International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
***Article Received:** 25 August 2023 **Revised:** 12 October 2023 **Accepted:** 20 October 2023*

Detection of Android Malware using Feature Selection with a Hybrid Genetic Algorithm and Simulated Annealing (SVM and DBN)

Dr. E. Padmalatha[1], M. Venkata Krishna Reddy[2], T. Suvarna Kumari[3], Kabeeruddin[4]
[1]Department of Computer Science and engineering
Chaitanya Bharathi Institue of Technology
Hyderabad,INDIA
Epadmalatha_cse@cbit.ac.in
[2]Department of Computer Science and engineering
Chaitanya Bharathi Institue of Technology
Hyderabad,INDIA
krishnareddy_cse@cbit.ac.in
[3]Department of Computer Science and engineering
Chaitanya Bharathi Institue of Technology
Hyderabad,INDIA
suvarnakumari_cse@cbit.ac.in
[4]Department of Computer Science and engineering
Chaitanya Bharathi Institue of Technology
Hyderabad,INDIA
kabeeruddin2297@gmail.com

Abstract: Because of the widespread use of the Android operating system and the simplicity with which applications can be created on the Android platform, anyone can easily create malware using pre-made tools. Due to the spread of malware among many helpful applications, Android users are experiencing issues. In this study, we showed how to use permissions gleaned from static analysis to identify Android malware. Utilising support vector machines and deep belief networks, we choose the pertinent features from the set of permissions based on this methodology. The suggested technique increases the effectiveness of Android malware detection.

Keywords: Support vector machine, Deep beliefNetwork.

I. INTRODUCTION

1. INTRODUCTION

The popularity of Android and the widespread use of smartphones have drawn the attention of those who can use malware to carry out their destructive intentions. It is very simple to enter the smartphone market because anyone can make the application accessible to the general public through a variety of markets. Android is widely used in society, as evidenced by the fact that 73% of smartphones today run the Android operating system [1]. Therefore, we need a reliable way to stop malware from spreading on Android smartphones. Machine learning can be used to detect malware by obtaining features from Android applications. The vectorization of features such as network addresses, permissions, and API calls. Each feature set has the potential to play a significant role in the detection of malware. Permissions are among the most significant and influential features. However, since attackers can request and obtain all necessary permissions during installation [2], they do not stop malware from being installed. It is possible to tell whether an application is malware by analysing and managing the permissions of Android applications because malware, like any other application, needs the permission of smartphone owners to carry out its destructive activities. The three categories of feature selection techniques are filter, wrapper, and embedded. In order to sort features and choose the features with the highest rankings, filtering techniques are used as preprocessing. The wrapper methods' criterion to choose features, one of the search algorithms' predictive performance is used to locate the pertinent subset. An example of an embedded method is choosing a variable during training without separating the data into training and test sets [3].

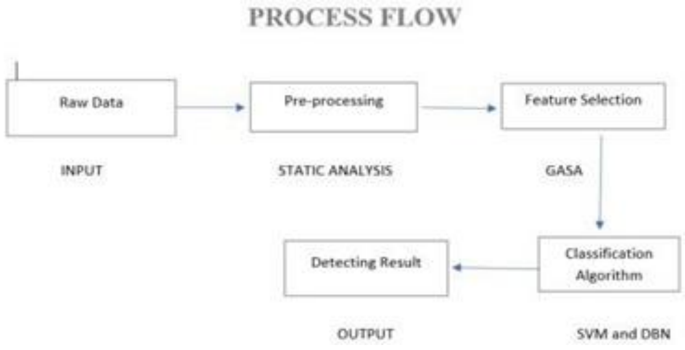


Fig.1: Android malware detection model

Both supervised and unsupervised classification algorithms are crucial for the creation of intelligent machine learning systems. In supervised classification methods, a model can be created and trained from a set of labelled data in order to classify new data [4]. In order to predict the classification of unknown feature vectors, machine classification algorithms are trained using feature vectors [5]. Algorithms for classification are employed to increase detection accuracy. In order to better detect Android malware, this paper combines evolutionary and classification algorithms. Support vector machine (SVM) and Deep belief network were used as classification algorithms, and a hybrid of the genetic algorithm (GA) and simulated annealing (SA) was used to select the features. The results of the current study show that the suggested approach can detect Android malware with greater accuracy.

2. LITERATURE REVIEW

A look at mobile malware in the wild

Mobile malware is increasingly posing a serious threat. In this paper, we examine the state of mobile malware at the moment. We examine the motivations behind 46 pieces of malware for iOS, Android, and Symbian that were active between 2009 and 2011.Additionally, the effectiveness of methods for identifying and preventing mobile malware is evaluated using this data set. After learning that four different pieces of malware use root exploits to launch sophisticated attacks on Android phones, we look into the motivations that lead non-malicious smartphone hackers to publish root exploits and survey the availability of root exploits.

A poll about feature selection methods

There are many feature selection methods in the literature because data with hundreds of variables are readily available, resulting in data with a very high dimension. In machine learning or pattern recognition applications, feature selection techniques help us shorten computation times, boost prediction accuracy, and gain a deeper comprehension of the data. This essay summarises a few of the techniques used in the literature. The objective is to give a general overview of the variable elimination method, which can be applied to a number of machine learning issues. We focus on Filter, Wrapper, and Embedded methods. By applying some of the techniques to common datasets, we also determine how feature selection can be used.

Machine learning classifiers for mobile malware detection are being evaluated.

A large number of people are using mobile technology as it has gained popularity and become an essential part of people's lives. Hackers are more likely to produce malicious software as user numbers increase. Furthermore, it is assumed that mobile devices will protect sensitive data. Since intelligent malware is constantly evolving and getting harder to detect, using currently available techniques is insufficient. This paper offers an alternative. An anomaly-based approach to malware detection uses a combination of machine learning classifiers. Basic data, content-based, time-based, and connection-based classifications were made among the various network traffic characteristics. Both public (MalGenome) and private (self-collected) datasets are used in the evaluation. According to evaluation results (TPR) on the MalGenome dataset, the Bayes network and random forest classifiers produced more accurate readings, with a 99.97% true-positive rate, in comparison to the multi-layer perceptron, which produced only 93.03%. This experiment, however, showed that the k- nearest neighbour classifier had a true-positive rate that was 84.57% higher than other classifiers for detecting the most recent Android malware.Using machine learning classifiers to detect malware on Android at scale .Due to their widespread use and context-sensitive nature, smartphone malware has become more and more popular. Malicious applications are currently being detected on smartphone systems using machine learning classifiers. This study examines a number of current classifiers using tens of thousands of real (as opposed to artificial) applications. We also introduce our stream framework, which was developed to enable rapid large-scale validation of machine learning classifiers for mobile malware.

An investigation into machine learning and malware detection methods for Android: Android OS is one of the most well-liked mobile operating systems. Adware and other malicious software are expanding at the same rate as mobile devices. There are many commercially available tools that stop the penetration and spread of malicious software. These tools are based on commercial signatures. Numerous studies claim that signature-based traditional detection methods systems are effective up to a certain point and that malware developers use a variety of techniques to evade these tools. Given this circumstance, a replacement, incredibly strong malware detection system is increasingly required to supplement and improve the signature-based system. Significant Recent research has focused on machine learning techniques that analyse malicious software characteristics and use those characteristics to classify and identify unknown malicious applications. The development of machine learning-based malware detection techniques for the Android operating system is summarised in this study.

Bayesian classification-based approaches to Android malware detection:
Mobile malware has grown in scope and complexity as a result of the continued global adoption of smartphones. The leading mobile platform, Android, is quickly gaining in popularity, which has led to an uptick in malware that targets the platform. Furthermore, Android malware is rapidly changing in order to evade detection by conventional signature-based scanning. Even with current detection technologies, timely detection of fresh malware is still a major issue. In order to combat the growing threat of unpatched Android malware, new strategies must be developed. As a result, in this essay, we develop and assess a proactive machine learning approach called Bayesian classification with the aim of discovering new Android malware through static analysis. The research, which is based on a sizable sample set of malware from the vast majority of families currently in use, demonstrates highly precise detection abilities. Empirical results and a comparative analysis presented here offer important insight into the development of effective static analysis programmes based on Bayesian classification for locating unknown Android malware.

3. METHODOLOGY

Anyone can easily create malware using pre-made tools thanks to the popularity of the Android operating system and the ease with which applications can be developed on the Android platform. Due to the spread of malware among many helpful applications, Android users are experiencing issues.

Disadvantages:
. Android users may experience issues with a variety of helpful apps.

In this study, we showed how to use permissions gleaned from static analysis to identify Android malware. In the suggested method, we choose the pertinent features from a set of permissions using SVM and DBN. The suggested technique increases the effectiveness of Android malware detection.

Advantages:
1. High accuracy.
2. Good results

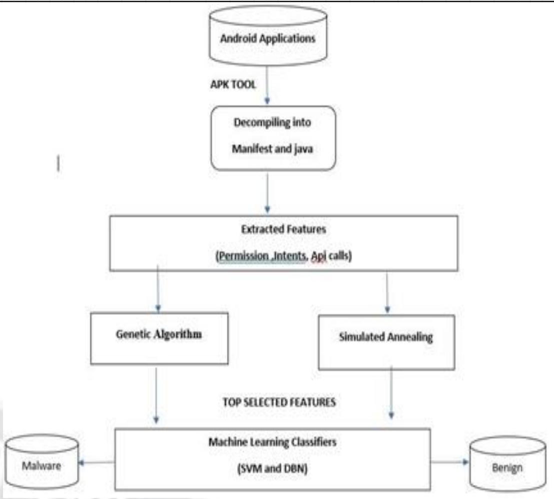

Android Applications
APK TOOL
Decompiling into Manifest and java
Extracted Features (Permission ,Intents, Api calls)
Genetic Algorithm
Simulated Annealing
TOP SELECTED FEATURES
Machine Learning Classifiers (SVM and DBN)
Malware
Benign

Fig.2: System Architecture

This paper proposes a method for extracting features, performing pre-processing, and identifying Android malware.

A. Dataset

The paper made use of the Drebin dataset. 545333 unique features and approximately

6.5 million duplicate features, taken from 129013 applications, make up this data set. The extraction features consist of ten feature set combinations. These attributes are organised into strings, such as network addresses, permissions, and API calls, and they are embedded in a shared vector space [16]. Data that has been transformed into vector space can be used for learning-based detection. Due to the large number of distinct features present in the data set, processing this many features (545333) will take a considerable amount of time on modern computers. As a result, we made the decision to limit our use in this paper to the permissions feature set. The high potential for malware detection, which is mainly used in other studies, was another justification forusing permissions. In total, there are 3812 permissions features.

B. Pre-Processing

Datasets Preprocessing operations are needed before features can be converted. appropriate vectors for machine learning algorithms After removing the features from the data set, the total features are calculated. divided into ten feature sets, with the following names and numbers: Table 1 lists them:

No.	Name	No. of features
1	Activity	185729
2	API Call	315
3	Call	733
4	Feature	72
5	Intent	6379
6	Permission	3812
7	Provider	4513
8	Real Permission	70
9	Service Receiver	33222
10	URL	310488

Table:1 Top Ten Features of Drebin Dataset

C. Feature Selection

The presence of unrelated data in the data set will lower classification accuracy. Eliminating irrelevant data is the main objective of the feature selection in order to increase accuracy. In order to find pertinent and appropriate features to increase classification algorithm accuracy, we will combine GA and SA algorithms. Below, each algorithm's procedure will be thoroughly explained. The fundamental genetic algorithm starts by creating a population of chromosomes at random. It then uses its operators to create a new population in the hopes of improving it. This process is repeated with each succeeding generation until the conditions for a halt are finally established. According to the definition of a genetic algorithm, the first step before developing the The first step in creating the initial population is determining appropriate coding.Due to the fact that the order of the features is irrelevant, only their presence or absence matters. As a result, Figure 3 shows that in the chromosome produced by the evolutionary algorithm, a feature with a one-valued value is present and a feature with a zero-valued value is absent.

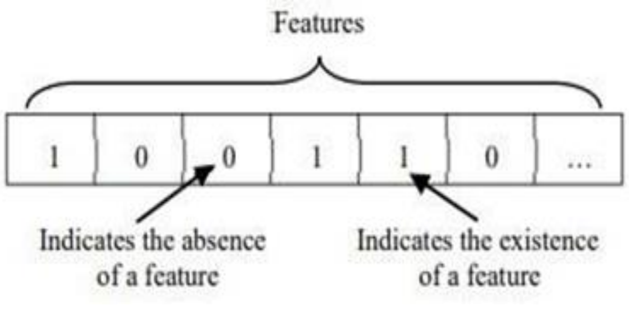

Features
1
0
0
1
1
0
...
Indicates the absence of a feature
Indicates the existence of a feature

Fig:3 Feature Vector in Genetics

The exact result of the fitness criterion is the fitness function. the accuracy of machine learning techniques like SVM, DBN The genetic flowchart is shown in Figure 4. The suggested approach uses an algorithm. The first population is used in the genetic algorithm. The fitness of each person in the population is then determined using the fitness function. Then a roulette wheel is used to select the parents. Then, using a crossover operator, the progeny of the current generation are created. The population's mutation operator is also used, and the children's fitness function is calculated. If the stop requirement—that is, the number of loop repetitions exceeds N—is met, the chromosome number with the highest value will be used. The fitness function's ultimate output is chosen. The algorithm will also select half of the population from the best members of the current generation as the initial population stage and create the other half if the end condition is not satisfied in order to continue looking for the best chromosome to usein machine learning algorithms.

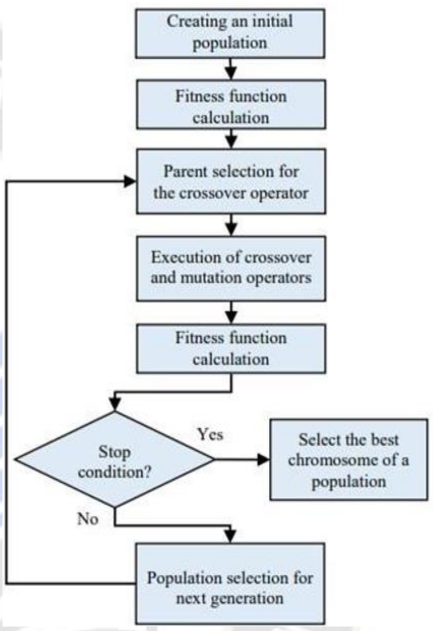

Creating an initial population
Fitness function calculation
Parent selection for the crossover operator
Execution of crossover and mutation operators
Fitness function calculation
Stop condition?
Yes
Select the best chromosome of a population
No
Population selection for next generation

Fig:4 Genetic Algorithm flow

One of the metaheuristic algorithms that enables you to find the best solution is the simulated annealing algorithm. The SA The cooling of the heated solid during the crystallization process serves as an inspiration for the gradual crystallization of the high-temperature solid. The advantages of this algorithm include guaranteeing an ideal solution, facilitating simple coding even for complex problems, and generally offering a good solution. A generalised model of the simulated annealing algorithm is shown in Figure 5, which first creates a population of neighbours from the top member of the population being created by the genetic algorithm. The stop condition for the temperature is then examined, and the best solution is found if the condition is satisfied, that is, the temperature is below zero. A machine learning algorithm calculates its fitness function, which is the measure of accuracy, by randomly choosing one of the neighbors if the

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
***Article Received:** 25 August 2023 **Revised:** 12 October 2023 **Accepted:** 20 October 2023*

condition is not met, at which point it enters the loop. The algorithm then checks whether the new solution is superior to the current one, and if it is, it will be selected as the current solution. If the solution is not better, acceptance is based on probability. To determine the probability of conditional selection, use equations (1) and (2).The local optimal can be avoided with conditional acceptance.

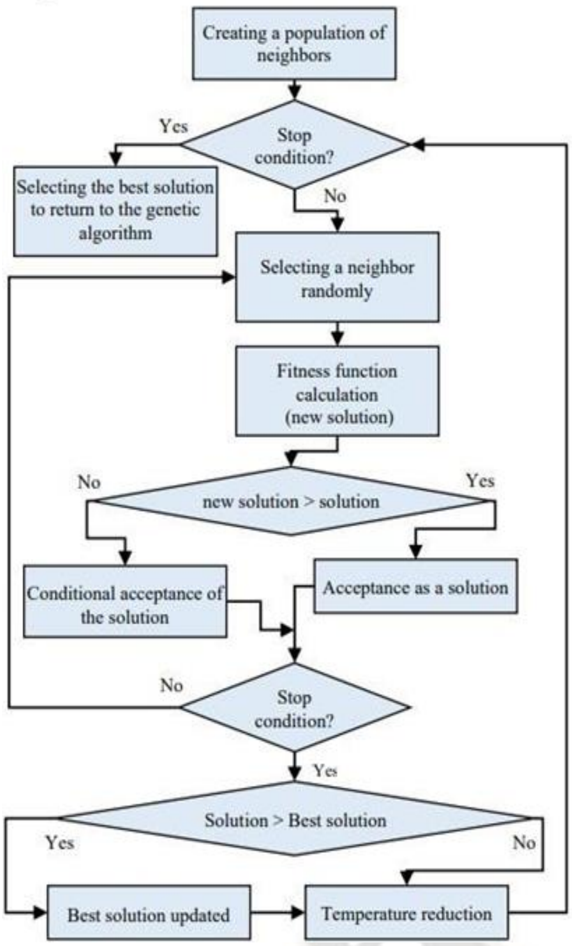

Creating a population of neighbors
Stop condition?
Yes
No
Selecting the best solution to return to the genetic algorithm
Selecting a neighbor randomly
Fitness function calculation (new solution)
No
Yes
new solution > solution
Conditional acceptance of the solution
Acceptance as a solution
No
Stop condition?
Yes
Solution > Best solution
Yes
No
Best solution updated
Temperature reduction

Fig:5 Simulated Annealing Algorithm

In addition, the neighbor is the point that differs from the current point in one of their features.

4. *CLASSIFICATION ALGORITHMS*

Figure 6 depicts an overview of the genetic algorithm and simulated annealing combination. The prerequisite for admission.The SA algorithm is comprised of Stagnation is the best result of the genetic algorithm's current generation. In other words, if the current generation of the genetic algorithm does not improve as a result of the previous generation, it will enter the SA.

.

SUPPORT VECTOR MACHINE:

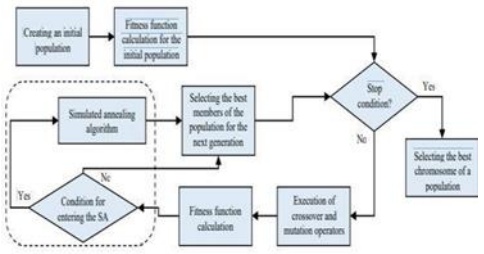

Creating an initial population
Fitness function calculation for the initial population
Stop condition?
Yes
No
Simulated annealing algorithm
Selecting the best members of the population for the next generation
Selecting the best chromosome of a population
No
Yes
Condition for entering the SA
Fitness function calculation
Execution of crossover and mutation operators

Fig :6 A model of the combination of a genetic algorithm and a simulated annealing algorithm

Both classification and regression issues can be solved using the SVM supervised machine learning algorithm. Despite being called regression problems, they are actually better suited for classification. The N-dimensional hyperplane that clearly classifies the data points is what the SVM algorithm looks for. SVMs are employed in a variety of applications, including web page generation, intrusion detection; face detection, email classification, gene classification, and handwriting recognition. For this reason, SVMs are employed in machine learning. It can handle classification and regression for both linear and non-linear data.

DEEP BELIEF NETWORK:

A neural network is a collection of algorithms that

locates underlying connections in a set of data in a manner resembling that of the human brain.

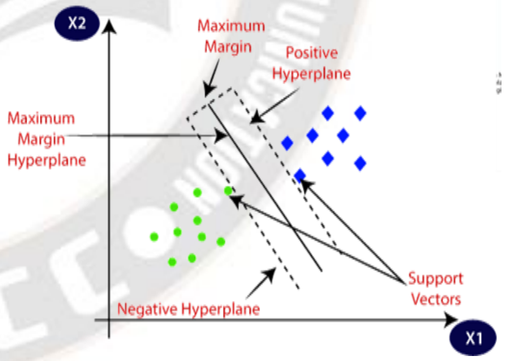


Fig.7: SVM model

Similar to how the human brain processes information, the deep belief network (DBN) does the same. Deep belief networks operate in a manner similar to that of their biological sources. With neurons serving as nodes and connections between neurons serving as weight edges, they can be compared to weighted directed graphs.

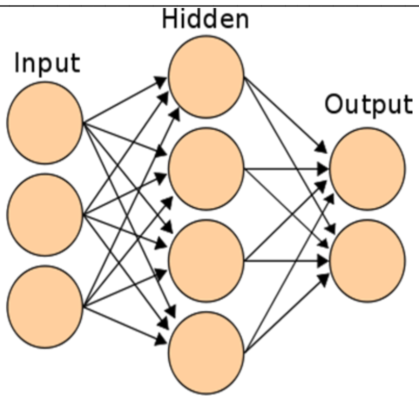


Fig:8 DBN Model

HIDDEN MARCHOV MODEL:

The probability of any random process is calculated or explained using a probabilistic model known as the Hidden Markov model. It basically states that a set of probability distributions will be related to an observed event rather than describing the progression of the event step by step. A hidden Markov model (HMM), a type of statistical model, can be used to explain how internal, erratic factors affect observable events. The invisible force known as a "state" or "symbol" in this context, is what underlies the observed event.

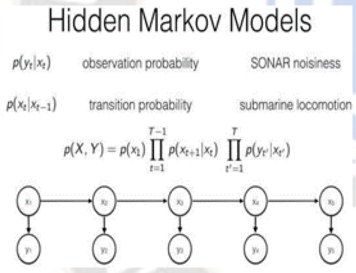

Hidden Markov Models
$p(y_t|x_t)$ observation probability SONAR noisiness
$p(x_t|x_{t-1})$ transition probability submarine locomotion
$p(X,Y) = p(x_1)\prod_{t=1}^{T-1} p(x_{t+1}|x_t) \prod_{t'=1}^{T} p(y_{t'}|x_{t'})$
x_1 x_2 x_3 x_4 x_5
y_1 y_2 y_3 y_4 y_5

Fig.9: HMM model

5.EXPERIMENTAL RESULTS

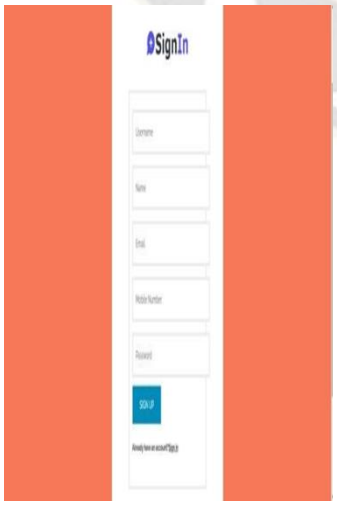
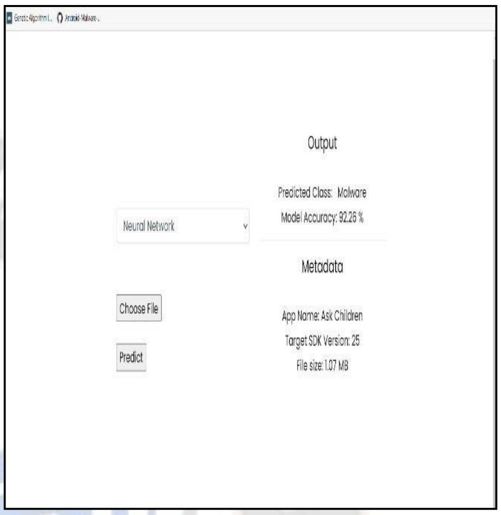


Fig.10: Registration

Fig.11: Login



Output

Predicted Class: Malware
Model Accuracy: 92.26 %

Metadata

App Name: Ask Children
Target SDK Version: 25
File size: 1.07 MB

Fig.1 2Upload data

6. CONCLUSION

The rise of Android malware can be a serious threat due to the growing popularity of the Android operating system among users. Simulated annealing, the genetic algorithm, and machine learning were used in this study to detect Android malware. In the proposed method, the genetic algorithm and the simulated annealing algorithm are combined. On the one hand, we want to reduce the number of features, and on the other, by choosing the best subset of features, we want to increase classification accuracy. Reduced training time and improved malware detection performance in classification algorithms are the results of selecting the best subset of features, which increases accuracy while reducing time and complexity.

7. FUTURE WORK

Future research in this area may find success by figuring out how to combine permissions with other sets of application features to enhance the accuracy criterion.

REFERENCES

[1] "os-market-share", [Online]. Available: https://gs.statcounter.com/osmarketshare/mobile/wor ldwide. [Accessed 13 06 2020].

[2] Felt, Adrienne Porter, Matthew Finifter, Erika Chin, Steve Hanna, and David Wagner. "A survey of mobile malware in the wild." In Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices, pp. 3-14. 2011.

[3] Chandrashekar, Girish, and Ferat Sahin. "A survey on feature selection methods." Computers & Electrical Engineering 40, no. 1 (2014): 16-28.

[4] Narudin, Fairuz Amalina, Ali Feizollah, Nor Badrul Anuar, and Abdullah Gani. "Evaluation of machine learning classifiers for mobile malware detection." Soft Computing 20, no. 1 (2016): 343-357.

[5] Amos, Brandon, Hamilton Turner, and Jules White. "Applying machine learning classifiers to dynamic android malware detection at scale." In 2013 9th international wireless communications and mobile computing conference (IWCMC), pp. 1666-1671. IEEE, 2013.

[6] Baskaran, Balaji, and Anca Ralescu. "A study of android malware detection techniques and machine learning." (2016).

[7] Yerima, Suleiman Y., Sakir Sezer, and Gavin McWilliams. "Analysis of Bayesian classificationbased approaches for Android malware detection." IET Information Security 8, no. 1 (2014):25-36.

[8] Shabtai, Asaf, Yuval Fledel, and Yuval Elovici. "Automated static code analysis for classifying android applications using machine learning." In 2010 international conference on computational intelligence and security, pp. 329-333. IEEE, 2010.

[9] Sanz, Borja, Igor Santos, Carlos Laorden, Xabier Ugarte-Pedrero, Pablo Garcia Bringas, and Gonzalo Álvarez. "Puma: Permission usage to detect malware in android." In International Joint Conference CISIS'12- ICEUTE 12-SOCO 12 Special Sessions, pp. 289-298. Springer, Berlin, Heidelberg, 2013.

[10] Peiravian, Naser, and Xingquan Zhu. "Machine learning for android malware detection using permission and api calls." In 2013 IEEE 25th international conference on tools with artificialintelligence, pp. 300-305. IEEE, 2013.

.