ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 8, Issue 11, November 2021

# A SVM classifier based approach for malware detection in android using machine learning technique

[1]Sadananda L, [2]Bolwar Aziz Musthafa

[1]Dept. of Computer Science and Engineering, P A College of Engineering, Mangaluru,
[2]P A College of Engineering, Mangaluru, Bearys Institute of Technology, Mangaluru
[1] sadanandaa.l@gmail.com and [2]azizmusthafa@gmail.com

***Abstract*— At present, Android has gotten one of the most notable working frameworks for PDAs by virtue of different versatile applications it bolsters. In any case, the downloaded pernicious Android applications (malware) from pariah markets have inside and out undermined protection and security of the customers. The huge part of malwares remains undetected in light of the nonattendance of powerful and exact malware acknowledgment techniques. In this commitment, examine a SVM based pattern to recognize the malware for Android framework, that fuses both dangerous approval blends and unprotected API's call and used in AI approaches. So as to test the presentation of introduced system, expansive investigations have been sorted out, that exhibited that proposed plan can perceive malicious Android applications suitably and adequately. By utilizing trial confirmation, demonstrate that SVM beats rest of the AI classifiers.***

***Index Terms*— NetworkSecurity, Machine Learning, Malware detection***

## I. INTRODUCTION

Late years, with the quick improvement of mobile phones, Android is ending up being progressively notable. Android has been progressively increasingly basic along its open source properties and good conditions of free in step by step life have a place with us. Regardless, the amount of noxious projects is in like manner growing rapidly. Thusly, how to distinguish the Android spyware with expanded precise rate will be a hot test.

The cell phone has become urgent these years with the creating number of customers of mobile phones tablets and with their growing number of multifaceted nature and limits. Current phones offer an enormous part of the applications and administrations that are executed by PCs. Meanwhile, there is different security risks center around the phones. The standard methodology for malware distinguishing proof reliant on mark is commonly used both have a place with Android contraptions and PCs by isolating the marks from APK and appearing differently in relation to signature which is malignant in the contamination database, in any case, such system is confined to perceive dark malwares that are not accessible in the disease database.

The transcendence of noxious applications in mobile phones can be recognized through different kinds of assessment like static, dynamic and half and half strategies. In static system, we assemble a great deal of applications and remember it for perilous signs. In unique technique, we test for malevolent records while executing on Android framework. At the point when both of these fates are consolidated, named it as half breed technique. In this paper, we receive, the static philosophy for malware ID, and dynamic investigation is talked about. A huge combination of investigating against Android malignant projects has been introduced. At present, static assessment and dynamic examination are the double guideline sorts of acknowledgment techniques. Each system have its advantages and insufficiencies.

The static assessment procedures, for instance, [1], [2], and [3] examine applications with no execution of projects requiring lower overheads. In any case, the strategies can't make preparations for obscurity and unfriendly to decompile. Regardless of what may be normal, unique assessment techniques, for instance, [4] and [5] run the applications continuously to perceive malware, yet it is difficult to get the entirety of the running pathways.

As the malware being rapidly creating, the Machine Learning procedure is used to acquire identification of Android malwares. Accordingly, gathering highlights which speak to in better way, the malignant lead as AI highlights is significant to build the malware discovery's exhibition. The instances of static features consolidate, (i) API calls, (ii) Authorizations that could be isolated from the AndroidManifest.xml record. Dynamic assessment sorts out features which were isolated from the applications when executing, that incorporates (i) organize traffic, (ii) utilization of battery, (iii) IP address, etc. The Android itself has a couple of security parts in its different layered stages. The approval approach used in the application stage is a noteworthy boundary segment to guarantee delicate resources on the Android framework. The Applications should specify dangerous agrees to get the informative data [6, 7]. A couple of assessments are checked malicious android application reliant on the broadcasted consents, uses a approval based procedures[1–3, 8–10].Despite the way that these methodologies keep up a key good ways from high

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 8, Issue 11, Novemebr 2021

overhead and it consider the articulated approvals the features of AI, that can't generally reflect the complexity between great applications and malevolent applications. As such, they can't perceive dangerous applications that report only a couple, or risky authorizations, which are moreover continually declared by chivalrous application.

**Table 1.** : Static Analysis Feature

| SL No. | Features |
|---|---|
| 1 | Meta data |
| 2 | XML Elements |
| 3 | API calls |
| 4 | Native Commands |
| 5 | Opcode from .dxe file |
| 6 | Task Intents |

The most normally used static exercises are the API Calls, and Permissions.

**Table 2.** Dynamic Analysis Feature

| SL. No. | Feature | ML Algorithm |
|---|---|---|
| 1 | System calls collected by strace, Returned values | SVM,Naïve Bayes |
| 2 | Network, sms, power, usage, cpu,process info,native and dalvik memory | Naïve bayes,SVM with SMO algorithm |
| 3 | Process id, system call | KNN,ST |
| 4 | Data collected by logger, internet traffic, battery percentage | NaïveBayes, J48 Decision Trees |
| 5 | Data packets being sent, IP address | Random forest |
| 6 | Network traffic– destination IP address | Classification |

As those are isolated out of Android Manifest.xml application and effect malware disclosure rating to further extent, wide explores have been finished with those as features similarly as joined with various features removed out of meta-data accessible in Google Play-Store, for example, name and number of the rendition, name of the creator, last invigorated time, etc. The table 1 exhibits the significant highlights of static investigation.

The data given in Table 2 consolidates the practically from time to time used features in Dynamic examination and the AI calculations which might be utilized for characterizations. As watched, Network traffic which fuses data packs sent, and rest of the social principles can provoke lively acknowledgment of harmful activity. Following the IP address can help us with getting the topographical scene of the assaulting surface. Additionally, SMS, information logged by Logger is particularly valuable in achieving a higher acknowledgment rate.

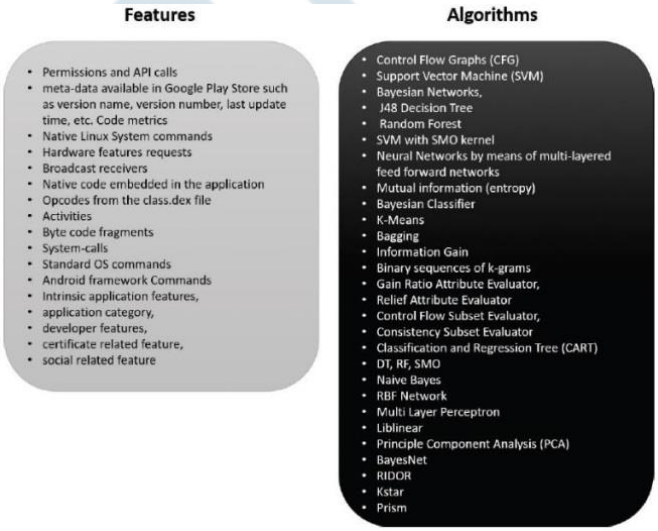


**Fig 1**. Static Analysis Feature and Algorithm

The Fig. 1 shows the highlights of Static examination and calculations which are used for process them to different investigating systems. In the interim, the Fig. 2 shows the highlights of dynamic examination which are used to process them for different exploration strategies, as given in [11]. In static examinations, the features are removed out of the applications report without running the applications. Such way of thinking can be asset and time useful as the applications are not executing. But then, this examination encounters source code disarray process the Malware makers make use to maintain a strategic distance from among static distinguishing proof methods. One of the most notable keeping away from technique is the Updates Attack. The applications are started on the cell phones and keeping in mind that the applications gets the updates, the poisonous substance are got-in from the

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 8, Issue 11, Novemebr 2021

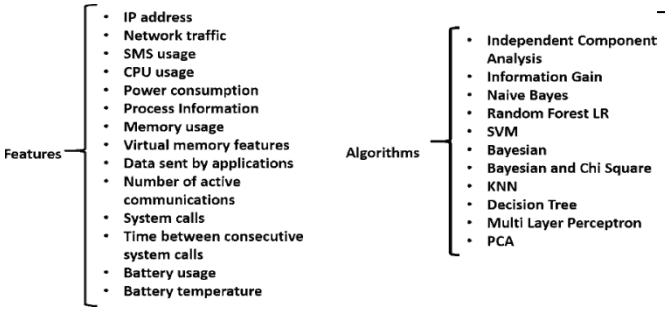


**Fig. 2.** Dynamic Analysis Feature and Algorithm

This can't be recognized by static distinguishing proof methods.

## II. Related Work

The extraordinary arrangement of investigates in the field of recognizable proof of malwares has been done previously. The story dynamic assessment methodology named Component Traversals is suggested that could normally run the source code timetables of each Android application as absolutely as could be normal the situation being what it is. Considering the removed framework calls of Linux, those further build up the weighted composed graphs and a short time later apply a significant learning structure laying on the outline based features for as of late dark Android malware distinguishing proof. In any case, Android applications are run in test system and from such data, structure calls are removed. In such circumstance, few malwares can recognize whether they run on real device or emulator and as necessities be changing the value. Due to which, some malwares can't be perceived from such methodologies

As per the proposition in [12], a component vector is evacuated out of Android Manifest archive, that joins the consent information and the section information of the Android application,together with the orders calculations, for example, Naive Bayes, this approach proposes a poisonous application acknowledgment procedure reliant on Android Manifest record information. This philosophy is a static strategy for malware ID which infers that applications are not executed or researched at execution time for direct assessment. Subsequently, it can't recognize any amateur malware that are prepared for repackaging and disarray to avoid interior methods has a place with them.

The work [13] uses a robotized highlight subordinate static examination system to perceive harmful adaptable applications on Android devices. Such strategy uses metadata of applications and Naïve_Bayes figuring for malware distinguishing proof. The approach is a static technique for malware acknowledgment so it can't shield the device from malwares which may change their selves subject to the ability to interpret, adjust and patching up the code having a place with their selves.

The imprint based strategies [14, 15] introduced in the ninety's, are generally used in malware area. The critical deficiency of such kind of strategies is its weakness in recognizing transformative and hid malware. As opposed to using earlier characterized marks for malware recognizable proof, data mining and AI strategies give an incredible technique to logically isolate malware plans [16].

One more conduct based foot printing strategy [17] additionally gives an amazing method to manage recognize self-multiplying malware. For mobile phone based adaptable preparing framework, continuous years have seen an extending number of progressively obfuscated malware attacks, for instance, repackaging. A progressing assessment introduced in [18] deliberately portrays existing Android malware from alternate points of view, including the foundation techniques, commencement segment similarly as passed on malignant payloads. It is convinced by the growing no. of Apps and the nonattendance of fruitful malware area gadgets, not many investigation [19] endeavor to perceive malware by viewing the estimation and also ground-breaking behavior and characters of applications. The researcher in [20] presents to use agree lead to perceive amateur Android malwares and thereafter make utilizes heuristic filtering for recognizing dark Android malwares.

The work in [21] made partition of the strings in the application, customer rating, check of assessments, size of application, assents and used Bayesian_Networks, Decision_Tree, SVM and Random Forest. A whole of 820+ models were used to check and the makers contemplated that they can achieve a higher exactness with diminished bogus positive proportion. The work proposed in [22], analyzed 796 liberal and 175 pernicious applications for the assessment. Approvals used out of manifest.xml record and API calls information from the classes.dex report are removed and along addition of data they picked a great deal of 20 material API calls. At that point took a gander at the results got by ML computations, for instance, SVM, Naïve_Bayes calculations, and so on.

The work in [23] united the double sorts of used approval, broadcasting authorities and activities, byte code pieces, system calls as features and arranged SVM alongside preparing datasets.

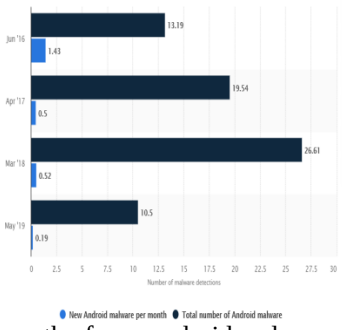


**Fig. 3.** The growth of new android malware worldwide (in millions)

Fig.3 gives the measurable examination of the progression of Android malwares worldwide as on May 2019. Till this month, all together tally of Android malware acknowledgments is gotten together to over 10.5 million tasks.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 8, Issue 11, Novemebr 2021

The creators attempted their proposed Malware distinguishing proof structure along a security analyzer for seat stamping where the system was attempted with 7,000 models. They assume that system could achieve generally 99.3% of positive rate alongside basically 0.14% fake ready proportion.

The Fig. 4 shows that, in second quarter of 2019, the Kaspersky distinguished 753,550 bundles of malignant applications establishments, that is 151,624 lesser contrasted with the first quarter.


**Fig. 4** Graphical representation of searched packages of malicious installation

In connections, our work is prodded by a segment of the above systems and approaches, anyway with base on making clear and convincing malware acknowledgment philosophies, without relying upon complex unique execution time examination and any malware marks which are static and predefined.

## III. Analysis of SVM Classifier based detecting Malware

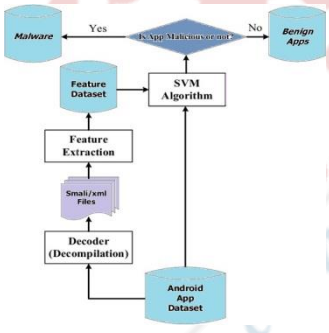


**Fig. 5** The Diagrammatic representation of detection of malware using SVM Classifiers

The Fig. 5 exhibits the general structure for the discovery plan of the malwares. The use of straight SVM (Support Vector Machine) approaches is examined. Later we utilize ML classifiers to exhibit the exploratory strategies and results utilizing SVM algorithm. In the showing of Fig. 5, there are 3 critical portions in the malware acknowledgment contrive, to be explicit decoder used to decompile, extraction of highlights, and classifiers. During the decompile procedure, the Android application unloads and interprets into documents a portion of the key features, for instance, dangerous approvals, URLs and dubious API calls are evacuated in extraction parts according to a couple of critical and comprehensively recognized measures, for instance, cosine similarity and TF-IDF. Finally, we embrace ML computation to evaluate on the Android application dataset by masterminding them into malware or pleasant applications.

**Table 3.** List of malwares to be considered for analysis

| Malware Category | Malware Name | Feature |
|---|---|---|
| Trojan | Zitmo | Disguises as an android security application |
| | DroidKungfu | Leaks personel information |
| | LightDD | Disguises as a adult application |
| | FalseInst | Disguises as game application |
| Spyware | Gemini | Carries out a backdoor function |
| | Snake | Leaks game app information |
| | PjApps | Adds malicious function t normal app |
| Root permission acquisition | RooterBT | Makes terminal rooting |
| | BaseBridge | Acquires root permission |
| Dropper | SMSHider | Guides to install malware through SMS |
| | Ansrever | Install other malware |

In this work, so as to identify the malwares direct SVM procedure is applied. The SVM is one of ML classifiers getting the most thought at this moment, and its various applications are being introduced because of its boss. The SVM could in like manner deal with the issue of collection nonlinear data. From the highlights coming as information, inconsequential ones are emptied by the SVM ML classifier and afterward the demonstrating is done, in this manner there are barely any overheads in the piece of time. In any case, it could be depended upon to acquire preferable outcomes over rest of the classifiers have a place with AI in the piece of precision or intricacy in examination. In this work, we select 14 of the ongoing malware applications for each arrangement to check the proposed method. Vindictive applications are picked dependent on the "normal instances of malware making phenomenal mischief to customers". Most of the Android-concentrated on malwares are separated into Spyware, Trojan, abuse, dropper. The reason behind Trojan having a tremendous degree of the picked malware is in light of the fact that by far most of the toxic codes that occurred in 2012 were Trojan. The Table 3 portrays the malware to be considered for examination.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 8, Issue 11, Novemebr 2021

This work uses all out of 28 including both ordinary applications and malevolent applications embedded with malware to check the identification of malwares. The information assortment is made out of 90% standard and 10% harmful applications. The reason behind making the enlightening assortment consequently is that standard applications are more run of the mill than harmful ones while examining the extent of applications used in the certifiable compact condition. The information is assembled from different gadgets thus way with the goal that the accumulated information is sorted out as the arrangements of preparing and tests.

So as to test and investigate the presentation have a place with the experimentation we present the measurements of assessment in this area. The measurements, for example, True_Positive_Rate(TPR), False_Positive_Rate(FPR), F_Measure, Precision and Accuracy are used. The True_Positive (TP) speaks to the numerical estimation of recognizing the uninfected condition of a run of the mill application. The True_Negative (TN) addresses a tally which precisely recognizes an application having malware. The False_Negative (FN) demonstrates the include which erroneously recognizes malware in an extremely standard application. The False_Positive (FP) shows the tally which wrongly finds no malware despite an application truly having malware. Contingent upon the information assembled, our work gets TPR, FPR, exactness, precision and F_measure utilizing the pointers as given in the underneath conditions. The TPR demonstrates the proportion of properly perceived conventional applications. The FPR demonstrates the apportion of malwares including applications mistakenly perceived as sheltered. The Precision shows the portrayal of a mistake of choice worth, that talks the proportion of properly examined common applications. The precision demonstrates the exactness of the framework, spoke to as the proportion of appropriately perceived conventional applications and ones having malwares, separately, from the results. The F_measure shows the exactness in the piece of choice results.

## IV. Experimentation and Result

The experimentation is finished with the huge measure of information assembled from Android applications including numerous favorable applications and malwares, the considerate cases are accumulated in Google Play stores utilizes the new crawler innovation. We utilized the SVM classifier calculation for building the classifier, up to 20% of the cases as the test informational collection, and up to 80% of the cases as the prepared informational collection. The Table 4 shows the posting of qualities got from the after effects of the malware recognition utilizing the SVM calculation of AI innovation.

**Table 4.** Result of malware detection based on SVM classifiers

| Normal and Malwares | TPR | FPR | Precision | Accuracy | F-Measure |
|---|---|---|---|---|---|
| Normal | 0.999 | 0.004 | 0.992 | 0.997 | 0.995 |
| Adrd.AQ | 0.957 | 0.002 | 0.939 | 0.996 | 0.948 |
| Anserver | 0.957 | 0 | 0.993 | 0.997 | 0.975 |
| Basebridge | 0.939 | 0 | 0.999 | 0.997 | 0.968 |
| DroidKungFu | 0.977 | 0.001 | 0.983 | 0.998 | 0.98 |
| FakeInst | 0.985 | 0.011 | 0.836 | 0.989 | 0.905 |
| Geimini | 0.893 | 0.001 | 0.957 | 0.995 | 0.924 |
| GoldDream | 0.994 | 0.002 | 0.962 | 0.997 | 0.978 |
| LightDD | 0.957 | 0 | 0.998 | 0.998 | 0.977 |
| Opfake | 0.82 | 0.005 | 0.9 | 0.985 | 0.858 |
| PjApps | 0.996 | 0.003 | 0.941 | 0.997 | 0.967 |
| RooterBT | 0.966 | 0.004 | 0.926 | 0.994 | 0.946 |
| SMSHider | 0.949 | 0.001 | 0.976 | 0.996 | 0.962 |
| Snake | 0.935 | 0.001 | 0.977 | 0.995 | 0.956 |
| Zitmo | 0.967 | 0.001 | 0.977 | 0.996 | 0.972 |
| **Average** | **0.953** | **0.002** | **0.957** | **0.995** | **0.954** |

The processed estimations of TPR, Precision, Accuracy, FPR, F-Measures are recorded for 14 malwares alongside the typical estimations of the equivalent. Concurring substance of Table 4. Among the point of view of TPR (0.999), the help vector machine gives a superior exhibition. The metric FPR is used as most critical evaluation marker while distinguishing malware, The SVM accomplishes FPR= 0.004, that can be settled as the better classifier since its extent of wrongly organizing conventional applications as dangerous is nearly nothing, and it shows up far prevalent execution than various classifier similarly to the extent exactness and precision. The malware GoldDream has recorded most noteworthy TPR=0.994 among all other 14 malwares. The RooterBT recorded FPR as 0.004 which is identical to the FPR of ordinary App. The malware Basebridge records accuracy (0.999) to the most extreme. The malwares Anserver, Basebridge, GoldDream and PjApps shares the most extreme precision (0.997). The GoldDream has greatest F-measure by recording to 0.978.

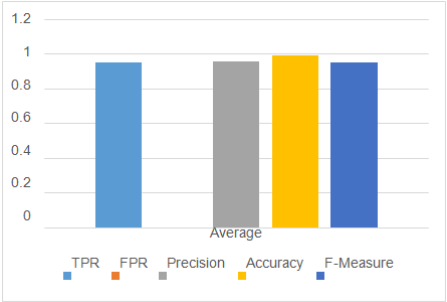


**Fig. 6** Graphical representation of average value of evaluation metrics

The Average qualities accomplished for all the assessment measurements are graphically dissected in Fig.6 The normal incentive for TPR is 0.953, the normal estimation of FPR is 0.002. Essentially, the normal estimations of Precision factor, Accuracy contents and F-measure are recorded as 0.957, 0.995 and 0.954 separately. Therefore it exhibits that, in examination with ordinary processed qualities the SVM shows the best exhibitions in recognition of the malwares.

## V. Conclusion

The Average qualities accomplished for all the assessment measurements are graphically dissected in Fig.6 The normal incentive for TPR is 0.953, the normal estimation of FPR is 0.002. Essentially, the normal estimations of Precision factor, Accuracy contents and F-measures are recorded as 0.957, 0.995 and 0.954 separately. Therefore it exhibits that, in examination with ordinary processed qualities the SVM produces the best exhibitions in recognition of malwares.

## REFERENCES

[1] W. Enck, M. Ongtang, and P. Mcdaniel, "On lightweight mobile phone application certification," in Proceedings of the 16th ACM Conference on Computer and Communications Security, pp. 235– 245, Chicago, USA, November 2009

[2] H. Yang, Y. Zhang, Y. Hu et al., "Android malware detection method based on permission sequential pattern mining algorithm," Journal on Communications, vol. 34, pp. 106–115, 2013C. A. Castillo, Android Malware Past, Present, and Future, Tech. rep., Mobile Working Security Group McAfee (2012).

[3] D. Arp, M. Spreitzenbarth, M. Hubner et al., "Drebin: effective and explainable detection of android malware in your pocket," in Proceedings of the 2014 Network and Distributed System Security Symposium (NDSS), pp. 23–26, San Diego, USA, February 2014 Android and security: Official mobile google blog, (Online; Last Accessed 15th October 2013).

[4] W. Enck, P. Gilbert, S. Han et al., "TaintDroid: an information- flow tracking system for real time privacy monitoring on smartphones," ACM Transactions on Computer Systems, vol. 32, no. 2, pp. 1–29, 2014

[5] Y. Zhang, M. Yang, B. Xu et al., "Vetting undesirable behaviors in android apps with permission use analysis," in Proceedings of the2013 ACM SIGSAC Conference on Computer & Communications Security (CCS), pp. 611–622, Berlin, Germany, November,2013AndroidHipposms, (Online; 2011).

[6] S. Kumar, R. Shanker, and S. Verma, "Context aware dynamic permission model: a retrospect of privacy and security in android system," in Proceedings of the International Conference on Intelligent Circuits and Systems (ICICS), pp. 324– 329, Phagwara, India, April 2018

[7] S. Rosen, Z. Qian, and Z. M. Mao, "AppProfiler: a flexible method of exposing privacy-related behavior in android applications to end users," in Proceedings of the third ACM Conference +on Data and Application Security and Privacy—CODASPY '13, pp. 221–232, San Antonio, USA, February 2013

[8] O. Yildiz and I. A. Do˘gru, "Permission-based android malware detection system using feature selection with genetic algorithm," International Journal of Software Engineering and Knowledge Engineering, vol. 29, no. 2, pp. 245–262, 2019.

[9] R. S. Arslan, ˙I. A. Do˘gru, N. Baris¸çi, and N. Baris¸çi, "Permission-based malware detection system for android using machine learning techniques," International Journal of Software Engineering and Knowledge Engineering, vol. 29, no. 1, pp. 43–61, 2019

[10] W. Wang, X. Wang, F. Dawai, J. Liu, Z. Han, and X. Zhang, "Exploring permission-induced risk in android applications for malicious application detection," IEEE Transactions on Information Forensics and Security, vol. 9, no. 11, pp. 1869– 1882, 2014

[11] Baskaran B,Ralescu A. A study of android malware detection techniques and machine learning .In : Phung PH, Shen Jglass M, editors. Modern artificial intelligence and cognitive science:22-23 April 2016.Dayton, OH, USA:CEUR ;2016,. 15-23.

[12] X. Li, J. Liu, Y. Huo, R. Zhang, Y. Yao, 'An Android malware detection method based on Android Manifest file', International Conference on Cloud Computing and Intelligence Systems (CCIS), 2016, pp. 239-243

[13] N. B. Akhuseyinoglu, K. Akhuseyinoglu, 'AntiWare: An automated Android malware detection tool based on machine learning approach and official market metadata', IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), 2016, pp. 1-7.

[14] Shabtai, A., et al., "Andromaly": a behavioral malware detection framework for android devices. J. Intell. Inf. Syst., 2012. 38(1): p. 161-190.

[15] Arnold, J.O.K.W.C., Automatic Extraction of Computer Virus Signatures. In Proceedings of 4th Virus Bulletin International Conference, 1994: p. 178-184

[16] M.G. Schultz, E.E., F. Zadok, S.J. Stolfo, Data mining methods for detection of new malicious executables. Proceedings 2001 IEEE Symposium on Security and Privacy. S&P 2001, 2001: p. 38-49.

[17] Zhu, X.J.X., vEye: behavioral footprinting for self-propagating worm detection and profiling. Knowledge and Information Systems, 2009. 18(2): p. 231-262

[18] Jiang, Y.Z.X., Dissecting Android Malware: Characterization and Evolution. IEEE Symposium on Security and Privacy, 2012: p. 95-109.

[19] Burguera, I., U. Zurutuza, and S. Nadjm-Tehrani,Crowdroid:behavior-based malwaredetection system for Android, in Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices. 2011, ACM: Chicago, Illinois, USA. p. 15-26.

[20] Yajin Zhou, Z.W., Wu Zhou, Xuxian Jiang, Hey, You, Get Off of My Market: Detecting Malicious Apps in Official and Alternative Android Markets. network and distributed system security symposium, 2012.

[21] B. Sanz, I. Santos, C. Laorden, X. Ugarte- Pedrero, and P.G. Bringas. On the automatic ategorisation of android applications. In Consumer Communications and Networking Conference (CCNC), 2012 IEEE, pages 149–153, Jan 2012.

[22] P.P.K. Chan and Wen-Kai Song. Static detection of android malware by using permissions and api calls. In Machine Learning and Cybernetics (ICMLC), 2014 International Conference on, volume 1, pages 82–87, July 2014.

[23] Ideses and A. Neuberger. Adware detection and privacy control in mobile devices. In Electrical Electronics Engineers in Israel (IEEEI), 2014 IEEE 28th Convention of, pages 1–5, Dec 2014.