International Journal of Innovations in Engineering
http://dx.doi.o

Malware Detection in Android Device
System calls under Dynamic Anal

Leesha Aneja[1], Sakshi Babbar[2]
[1,2]GD Goenka University, Gurugram, India

Abstract- This paper is focused on Android malware detection using system calls under dynamic two main analysis techniques utilized for Android Malware detection are:Static Analysis and Dy analysis ,which is used for Android malware detection makes use of signatures to detect malic features are extracted from the application without executing it. It can accurately detect malware by from test data and then comparing the test data with the signature samples of virus and ben examination experiences code obfuscation procedures which the Malware creators utilize to sidestep strategies. Therefore, it is important to focus on dynamic analysis where code of an application is a it's execution. System calls have been heavily utilized to detect malicious behavior of applica Analysis.As,Current state-of-the-art research shows that recently, researchers and other organiz machine learning methods for malware analysis and detection .Hence, this work is focused on obs logs, constructing the robust dataset utilizing the same and classifying application as benign or ma machine learning models.Moreover,it involves validating the performance of these models usin metrices and identifying the best predictive model.An experimentation is further done based on eventually confirm that an application from a certain category demands similar system calls utilize in that category.Our analysis reveals the similarities and differences between benign and malware s applications of certain category and shows how frequenciesof these system calls help us in anal malicious activity during run time.

Keywords:Malwares,Android Malware Detection,Static Analysis, Dynamic Analysis,System Calls,M

I. INTRODUCTION

Mobile phones have become the necessity of modern human lives to store our valuable passwords, reminders, messages, photos, videos and social contacts. The advent in mobile human life easier and more efficient. However, at the same time, our excessive dependency o drawn attention of malware authors and cyber criminals leading to large number of cyber-atta the digitization of trivial daily-life tasks ,people have become highly dependent on the mobile p in the mobile market are - Apple's iOS and Google's Android that brings new security techn additional features. iOS malware rate in comparison to Android Malware is not too acute. A concern of security threat is on Android smartphones. The key reason for it is that it is open download applications from unsafe sites. So, it is important to develop robust and efficie detection system in order to protect our sensitive data from cyber-attacks on Android platform[
As Android has become the prime target for cybercrimes by means of malware and viruses.A market share of android system gives us the view of its comparatively higher user base. The ke does not restrict its users to install the applications from unsafe sites apart from the official sto updates over the past few years, security remains the ultimate battleground in the field phones.There is a constant increase innew android malware samples every passing year a Securing Android mobile devices from malicious applications, have become an active area of r few years.

Lot of Emphasis has been given on Android Malware detection.There have been different app which are broadly categorized into Static and Dynamic.Static analysis , makes use of signature applications. The features are extracted from the application without executing it. It can accu by extracting signatures from test data and then comparing the testdata with the signature benign samples.Itis an approach that includes analyzing the code of an application without applications are stored in .apk file. This .apk file is a zip heap of AndroidManifest.xml, clas other files. Reverse Engineering is utilized for feature extraction. This is done using different AndroidManifest.xml document contains a great deal of permissions that are used for st philosophy is asset and time productive as the application is not executed. As mentioned,thes static components are the Permissions. Since these are isolated from the application AndroidM the malware area rate to a high degree, extensive research has been made with these as co combined with various components expelled from meta-data open in Google Play-Store, for in version no., author's name, last updated time, etc.DREBIN [23] presents a wide static inve features from the Manifest file including intent filters using Support Vector Machine(SVM) a algorithm. The consequences of the examination appeared that DREBIN distinguished 94% false alarm.But, this examination experiences code confusion procedures ,the Malware creat from static discovery strategies. One of extremely mainstream avoidance procedure is th application is introduced on the cell phone and when the application gets an upgrade, the m downloaded andintroduced as a component. This cannot be identified by static investigation filter just the considerate application. Thus, dynamic analysis came as a solution for this proble Dynamic Analysis,which is also referred to as behavioral analysis, is utilized to study and behavior of applications . As a rule, this procedure checks for API calls, framework calls, sy ,network traffic and so forth. This strategy is valuable when the source code of an applica main fundamental building block of dynamic investigation is system calls.In computing programmatic way in which a computer program request a service from the kernel of the android uses Linux 2.6 kernel, applications make use of services of kernel with the help instance, whenever a user wants to make a call through dialer application, telephony m framework receives the request. The user call is then converted in library call by Dalvik Vir that finally results in various system calls to kernel. Thus, request from all applications is pass interface before the execution. There are more than 250 types of system calls utilized by And like allocating memory, accessing files etc. Capturing these calls give the detailed behavi Furthermore, frequency of occurrence of system calls is considered/ taken as the prope application’s behavior.It has been heavily utilized to detect malicious behavior of applica Analysis.

Our approach is mainly based on system call log generation. In our study, the system call log i benign and malware application is collected with the help of an environment like Genymo system calls and creation of robust dataset.Then after,with the help of machine learning classified applications as malware or benign.AsMachine learning is an application of artificial provides systems the ability to automatically learn the data model and make predictions. The was to determine the malware on the basis of the behavior of system calls by using classificati in good accuracy and identification of the best predictive model for this study.But this strategy for prediction.So, there was a need to understand whether similar category based applications i calls.If that is the case,then malware prediction can be done in a better way by formulating the category based application and testing an X application from that category to classify it as m comparing it with its general behavior of invoking system calls.So an attempt was made to e

3. To integrate machine learning models with domain of Android and to validate the models to predict malware attacks.
4. To perform deep analysis of utilization of system calls by benign and malicious applica
5. To analyze the key patterns of various system calls by dividing the applications in cate

The further organization of the paper proceeds as follows.Section 2 provides literature review the Dynamic Analysis and the need of Machine learning to capture Malwares.Section 4 illustra Section 5 showcases results and discussions.

II. RELATED WORK

The capability to early distinguish malware applications is essential to ensure user's security tagged, reported, and removed from the market and their signatures can be black-listed. Thi classification problem and, accordingly, many authors have utilized machine learning over Android application. In [37], the authors use permissions and control flow graphs along Machines (SVMs) to differentiate malware from good applications ("goodware" in what follov explores the intents of each application as features for the classification task.

As discussed,our aim is to classify the unknown sample as benign or malicious based on system calls using machine learning.In computing,a system call is the programmatic way program request a service from the kernel of the Operating System.A system call is a way f interact with OS.It has been heavily utilized to detect malicious behavior of applications und There has been a lot of research in the field of Malware Detection. In [7], a novel dynamic an Component Traversal is proposed that can automatically execute the code routines of application (app) as completely as possible. Taindroid is another dynamic examination framew system information for breaking down applications. In another examination by the creators of a malware recognition instrument, in view of following frame-work calls and order them learning calculations.Shabtai et al. discussed a behavior based anomaly detection system for deviations in a mobile applications network behavior by detecting mobile malware with self- The detection of such can be performed based on applications' network traffic patterns only.M utilized system call features for malicious application detection .In their work Schmidt et al.[2 detection system that tracked the system activities through process list of open files, network and system call traces to find any abnormal behavior. Kolbitsch et al. [2] performed an analysis families by finding the correlation between them in terms of the System Call. A.lanziet al. [4] detection system on the analysis of System Call invoked by the application, and achieved the d Authors considered various algorithms such as KNN, SVM, J48, Random Forest etc. Sato e method of calculating the malignancy score of the android application based on the informati filter (action), Intent filter (category), and Process for classification of android applications and 91.4 %. Huang et al. [8] also used machine learning technique for classification of Android A maximum accuracy of 81 % with J48. Canfora et al. [9] discussed about malware detection a analysis of System Call and permission feature, and classified the malicious application.

Sapna Malik et al.[4], explored the behavior through system call hint of 345 malicious applic learning.In our work,we have used different supervised algorithms because of supervis Neighbor (KNN) classifier is one of the Non –parametric machine learning algorithms that data. It utilizes a database in which the data points are separated into classes to predict the cl sample point.This strategy classify an unknown sample dependent on the class of the instanc training space by estimating the separation between the training instances and the unknown s similarity between data points which is measured using distance metric. For example in the

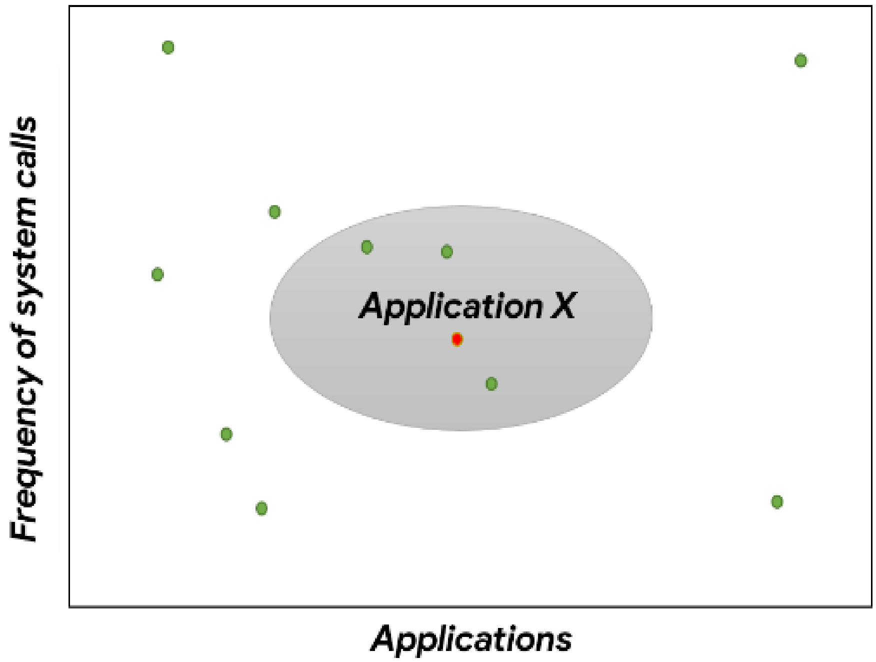
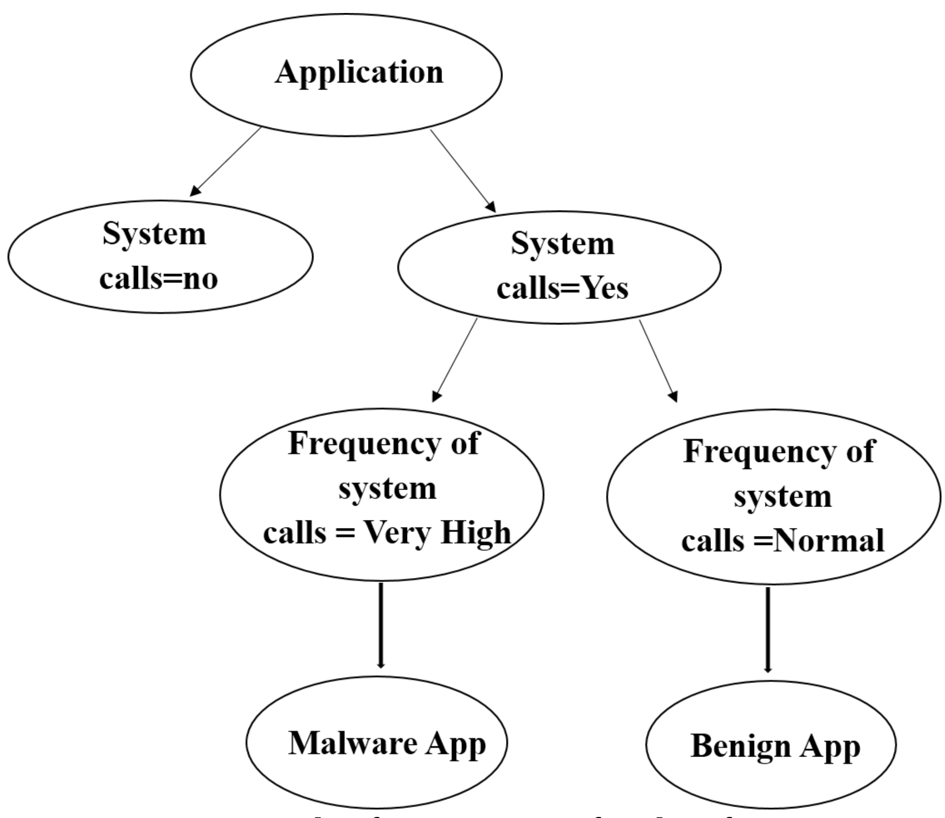


Fig. 2.Example of KNN algorithmfor classification

Decision Tree classifiers are another sort of AI classifiers that work on supervised data and a
nature and are graphically represented as trees. Interior nodes indicate conditions with respe
problem, while last nodes or leaf speak about ultimate decision of the algorithm. The authors
and permissions as features to train SVMs and Decision Trees (DTs).One of the example is
figure 3 where an application which invokes a system call with higher frequency than norm
denoted by leaf node as malware application.



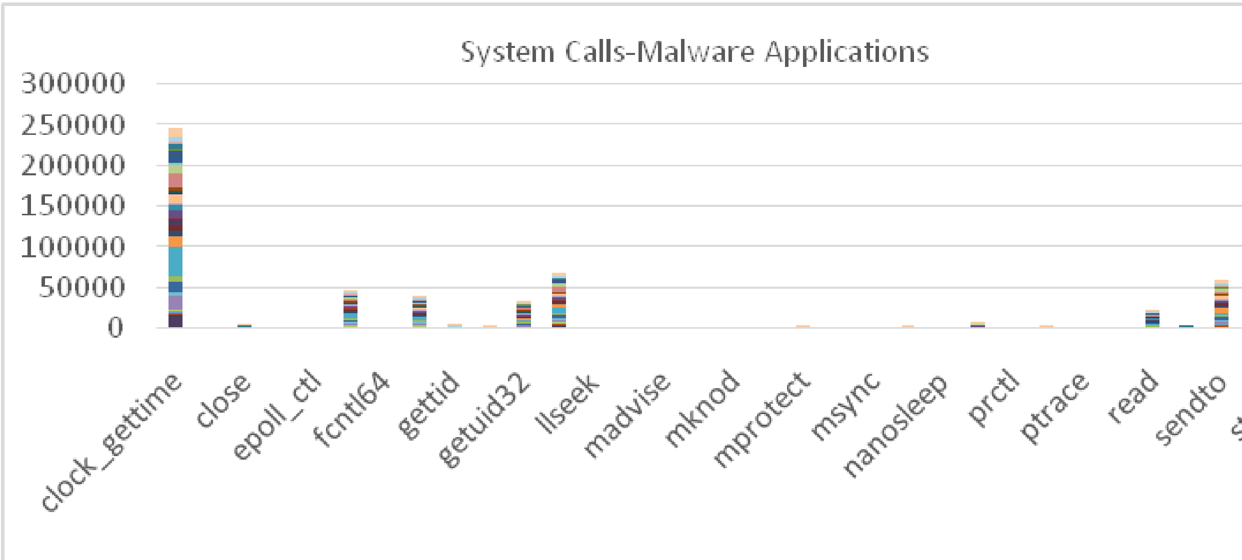


Fig. 4.Example ofpresence of various system calls in Malware application

Naïve bayes classifiers were our other choice and are characterized as probabilistic models f have the significant capacity to decide the likelihood of an application being malware. The au Bayesian-base machine learning techniques for Android malware detection.Naive Bayes mode particularly useful for very large data sets.

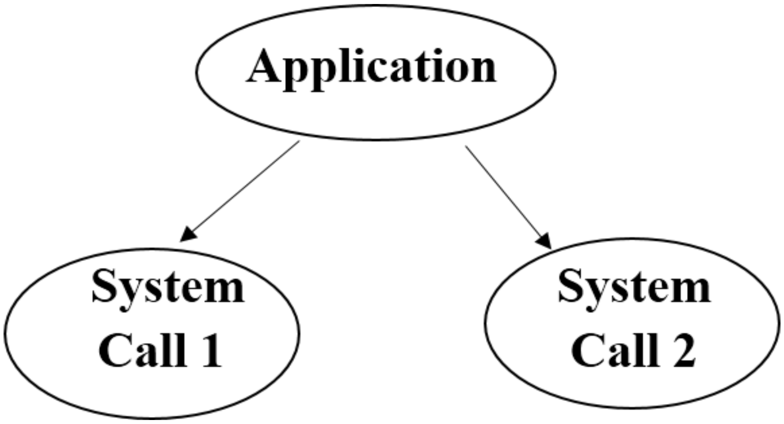


Fig. 5.Example of Naïve Bayes algorithm for classification

Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classif assume features are independent.It is illustrated in the figure 5,where an application is categ benign through behavior of system calls 1 and 2.Let us assume these system calls be read(interdependent on each other. In such cases, naïve bayes will be unable to classify becaus ofsystem calls on each other.As Naïve bayes takes conditional independence as assumptic classification where two classes are involved ,like our case of malware and benign.
The authors in [23] gather features from application code and manifest (permissions, API calls, Vector Machines (SVMs) to identify different types of malware families. SVM calcula dimensional space representation of the data into two locales utilizing a hyperplane. This hy boosts the edge between those two locales or classes. The margin is characterized by the m between the instances of the two classes and computed dependent on the distance between th the two classes, which are called supporting vectors .Being a supervised algorithm,it has

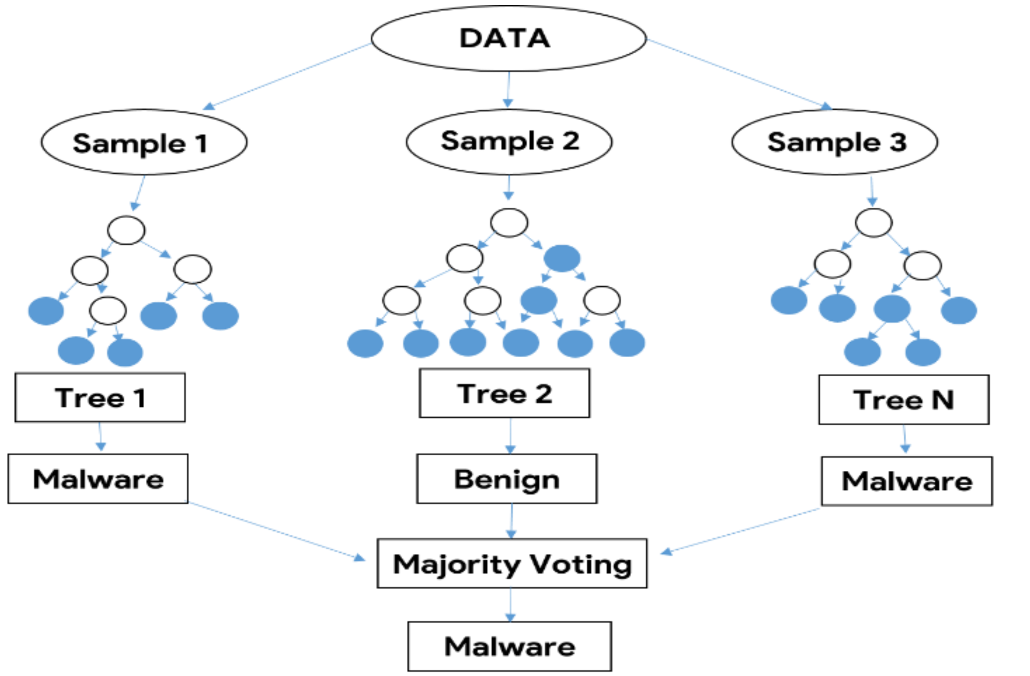


Fig. 6.Example of Random Forest for classification

III. DYNAMIC ANALYSIS

Dynamic Analysis,also known as behavioral analysis, includes studying and analyzing the ap
of their execution . Generally, this procedure can include API calls, system calls, IP address ,n
network traffic and system calls are being frequently used for dynamic analysis. Monitorin
mobile devices is one of the ways of detecting the malware ,as applications send and receive d
and the same can be utilized for leaking data to attackers maliciously. Shabtai et al.[22] discus
anomaly detection system for detecting meaningful deviations in a mobile applications n
detecting mobile malware with self-updating capabilities. The detection of such can be
applications' network traffic patterns only. The other fundamental building block of dynamic in
calls. A system call is the component through which a user cooperates with the kernel in t
activity to be performed. Likewise in Android ,interaction is done by the user with OS through
Researchers have utilized system call features for malicious application detection.In their wor
proposed intrusion detection system that tracked the system activities through process list o
traffic, symbol table and system call traces to find any abnormal behavior. Kolbitsch et al. [2] p
of different malware families by finding the correlation between them in terms of the System C
The general system calls used by malicious and benign applications are OPEN(opening a fi
file),GETID(related to app ID),etc.These system calls are common and are likely to be issue
irrespective of malware and benign applications. As there are more than 250 system calls w
applications, system calls utilized by our datasetare explained in Table 1 below.

Sno	System call	Description
1	Access	Check user's permissions for a file
2	Brk	Change the location of the program break, which defines the end data segment.
3	Chmod	Change permissions of a file
4	Clock_gettime	Retrieve the time of the specified clock.

16	Epoll_wait	Wait for an I/O event on an epoll
17	Writev	Write data into multiple buffers
18	Sched_yield	Yield the processor
19	Nanosleep	High resolution sleep
20	Sigprocmask	Examine and change blocked signals
21	Munmap	Deletes the mappings for the specified address range
22	Fsync	Synchronize a file's in core state with storage
23	Pread64	Read from a file descriptor at a given offset
24	Stat64	Get file status
25	Close	Close a file descriptor by the kernel
26	Dup	Creates a copy of a file descriptor.
27	Epoll_ctl	For a scalable I/O event notification mechanism
28	Fcntl64	Open file descriptor fd
29	Fdatasync	Modified data of fd to be moved to a permanent storage device.
30	Flock	Applies or removes an advisory lock on the file associated with t
31	Fstat64	Get information from the file specified by filedes and stores it in pointed to by buf .
32	Ftruncate	Regular file named by path or referenced by fd to be truncated to precisely length bytes.
33	Futex	Implement basic locking, or as a building block for higher-level
34	Getdents64	Reads several linux_dirent structures from the directory
35	Getlimit	Get and set resource limits.
36	Getpriority	Obtain the nice value of a process, process group, or user.
37	Getsockopt	Manipulates options associated with a socket.
38	Gettid	Gettid() returns the caller's thread ID (TID).
39	Gettimeofday	Can get and set the time as well as a timezone.
40	Llseek	Implements the lseek and llseek system calls.
41	lstat64	All of these system calls return a stat structure
42	Madvise	Give advice about use of memory
43	Mkdir	Attempts to create a directory named pathname
44	Mknod	Creates a filesystem node
45	mmap2	Asks to map length bytes starting at offset offset
46	Mprotect	Function shall change the access protections
47	Mremap	Expands (or shrinks) an existing memory mapping
48	Msync	Flushes changes made to the in-core copy
49	Prctl	First argument describing what to do
50	Ptrace	Provides a means by which one process may observe and control another process
51	Pwrite64	pwrite() became pwrite64() in kernel 2.6
52	Rename	Change the name of the file or directory
53	Setpriority	Scheduling priority of the process, process group, or user,
54	Statfs64	Statfs() and fstatfs() system calls were not designed with extreme mind

application denotes abnormal behavior.As in figure 8 ,frequency of a system call is sim applications which showcases normal behavior in comparison to figure 9,where there is lot of of sample S1 in contrast to S2,S3,S4.

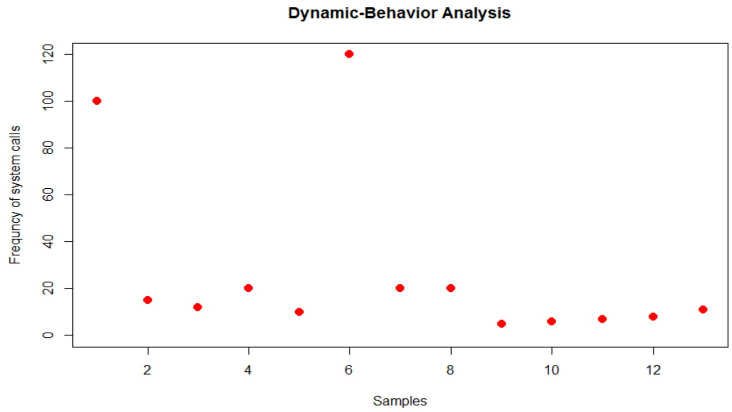


Fig. 7. Behavior Analysis of Applications

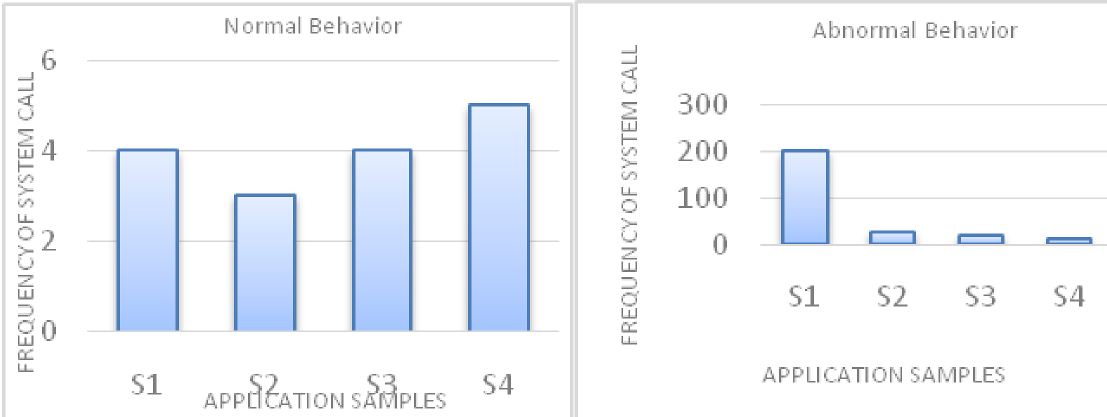


Fig. 8.Example of Normal behaviorFig. 9.Example of Abnormal behavior

Formulation of dynamic analysis involves a series of steps.The different steps followed for the follows:

1. Initializing the emulator and launching it with android (nexus 5).
2. The application is then installed on the emulator.
3. Strace command is then executed for hooking the system call on emulator for an interv
4. Frequency of all system calls utilized by the application during the execution gets colle
5. Dataset is generated using the applications and frequency of their calls.Furthermore,Machine algorithms are applied and an application is tested for a ma

Recently, researchers and other organizations prefer applying machine learning methods for detection. AsMachine learning is an application of artificial intelligence (AI) that provides s automatically learn the data model and make predictions. It enhances the decision making conformity of an application being a malware or a benign application.

Our work has been illustrated in figure 10 below where different samples of malware and ben run through Geny motion and their frequency of system calls are extracted,thus generating a da of malware and benign.To test any application,test data is inputted into classification model made for malware or benign applications.

IV. METHODOLOGY

Our work involves developing a robust environment with the help of an emulator named device) for running each application, to protect our own devices from getting affecte application.Each application is executed to observe its behaviour. This involves system call 'strace' and further creating a robust dataset based on the same.As discussed , the system c between the user and the kernel. This means all requests from the applications will pass thro Interface before its execution through the hardware. So capturing and analyzing the system malware detection.Let us consider an example in figure 11, where on x-axis,there are n applica frequency of invoking system calls on y-axis reflects the application behavior for benign or ma

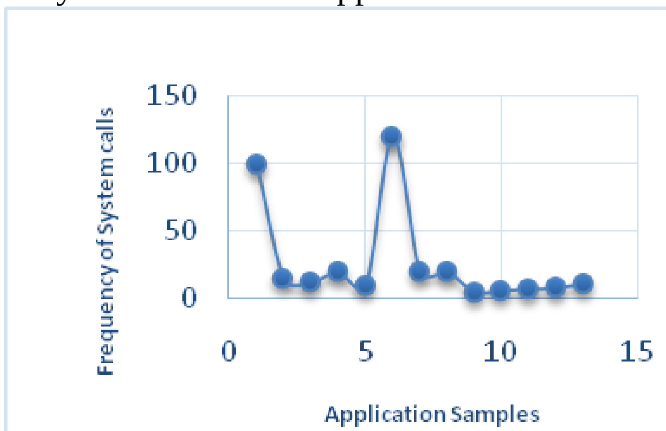


Fig. 11.System calls representing behavior of Applications

Though,System calls like OPEN(opening a file),CLOSE(closing a file),GETID(related to app and are likely to be issued by all applications irrespective of malware and benign applications. sendto(), recvfrom() which are used for sending and receiving data from the socket are often Further, the process control related system call like ptrace() is used for process tracing and processes, and the sigprocemask() is used for blocking signal to the process, wait4(), futex process id, getuid() for getting user id of the owner of the process, prctl() for controlling exe are also heavily used. Sapna et al.[4] also found that the malware also executes the system call reading data from the files stored on phone and SD memory like write(), read(), ioctl(), fcntl open(), mmap(), munmap(), lseek(), dup() etc.

To understand the behavior of an application, we have utilized machine learning algorithms part of Artificial Intelligence that generates new calculations to sum up behaviors utilizi learning models learn and explore data, find relevant patterns in data and predicts similar patte are different types of machine learning,but we have considered supervised learning in our utilized is supervised and have labels of malware and benign samples.So,we have utilized algorithms.In Supervised learning method, the historical data consists of expert knowledge in t corresponding outputs with labels, and is used to train the models and based on the patterns performs classification.Classification is a technique to categorize our data into a desired an classes where we can assign label to each class.Classification with only 2 distinct classe outcomes is referred to as binary classification.In a binary classification problem, we are ofte with labeled data {xi , yi} ntr i=1, where yi∈ {0, 1} and xi is a vector containing the values features, namely, xi = (xi1, . . . , xiP). In our case, System callsfall under predictors.Machine le responsiblefor constructing a function from the training set that separates the two classes popular classification techniques namely k-Nearest-Neighbors(kNN), Decision Trees(DT), N

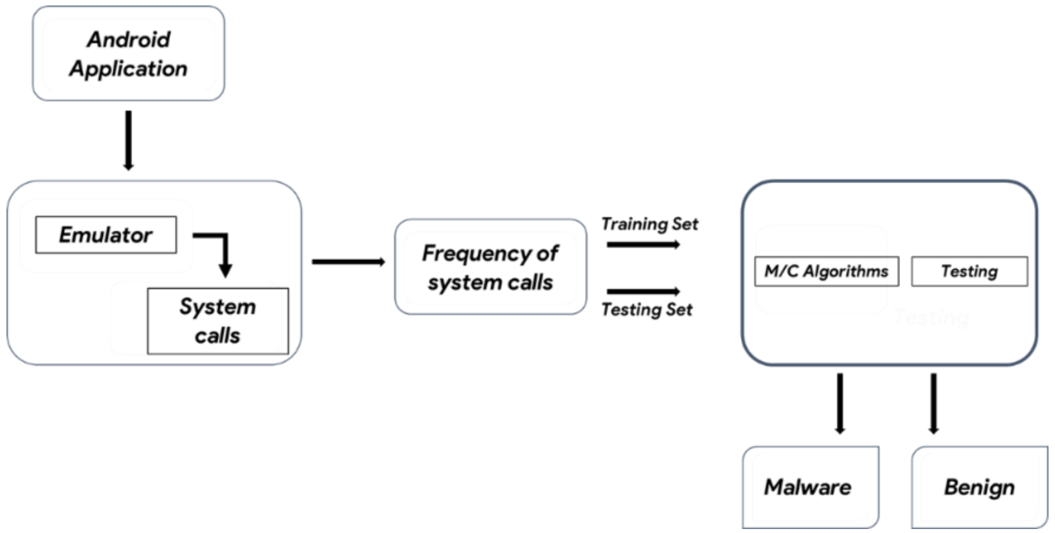


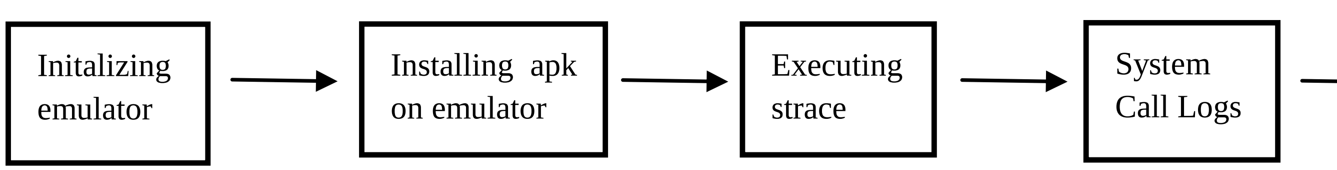
Fig. 12.Steps Involved in Behavior Analysis

But this strategy alone is not sufficient for malware prediction and there is a need to unders category based applications invoke similar system calls.As malware prediction can be don formulating the behavior of a certain category based application and testing an X application classify it as malware or benign by comparing it with its general behavior of invoking syster analyzing the system call based on certain category of an application can give further informati of a specific type of application,which is lateron considered in our experimentin detail.The re section 5.

V. RESULTS AND IMPLEMENTATION

As noted in the introduction, several researchers have studied different permissions used b different strategies to detect malware. In order to evaluate the effects that system calls hav applications, we have used the well-known R open-source statistical software, along with a n machine learning models (randomForest, e101,and caret).Generation of dataset is a p construction.In our work,the original dataset is built using 1000 applications each for benign a if data mining is required as a tool to uncover patterns in data,then dataset should be large en patterns.

The system calls have been separated in the Dynamic examination stage from the applicatio system calls is recorded to detect the presence of malware with the help of machine lear purpose of this work was to determine the malware on the basis of the frequency of system ca top of Sandbox environment and utilization of classification methods that result in the best pre As mentioned before,to accomplish the entire process, we have utilized the Geny Motion execute every Android application in emulator .Furthermore, the system calls are recorded w introduced in the emulator. This procedure records the frequency of system call logs,thus h dataset as shown in figure 13.



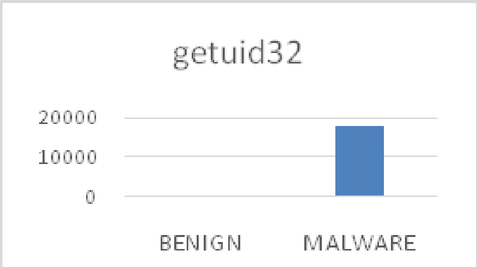
training set(80%) and a test set(20%) respectively.The models were evaluated on test data a
recorded using the above metrices.Since the approach of each algorithm is different,eva
algorithms is important to find out which one is better.We can clearly justify the quality of th
algorithms are able to identify a considerable number of instances.The overall misclassificatio
very low,indicating that classifiers performed really well. The results show that algorithms ac
but performance was slightly better in the case ofk nearest neighbor (kNN),Decision Tree(DT
(RF).
As KNN classifier operates differently and does not learn anything from data rather finds a gro
training set that is closest to the test object.It does not rely on the knowledge of domain.It simp
between two features in order to make classification decisions.Random forest also performed e
accuracy of 1 and correctly predicted the actual class due to majority of decisions taken into c
different decision trees.

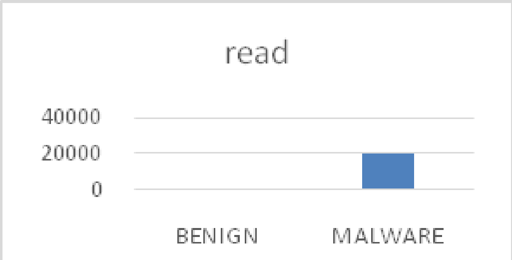
Table 2 presents the performance evaluation of different classifiers used in this study.It hel
which algorithm is more applicable for the Android malware detection. The experimental resul
malware and benign apps indicate good average accuracy rate using Naïve Bayes,KNN,RF,SV
respectively.Dynamic analysis results find no significant difference in the detection accuracy
naïve bayes algorithm gives more false positives (benign apps flagged as malware) as a
malware more comprehensively.Besides,parametric nature of this classifier,it is also prone to p
as bias. Overall, when the frequency of system calls are considered as features, there is mini
detection performance of other algorithms with respect to accuracy and true positive rate as
conclusion, analyzingfrequency of system calls offer a moderate approach to detect Android ma

Metrices	Naive Bayes	KNN	Random Forest	SVM	Decisi
Accuracy	0.91	1.0	1.00	0.99	1.00
Precision(p)	0.90	1.0	1.00	1.00	1.00
Recall(r)	0.91	1.0	1.00	0.97	1.00
F measure	2.7	3.0	3.00	3.00	3.00

Table 2.Performance of Different Algorithms

The main goal was to develop the proof of concept for the machine learning based malware cl
utilized for the extraction of the behavior of the samples, which was used as an input to
algorithms. The accuracy was measured for the case of detection of whether the file is maliciou
which method performs better was made.
As the top system calls used by malicious and benign applications are OPEN(opening a fi
file),GETID(related to app ID),etc.These system calls are common and are likely to be issue
irrespective of malware and benign applications.But our work found out (as illustrated in
frequency of system calls such as Getuid,read,sendto,getpid,recvfrom reflected the presence of





5.1 Category based analysis of Applications

Our analysis reveals the similarities and differences between benign and malware syste
applications of certain category and shows how frequency of these system calls helps us in anal
malicious activity during run time.Thus, making malware detection more effective and easier.

As Malicious applications usually makes use of different permissions to launch malicious activ
with system calls. As there are hundreds of system calls in Android system ,different applicatio
requirements of system calls .To prove this fact,an experiment was done and 25 samples of
each for Banking and Gaming applications which belong to two differen
collected.Ourworkthenincluded comparing the system calls of benign Application of
Applications) with benign application of Category 2(Game Applications) .The system calls in
benign Banking applications includedaccess,clone,dup,ioctl,recvfrom,sches_getparan,writev,
system calls invoked by benign Gaming application included access, brk, clock gettime, clo
getid, getrlimit, llseek, mkdir, munmap, prctl, read,sched_yield,pread64 ,write,etc.As shown i
system calls were similar in both the cases of Gaming and Banking application like access,clon
which are being utilized generally to check users' permissions for a file,to create child process
descriptor,to open the file for reading/writing and to write data into multiple buffers.But the
which were being taken by gaming application samples and not by banking application sample
like fchown32,futex,pread64,makedir,getrlimit and llseek were taken by gaming applicatio
banking applications as it involves changing ownership of file,basic locking,getting and
etc.Thus ,different categories of applications can vary in terms of their demands of system calls

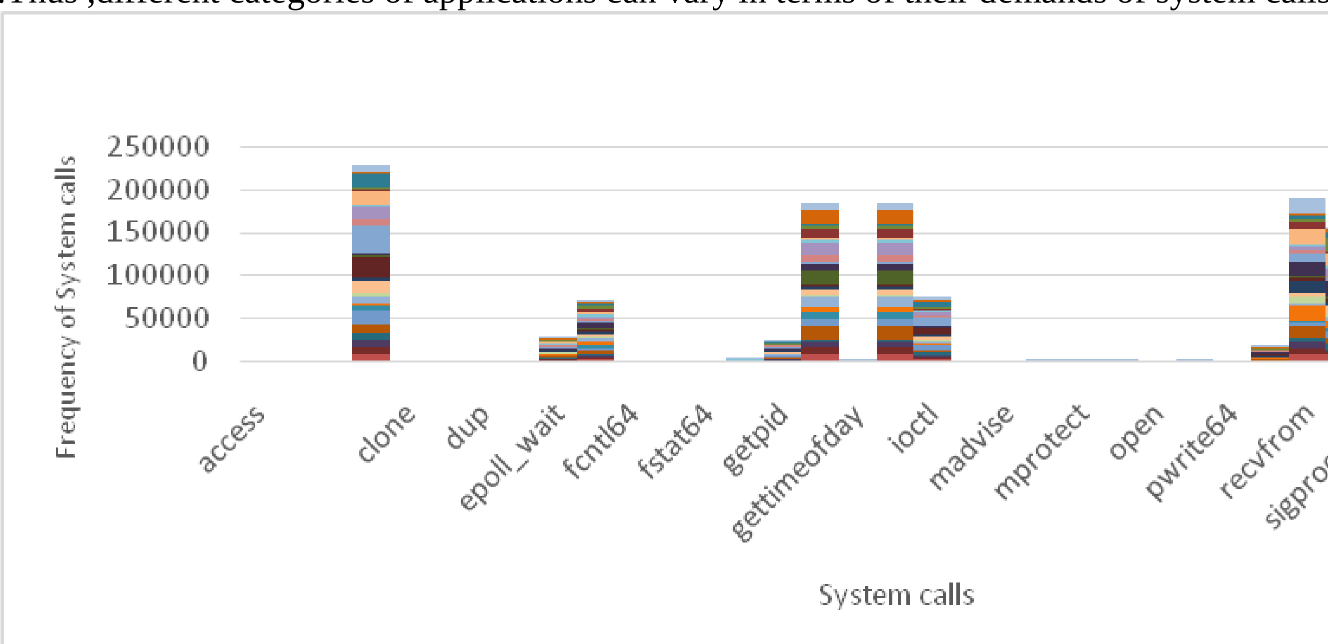
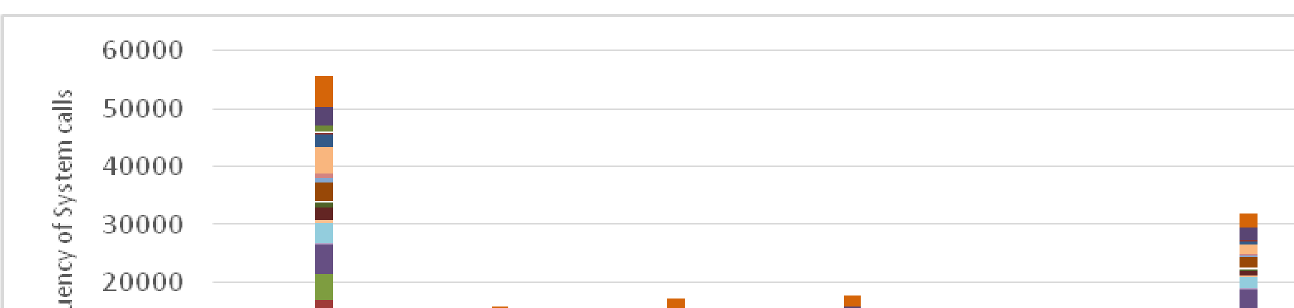


Fig. 15. System calls invoked by Benign Bank applications



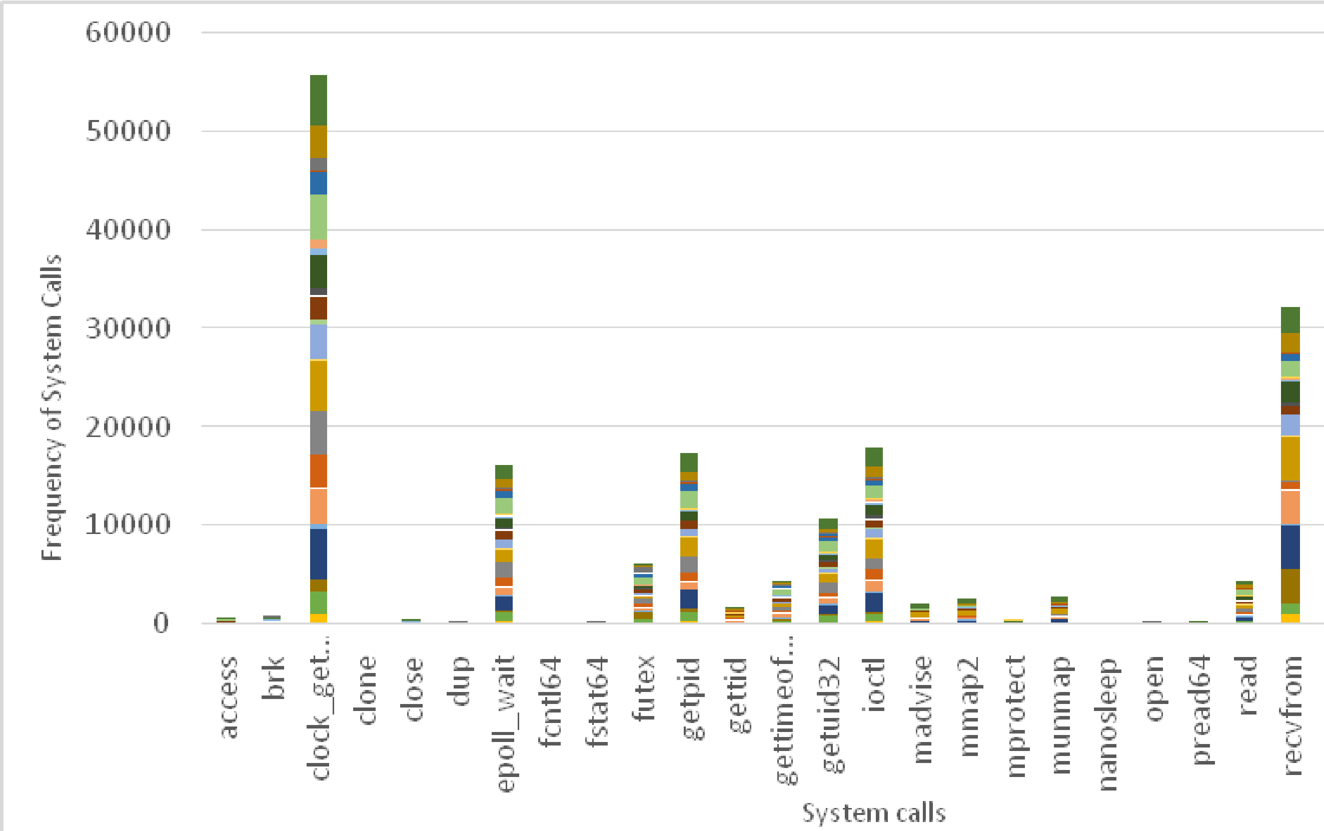


Fig. 17. System calls invoked by Malware Bank applications

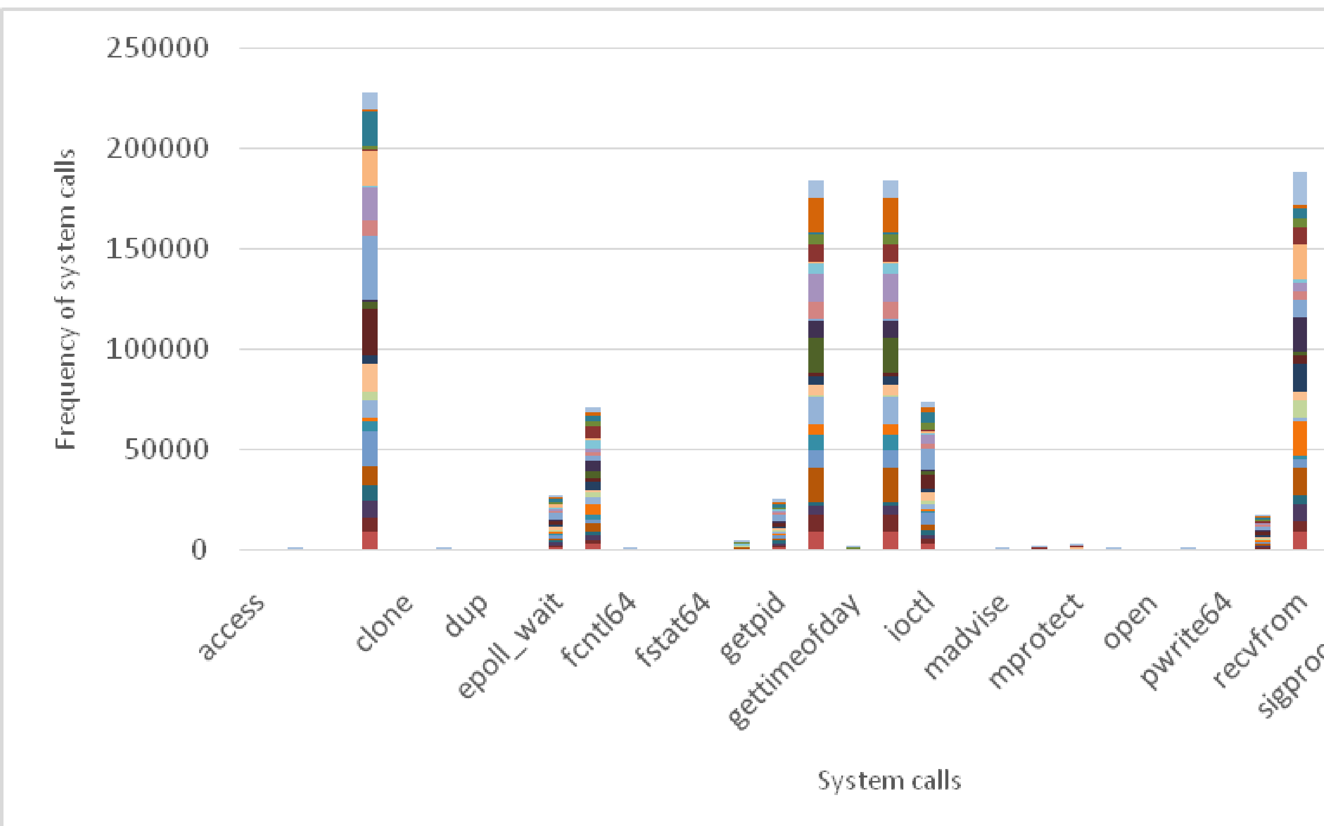


Fig. 18. System calls invoked by MalwareGame applications

Figure 16 and 17 clearly showcases that there is high frequency of certain

Static approach could not detect the unknown malwares so, we have defined an approach usi dynamic analysis data set was created on the basis of frequency of system calls and diffe applied and performance was calculated using machine learning algorithm. Well known data techniques like Naive Bayes, RandomForest,Decision Tree,SVM and KNN were conside analysed and accuracy is calculated.Based on the results,it was concluded that random forest,K proved to be the best classifiers because they achieved statistically valid results. The main feat include: Firstly, usage of system call logs i.e. working at the kernel level to find the malici applications. Secondly, dataset is generated and machine learning algorithms are applied.The c of the dataset is justified with the high accuracy results we obtained.

This study confirms the potential of data mining techniques in prediction of malwares .M category based analysis of applications can further help in better prediction of malwares if ther from the expected behavior of that category.

Our future work will include extending our methodology tohybrid malware analysis in Andro the results with our findings in this research.

VII. REFERENCES

[1] N.Varol, A.F.aydogan and A.Varol, "Cyber attacks Targetting Android Cell-phones," IEEE,2017

[2] Leesha Aneja and Sakshi Babbar,"Research Trends in Malware Detection on Android Devices",springer,2017

[3] A. Feizollah, N. B. Anuar, R. Salleh, G. Suarez-Tangil, and S. Furnell, "Androdialysis: Analysis of android inter detection," Computers & Security, vol. 65, pp. 121–134, 2017, http://www.sciencedirect.com/science/article/pii/ at Publisher · View at Google Scholar

[4] Sapna Malik, Kiran Khatter ,"System call analysis of Android Malware families",Indian Journal of Science 9(21),June 2016.

[5] Saksham Rana , Leesha Aneja, "Static and Dynamic Analysis of Android Malware ,"International Conferen

[6] Aashima Malhotra ,Karan Bajaj ,"A survey on various malware detection techniques on Mobile Platforn Computer Applications (0975-8887),Volume 139-No.5,April 2016.

[7] Balaji Baskaran and Anca Ralescu, "A Study of Android Malware Detection Techniques and Machine Learning"

[8] D.Kapratwar,Ankita,"Static and Dynamic Analysis for Android Malware Detection",San Jose State University,20

[9] Fei Tong,Zheng Yan, "A hybrid approach of mobile malware detectionin android,"J.ParallelDistrib.Comput.(201

[10] Nihar Ranjan Roy, Anshul Kanchan Khanna & Leesha Aneja," Android Phone Forensic: Tools conference,GalgotiasUniversity,Greater Noida,2016.

[11] Ali Feizollah, Nor Badrul Anuar, Rosli Salleh &Ainuddin Wahid AbdulWahab," A review on feature se detection, "Digital Investigation, Elsevier,2015.

[12] Babu Rajesh V, Phaninder Reddy, Himanshu P & Mahesh U Patil," Androinspector:A system for comprehe Applications,"Inter-national Journal of Network security and its Applications(IJNSA), vol. 7,No.5,September 201

[13] Dr. S.Vijayarani1 and Ms. Maria Sylviaa ,"Intrusion Detection System – aStudy", international journal of s Management (IJSPTM) Vol 4, No 1, February 2015.Martina Lindorfer, Matthias Neugschwandtner & Chri

[14] "MARVIN:Efficient and comprehensive Mobile App Classification Through Static and Dynamic analysis," IEEE Computers,software and applications conference,2015.

[15] Pallavi Kaushik, Amit Jain "Malware Detection Techniques in Android" ,International Journal of Computer A ,volume 122 – No.17, July 2015.

[16] P.Mahesh,A.Jayawant,G.Kale ,"Smartphone Security :Review of Attacks,De-tection and Prevention",Internati Research in Com-puter Science and Software Engineering,Volume 5,Issue 3,2015.

[17] S. Y. Yerima, S. Sezer, and I. Muttik, "High accuracy android malware detection using ensemble learning," IET 9, no. 6, pp. 313–320, 2015. View at Publisher · View at Google Scholar · View at Scopus.

[18] Shina Sheen,R.Anitha & V.Natarajan,"Android based malware detection using a multifeature colla approach,"Neurocomputing,Else-vier,2015.

[19] Sapna Malik,Kiran Khatter "AndroData:A tool for static and dynamic feature extraction of Android App",Th Applied Engineering Research, Scopus Indexed,2015.

[20] Walnycky, I. Baggili, A. Marrington and J. Moore, "Network and device forensic analysis of Android soci ,Digital Investigation, Elsevier,vol. 14, pp. S77-S84, 2015.

[21] "The volatility framework: volatile memory artifact," Systems., Volatile, [Online]. Available http://secxplrd.blog

[28] R.Raveendranath,V.Rajamani,A.J.Babu and S.K.Datta,”Android Malware At-tacks and Countermeasu
Directions”IEEE,2014.
[29] Lovi Dua and Divya Bansal,”Review on mobile threats and detection techniques,” International Journal of Distrib
(IJDPS),
[30] M. Kaart and S. Laraghy, "Android forensics: Interpretation of timestamps," Digital Investigation, vol. 11, p. 234
[31] S. Y. Yerima, S. Sezer, and G. McWilliams, “Analysis of bayesian classification-based approaches for androi
Information Security, vol. 8, no. 1, pp. 25–36, 2014. View at Publisher
[32] Nilotpal Chakraborty,”Intrusion Detection System And Intrusion Prevention System: A Comparative Stud
Computing and Busi-ness Research (IJCBR),Volume 4 ,Issue 2, May 2013.
[33] N. Peiravian and X. Zhu, “Machine learning for Android malware detection using permission and API calls,”
IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2013, pp. 300–305, USA, November
[34] Sanz B, Santos I, Laorden C, Ugarte-Pedrero X, Bringas P & lvarez G.,” PUMA:permission usage to detect malv
in intelligent sys-tems and computing, Vol. 189. Berlin Heidelberg: Springer; p. 289e98,2013.
[35] Zhang Y, Yang M, Xu B, Yang Z, Gu G, Ning P et al. ,”Vetting undesirable behaviors in android apps with p
ACM SIGSAC conference on Computer & communications security,p. 611e22,2013.
[36] John Demme, Matthew Maycock, Jared Schmitz & Adrian Tang ,”On the Feasibility of Online Malware D
Counters ,“ISCA ’13 .Kuo-Ping Wu, “DroidMat: Android Malware Detection through Manifest andAPI C
Security (Asia JCIS), pp No.: 62 – 69,2012.
[37] J. Sahs and L. Khan, “A machine learning approach to android malware detection,” in Proceedings of the 2012
Security Informatics Conference, EISIC '12, pp. 141–147, August 2012.
[38] Kavesh Shaerpour,Ali Dehghantanha & Ramlan Mahmod,”Trends in Android Malware Detection Journal of D
and law,vol.8(3).
[39] E. Casey,” Digital Evidence and Computer Crime” in Forensic Science, Computers, and the Internet, Academic P
[40] J. Oh, S. Lee and S. Le, "Advanced evidence collection and Analysis of webBrowseractivity
http://dx.doi.org/10.1016/j.diin.2011.05.008., vol. 8, no. SSN 1742-2876, pp. S62-S70, August 2011.
[41] V. L. Thing, K.-Y. Ng and E.-C. Chang, "Live memory forensics of mobile phones," Digital Investigation, Vo
no. ISSN 1742-2876,http://dx.doi.org/10.1016/j.diin.2010.05.010., pp. S74-S82, August 2010.
[42] William Enck.,Machigar Ongtang,and Patrick Drew ,”Understanding Android Security”,IEEE Security and Priva
[43] R. P. Mislan and T. Wedge, "Designing Laboratories for Small Scale Digital Device Forensics," in ADFS
Forensics, Security and Law, 2008.
[44] L.Aron and P.Hanacek ,”Overview of Security on Mobile Devices,” IEEE,2015
[45] Y.Zhou and X.Jiang,”Dissecting Android Malware:Characterization and Evo-lution”IEEE Symposium on Securit