware detection on permission and API call features from the (CIC InvesAndMal2019) dataset and applied feature importance selection using the Recursive Feature Elimination (RFE) on the Logistic Regression model. The model is based on three well-known machine learning algorithms: support vector machine (SVM), K-nearest neighbors (KNN) and Naive Bayes (NB), which shows that the (SVM) classifier had the highest rates among other classifiers compared. It delivered an average 94% accuracy rate using permission features and an 83% accuracy rate using the API call features in pursuit of achieving high malware detection rates.

Mohamed (2022) Present an Android Malware Detection using Machine Learning Techniques. This research work used permission-based and signature-based methods on two different datasets to distinguish between benign and malware applications. Different classification models (KNN, Logistic Regression, and Random Forest) were built to distinguish between malware and benign applications. The first data source contains information on permissions granted to the applications, while the second data source contains information on API call signatures of the applications. Twenty of the best features were selected for the classifiers using three feature selection techniques such as frequency count, correlations, and chi-square. The models are trained, and their performance metrics, such as precision and recall, are analyzed and compared. This comparison is first made for different classification models within an approach. Then, the best results for each approach are compared to understand which of the two systems is better for detecting malware. The best model for the permissions-based approach is Random Forest, and the best model for the signatures-based approach is KNN Classifier.

Fahad et al.(2022) proposed permissions-Based Detection of Android Malware Using Machine Learning. The proposed Permission-based Malicious Apps detection system (PerDRaML) extracts permission from Android application packages. However, rather than analyzing all requested permissions, PerDRaML focuses on a subset of permissions that are effective in distinguishing and improving malware detection rates, and Random Forest-based feature importance was used to enlist the important permissions. The proposed scheme used the Support Vector Machine, Rotation Forest, Nave Bayes, and Random Forest classifiers for classification and produced an average accuracy of 89.7% with Support Vector Machine, 89.96% with Random Forest, 86.25% with Rotation Forest, and 89.52% with Naïve Bayes models. In this regard, the summary of the previous works is tabulated in Table 1.

4. Comparison and performance Evaluation of classification techniques

This section shows the analysis of existing approaches in Tables 1 in the bar chart based on machine learning classifiers. These five machine learning classifiers (SVM, RF, and KNN) are represented in the bar chart. The x-axis and y-axis of the bar chart represent the author's work and the percentage of accuracy used in specific approaches, respectively. The bar chart was drawn on the basis of the accuracy of each algorithm used for Android malware detection corresponding to the author.

Table 1. Performance evaluation of existing approaches with and without feature selection techniques
(x= not calculated and N=not applicable)

S/No	Author	Analysis	Data Set	Features	Feature selection techniques	Algorithm	Accuracy (%)
1	Harry et al., (2015)	Dynamic	200 malwares from Android Malware Gnome Project. 200 benigns from google play store	Internet traffic, battery usage battery temperature	-	RF SVM LMT	86.00 84.00 85.00
2	Hsin-Yu and Sheng-De, (2015)	Static	6005 benigns and 3368 malwares	API calls	-	SVM	96.69

Sule Lamido University Journal of Science & Technology Vol. 6 No. 1&2 [March, 2023], pp. 235-246
https://doi.org/10.56471/slujst.v6i.371
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

3	Westyarian etal.,(2015)	Static	205 benigns and 207malwares	API calls	-	SVM RF	x
4	Songyang et al., (2016)	Static	1,160 benigns and 1,050 malicious	API calls	-	KNN	97.66
5	Long and Haiyang (2017)	Static	1000 benign- google play store, 1000 malware- Drebin Project and Genome Project	Permission, intent, CPU usage, battery usage, number of running processes, number of short message and API calls	PCA-RELIEF	SVM	95.2
6	Nikola et al., (2017)	Static and Dynamic	M0Droid dataset 200 malicious and 200 benign	Permission and source code	-	SVM	95.1
7	Mohsen et al., (2018)		M0Droid- 200 benigns and 200 malwares	Permission and system call logs	-	SVM KNN	79.08 67.00
8	Oktay and Ibrahim (2019)	Static and Dynamic	Genome Project (AMGP) with 1119 malwares and 621 benign from google play store	Permission	Genetic algorithm	NB DT SVM	95.69 97.24 98.45
9	Zhuo et al., (2019)	Static	AndroZoo with 10010 benign and 10683 malwares from VirusShare, Google Play and third-party security companies	API calls	Control flow graph	API usage API frequency API sequence	x
10	Hyoil et al., (2020)	Static	28,489 benign from Play store, Amazon AppStore and APKPure. 30,113. malwares from AMD and Drebin with 133,227 attributes	API calls	noise filter	SVM	99.75
11	Arindaam et al., (2020)	Static	1100 malware from DREBIN 1100 benign from "CICInvesAndMal2019	API calls	Non-negative Matrix factorization	L. Reg. SVM RF KNN	91.86 93.35 93.77 93.15
12	Abijah et al. (2021)	Static	Drebin: 5560 malware apps, 9476 cleanware apps and each sample has 215 features	Permission and API calls	Leave One Features Out	DT RF XGBoost	93.86 95.57 95.59
13	Durmus et al., (2021)	Static	1000 malware apps randomly selected from the Android Malware Dataset and 1000 benign apps are downloaded from APKPure	Permission	linear regression	Multi-Layer Perceptron (MLP)	x

Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

14	Durmus et al., (2021)	Static	3000 malicious apps from (APKPure 2020) and 3000 benign apps from VirusShare dataset (2020).	Permission	filter-based	RF	x
15	Ahmed et al., (2022)	Static, dynamic and hybrid	104747 malware apps and 90876 benign apps from Palo Alto Networks	Permission, API call, intent, DNS, IP address, port address and action repeat	Random Forest Algorithm	GB XGBoost DT RF	99.51 99.55 99.25 96.03
16	Ahmed et al., (2022)	Static	426 malware apps and 5,065 benign apps from CIC InvesAndMal2019	API calls and Permission	Recursive Feature Elimination	SVM KNN NB	94.36 93.42 84.33
17	Beenish et al., (2022)	Static	1795 benign and 10,516 malwares from MalDroid 1500 benign and 3062 malwares from DefenseDroid 2421 benign and 5000 malwares from GD	API calls, permissions, intents, packages, receivers and services	-	AdaBoost DT SVM KNN NB RF	96.24 90.12 92.00 89.45 88.65 89.00
18	Eslam et al., (2022)	Static	Malgenome dataset-1260 malwares and 2539 benigns	API and permission	-	KNN NB SVM DT	99.30 97.40 100.00 100.00
19	Fahad et al., (2022)	Static	5,000 malware apps from VirusShare 5,000 benign apps from google play store	Permission	Random Forest-based feature importance	SVM RF Rot. F. NB	89.7 89.96 86.25 89.52
20	Juliza et al., (2022)	Static	5,000 benign -Androzoo and 5,000 malware from Drebin	Permission	PSO, information gain, and evolutionary computation	RF KNN	91.59 91.56
21	Kumar et al., (2022)	Static	3799 Android apps from Google play store	Permission and API call signature	Genetic algorithm	SVM NN	93.00 98.02
22	Mohamed (2022)	Static	Mahindru, Arvind (2018), Mendeley Data, and Malgenome	permissions and signatures	Frequency Counts, correlations and Chi-Square Test	Permission: RF Signature - based: KNN	x
23	Todd McDonald et al., (2022)	Static	4597 benign from Google Play store 6000 malicious from AndroZoo	permission	-	RF SVM	81.53 79.87

Sule Lamido University Journal of Science & Technology Vol. 6 No. 1&2 [March, 2023], pp. 235-246
https://doi.org/10.56471/slujst.v6i.371
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

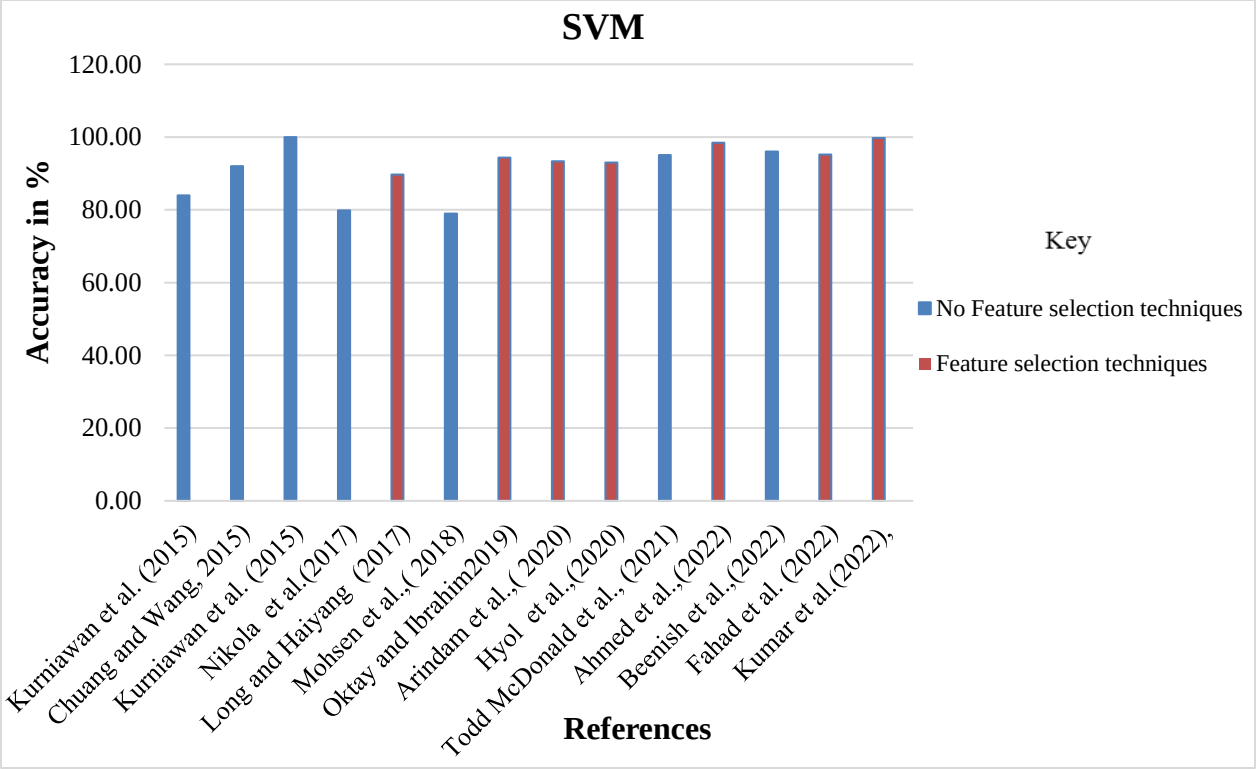

Figure 3. SVM analysis with and without feature selection techniques

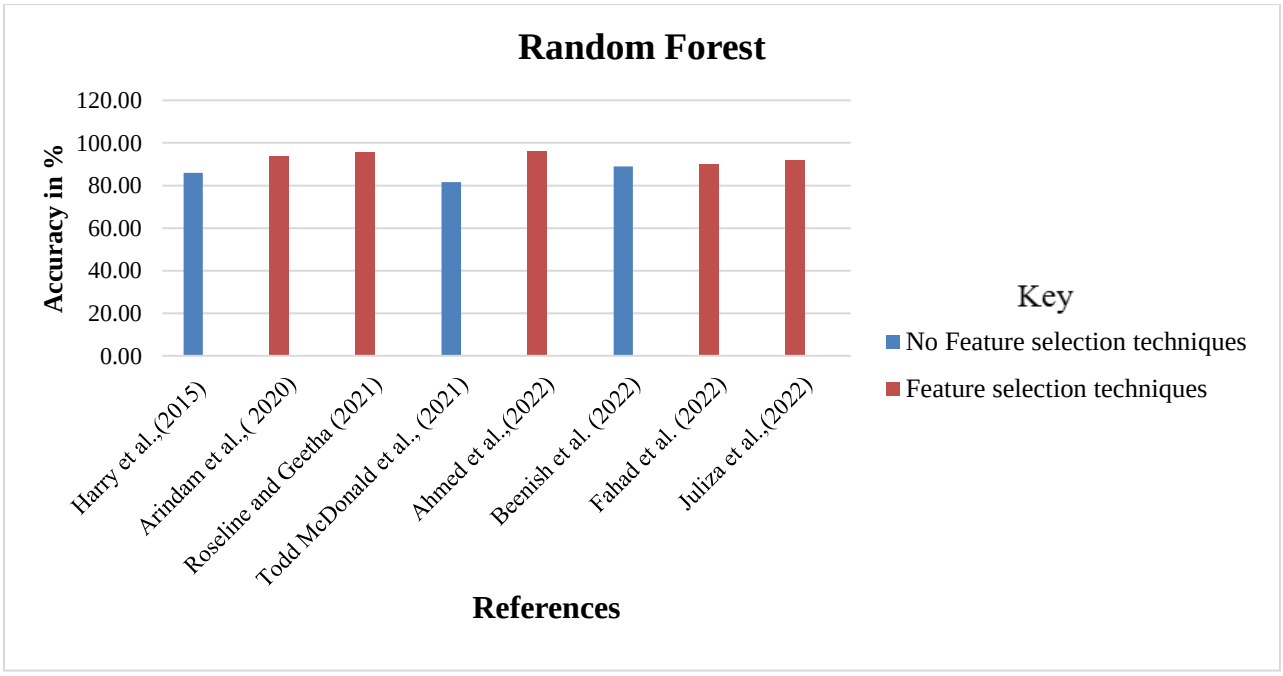

Figure 4. Random Forest analysis with and without feature selection techniques

Sule Lamido University Journal of Science & Technology Vol. 6 No. 1&2 [March, 2023], pp. 235-246
https://doi.org/10.56471/slujst.v6i.371
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

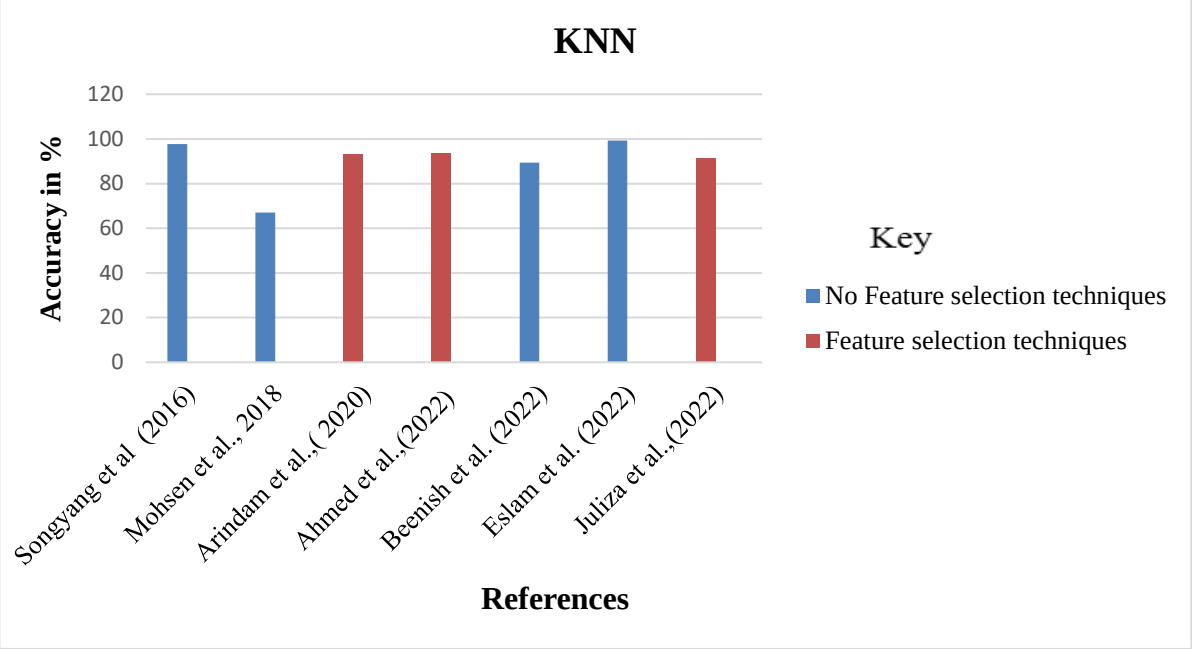

Figure 5. KNN analysis with and without feature selection techniques

As illustrated in each of the above charts, accuracy was used to demonstrate the effectiveness of using feature selection techniques in Android malware detection. Feature selection techniques in Android malware detection give the advantage of reducing the running time of the model and of higher accuracy compared to not using feature selection techniques. Figures 3, 4 and 5 are related works with and without feature selection techniques, and the results obtained in each work are charted for comparison. The blue color in the charts representing the related work with without feature selection techniques while those in orange color representing the work with feature selection techniques. Furthermore, Analyzing figure 3, most of the related work with feature selection techniques' SVM accuracies are higher than their counterpart without feature selection techniques, and the same for other charts that follow. The comparison of these charts proved that using a feature selection technique produces a higher accuracy than not using the technique. The high accuracy of SVM in related works of Hsin-Yu and Sheng-De (2015) and Fahad et al., (2022) is achieved through the use of lighter datasets in their content. This dataset doesn't contain most of the newest features of Android malware, and obfuscated malware can trick the model to avoid detection.

Figure 3 shows a comparison of each accuracy level, Fahad et al., (2022) has the highest accuracy, owing to the fact that it takes only one aspect of the Malgenome dataset, which is permission, making the dataset lighter in its content. The dataset with many features will produce high level accuracy with SVM and Random Forest and less running time by the model that uses feature selection techniques compared to without feature selection techniques in Android malware detection.

5. Summary

With the recent rise in malware and the dynamic changes in mobile phone technology, most especially Android mobile phone devices, there is a need to adopt dynamic approach like machine learning in handling the security aspect of Android phone devices. As illustrated in this paper, several machine learning techniques have centered their proposed techniques on detecting malware infiltration on Android phone devices. Nevertheless, the accuracy of detecting this malware remains a hot issue. Many works have focused on feature selection techniques to detect Android malware, while others detect the malware using non feature selection techniques. For the benefit of the research community, this paper critically reviews the existing works on Android malware detection and gives Android malware researchers an insight into using feature selection techniques on the areas that need further research, considering the degree of high accuracy it

Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

produces in Android malware infiltration and model performance enhancement. As such, these feature selection techniques remove redundant data from the dataset and lighten the workload on the machine learning algorithm.

6. Conclusion

Recently, the mobile community has faced a serious security threat from mobile malware. In order to handle security challenges, machine learning algorithms must be more accurate. These issues are typically concerned with the impact of features selection and the effective operation of the chosen classifiers. By comparing and contrasting the accuracy of machine learning classifiers in Android malware detection with many related works in this paper to demonstrate that it is possible to achieve a higher level of accuracy by reducing the number of features while maintaining high levels of efficacy. Using compare and contrast figures 3 and 4, and 5, it is clearly shown that using feature selection techniques in Android malware will produce high level detection accuracy. In the future, the work will focus on comparison with and without feature selection techniques by using the same dataset, and measure the accuracy, precision, recall, and f-measure of machine learning classifications.

References

Abijah S. R., and Geetha, S (2021). Android Malware Detection and Classification using LOFO Feature Selection and Tree-based Models. *Journal of Physics: Conference Series 1911(2021)*

Ahmed, S. S., Aya J., Tuqa B. Y., Eyad T., Mahmoud and Dheya M.(2022). An Android Malware Detection Leveraging Machine Learning. *Wireless Communications and Mobile Computing*, .

Ahmed S. S., Qussai Y., and Abdulrahman Y. An Android Malware Detection Approach Based on Static Feature Analysis Using Machine Learning Algorithms(2022).The 3rd International Workshop on Data-Driven Security (DDSW 2022), March 22 - 25, Porto, Portugal

Arindaam, R., Divjeet S. J., Gitanjali J., and Kapil S. (2020). Android Malware Detection based on Vulnerable Feature Aggregation. *Procedia Computer Science*, 173, 345-353.

Beenish U., Munam A. S., Carsten M., Muhammad K. A., and Sidra R(2022). Malware Detection: A Framework for Reverse Engineered Android Applications through Machine Learning Algorithms. *IEEE Access*, 10, 89031-89050.

Durmus O. S., Oguz E. l, Sedat A., and Erdal., (2021). A novel Android malware detection system: adaption of filter-based feature selection methods. *Journal of Ambient Intelligence and Humanized Computing* , 1-15

Eslam A., Seif E. M., Mostafa A., Amr E., Omar S., Haytham M., and Ammar M.,(2022). Using Machine Learning to Identify Android Malware Relying on API calling sequences and Permissions. *Journal of Computing and Communication* 1(1), 38-47,

Fahad A., Mehdi H., Rafia M., Qaiser R., Ainuddin W., A and Ki-Hyun J. (2022) .Detection of Android Malware Using Machine Learning. *Symmetry* 14, 718.

Harry K., Yusep R., and Budiman D., (2015).Android Anomaly Detection System Using Machine Learning Classification. The 5th International Conference on Electrical Engineering and Informatics August 10-11, 2015, Bali,Indonesia.

Hsin-Yu C,. and Sheng-De W., (2015). Machine learning based hybrid behavior models for Android malware analysis. IEEE International Conference on Software Quality, Reliability and Security

Hyoil H., SeungJin L., and Kyoungwon S.,(2020). Enhanced Android Malware Detection: An SVM-based Machine Learning Approach. IEEE International Conference on Big Data and Smart Computing (BigComp) pp. 75-81.

Juliza M. A., Mohd F. A., Suryanti A., Sharfah R. T., Nor S. N. I. and Ahmad F., (2022).A static analysis approach for Android permission-based malware detection systems. *PloS one*, 16(9), e0257968

Kumar, N. S., Vishnu V., Akhil., P. C. and S.V Spandan K., (2022). Analysis of Malware in Android Features Using Machine Learning. *Journal of engineering sciences.* 13(1).

Long W., and Haiyang Y., (2017). An Android malware detection system based on machine learning. AIP Conference Proceedings 1864, 020136

Mohsen K., Mohammad D., and Ali D., (2018).Application of Machine Learning Algorithms for Android Malware Detection. ACM ISBN 978-1-4503-6595-6/18/1

Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

Mohamed Salem Alhebsi (2022).Android Malware Detection using Machine Learning Techniques. Thesis. Rochester Institute of Technology. Accessed from https://scholarworks.rit.edu/theses/11178

Nikola M., Ali D., and Kim-Kwang R. C., (2017). Machine learning aided Android malware classification. *Computers & Electrical Engineering*, 61, 266-27.4

Oktay Y., and Ibrahim A. D, (2019). Permission-based Android Malware Detection System Using Feature Selection with Genetic Algorithm. *International Journal of Software Engineering and Knowledge Engineering*. 29(2) 245–262

Rashid, A. M., and Al-Oqaily, A. T. (2015). Detect and prevent the mobile malware. *International Journal of Scientific and Research Publications*,5(5), 487-89

Robiah, Y., Rahayu, S.S., Zaki, M.M., Shahrin, S., Faizal, M.A, & Marliza, R. (2009). A new generic taxonomy on hybrid malware detection technique. Int. J. Comput. Sci. Inform. Secur., 5: 56-61

Songyang W., Pan W., Xun L., and Yong Z., (2016). Propose Effective Detection of Android Malware Based on the Usage of Data Flow APIs and Machine Learning. *Information and software technology*, 75, 17-25

Todd McDonald, J., Nathan H., William B.G., and Ryan K. B. (2021). Machine Learning-Based Android Malware Detection Using Manifest Permissions. Proceedings of the 54th Hawaii International Conference on System Sciences.

Westyarian, Y., R., and Budiman D., (2015). Malware Detection on Android Smartphones using API Class and Machine Learning. The 5th International Conference on Electrical Engineering and Informatics 2015. August 10-11, 2015, Bali, Indonesia

Zhuo M., Haoran G., Yang L., Meng Z. & Jianfeng M., (2019). A Combination Method for Android Malware Detection Based on Control Flow Graphs and Machine Learning Algorithms (2019). *IEEE access* 7(2019). 21235-21245.