Various Data Mining Techniques to Detect the Android Malware Applications: A Case Study

Rincy Raphael

***Abstract* - Android has become the most popular smartphone operating system. This rapidly increasing adoption of Android has resulted in significant increase in the number of malwares when compared with previous years. There exist lots of antimalware programs which are designed to effectively protect the users' sensitive data in mobile systems from such attack. But the attacking rate is increasing year by year. In this paper we conduct a survey of various datamining techniques conducted to analyse and detect the android malware applications. We also analysing the classification algorithm used, dataset size and accuracy of the system.**

***Index Terms* – Android Malware, Data mining, Classifiers, Mobile Application**

I. INTRODUCTION

Android is one of the interesting platforms for Smartphone users and number of users are increasing every year. In 2019 beginning the number of smartphone used are reached 2.7 million and the study shows that it will reach 2.87 million in 2020 (refer Figure 1). Smartphones provide different connectivity options such as Wi-Fi, GSM, GPS, CDMA and Bluetooth etc. which make them a ubiquitous device.

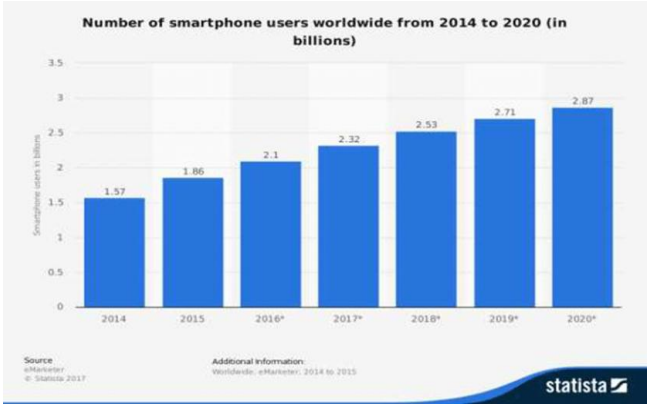

Figure1: Number of smartphone users worldwide from 2014-2020 [1]

Android operating system left its competitors far behind by capturing more than 74.85% of total mark It could be observed that Android has become the most widely used operating system over the years (refer Figure 2). Android platform offers sophisticated functionalities at very low cost and has become the most popular operating system for handheld devices. Apart from the Android popularity, it has become the main target and attraction for attackers and malware developers. Android apps available in the official Android market as well as the third party android market where no security is provided to control the attack of embedding malicious content into applications. The millions of applications that are being downloaded by the users in a large number everyday [46] and the markets are not providing any security. Attackers use dynamic execution, stealth techniques, code obfuscation methods, encryption and repackaging to bypass the existing antimalware techniques provided by Android platform.

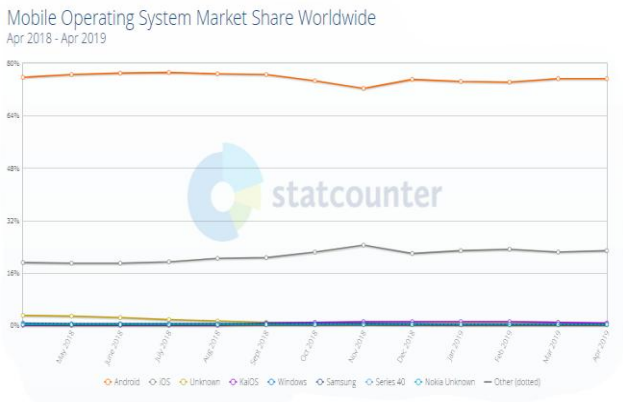

Figure 2: Mobile Operating System Market share worldwide 2019 [2]

II. ANDROID MALWARE ANALYSIS

Wide range of malwares has been detected and the number of malwares is increasing every year. In 2018, Kaspersky Lab products and technologies detected 5,321,142 malicious mobile installation packages, which is down 409,774 on last year [3]. The behavior of different malware families is given below.

a. TROJANS

Trojans appear to a user as a Benign app [4]. In fact, they actually steal the user's confidential information without the user's knowledge. Such apps can easily get access to the browsing history, messages, contacts and device IMEI numbers etc. of victim's device and steal this information without the consent of user.

b. BACKDOORS

Backdoors employ the root exploits to grant root privileges to the malwares and facilitate them to hide from antiviruses. Exploid, Rage against the cage (RATC) and Zimperlich are the top three root exploits which gain full- control of device [5]. If the root exploit succeed to gain control over device and root privilege, the malware become able to perform any operation on the device even the installation of applications keeping the user unaware of this act [6].

Rincy Raphael, Department of Computer Science & Engineering, Anna University, Chennai, India

c. WORMS

Such malwares create copies of it and distribute them over the network. For example, Bluetooth worms spread malware through the Bluetooth network by sending copies of it to the paired devices.

d. Botnets

Botnet is a network of compromised Android devices. Bot master, a remote server, controls the botnet through the C&C network. Geinimi [7] is one of the Android botnets.

e. RANSOMWARES

Ransomware prevent the user from accessing their data on device by locking the device, until ransom amount is paid. It locks the victim's device and force the user to pay ransom amount to unlock the device.

f. RISKWARES

Riskwares are the legitimate software exploited by the malicious authors to reduce the performance of device or harm the data e.g., delete, copy or modify etc. [8].

III. MALWARE DETECTION APPROACHES

The machine learning techniques are divided into supervised and unsupervised. Malware detection approaches are divided into two main categories that include behaviour based and signature based methods [16]. Malware Analysis is performed in two ways such as static and dynamic [18].

A. Signature Based Malware Detection

Recently, signature-based detection is the most generally utilized procedure in antivirus programming highlighting exact correlation. Malware recognition has essentially centered on performing static investigations to review the code-structure mark of infections, instead of element behavioral methods [19]. The signature-based system finds interruptions utilizing a predefined list of known assaults. Despite the fact that this arrangement has the ability to identify malware in the versatile application, it requires steady overhauling of the predefined signature database. Moreover, it is less effective in identifying noxious exercises utilizing the signature-based technique because of the quickly changing nature of portable malware [20, 21]. The main advantage of signature-based techniques is their thoroughness since they follow all conceivable execution ways of a given document.

B. Behavior Based Malware Detection

Behavior-based methodologies require execution of a given example in a sandboxed situation and run-time exercises are checked and logged. Dynamic investigation systems utilize both virtualization and imitating conditions to execute a malware and to remove its practices. The primary advantage of the behavior-based approach is that gives a superior comprehension of how malware is produced and implemented [10, 13]. In the behavior-based malware approach, the suspicious objects are assessed based on their activities that they cannot execute in system. Efforts to achieve activities that are clearly irregular or unofficial would specify the suspicious object is malicious, or at least apprehensive. A malicious behavior is known using a dynamic analysis that evaluates malicious intent by the object's code and structure**.**

IV. REVIEW OF THE MALWARE DETECTION APPROACHES

In this section, the existing malware detection approaches are analyzed according to some evaluation factors such as the method, concept, classification techniques, method of data analysis, dataset size and accuracy. We analyze the selected studies according to existing approaches and discuss on them.

A. Signature Based Approaches

Cui et al. [24] illustrated a novel recognition framework in light of cloud environment and packet examination. The framework identifies the malicious mobile malware behavior through their bundles with the utilization of information mining strategies. This approach totally keeps away from the deformities of customary techniques. The framework is administration arranged and can be sent by portable administrators to send cautions to clients who have malware on their gadgets. To enhance framework execution, another bunching technique called withdrawal grouping was made. This technique utilizes earlier learning to lessen dataset measure. In addition, a multi-module location plan was acquainted with improve framework precision. The aftereffects of this plan are created by incorporating the location consequences of a few calculations, including Naive Bayes and Decision Tree.

Wu et al. [22] have utilized an artificial immune-based smartphone malware detection model (SP-MDM) both static malware examination and element malware investigation as indicated by the component of the biologic resistant framework that can shield us from disease by creatures. In this model, the static marks and dynamic marks of malware are separated, and in view of the genuine esteemed vector encoding, the antigens are produced. The youthful identifier develops into a develop one on the off chance that it experiences self-resistance. Finder posterity with higher fondness is made after the streamlining of developing identifiers utilizing clonal determination calculation. Also, they collected twenty malware and twenty benign files as testing samples set.

Fan et al. [25] proposed a compelling arrangement mining calculation to find vindictive quintal examples, and afterward, All-Nearest-Neighbor (ANN) classifier is constructed for malicious position in the established samples. The created information mining structure made out of the proposed consecutive example mining technique and ANN classifier can well describe the malevolent examples from the gathered record test set to adequately distinguish recently concealed malware tests. A thorough exploratory review on a genuine information accumulation is performed to assess our recognition structure. The promising test comes about demonstrate that their structure beats other to exchange information mining based discovery techniques in distinguishing new vindictive executable.

Bat-Erdene et al. [23] presented a strategy for characterizing the packing algorithms of given unknown packed executable. To begin with, they measured the entropy estimations of a

given executable and change over the entropy estimations of a specific area of memory into typical representations. Their presented strategy utilized symbolic aggregate approximation (SAX), which is known to be viable for huge information changes. Second, we order the conveyance of images utilizing managed learning order strategies, i.e., credulous Bayes and bolster vector machines for recognizing pressing calculations. The aftereffects of our examinations including a gathering of 324 pressed kindhearted projects and 326 stuffed malware programs with 19 pressing calculations illustrate that our strategy can distinguish pressing calculations of given executable with a high precision of 95.35%, a review of 95.83%, and an accuracy of 94.13%. We propose four likeness estimations for distinguishing pressing calculations based on SAX representations of the entropy values and an incremental total examination. Among these four measurements, the loyalty closeness estimation shows the best-matching result, i.e., a rate of precision running from 95.0 to 99.9%, which is from 2 to 13 higher than that of the other three measurements. Our review affirms that pressing calculations can be recognized through an entropy examination in view of a measure of the instability of the running procedures and without earlier information of the executable.

Wang and Wang [20] presented a malware recognition framework to ensure a little order mistake by machine learning using the speculation capacity of support vector models (SVMs). This review built up a programmed malware location framework via preparing a SVM classifier in light of behavioral marks. Over approval, plan was utilized for taking care of grouping exactness issues by utilizing SVMs connected with 60 groups of genuine malware. The trial comes about uncover that the characterization blunder diminishes as the measuring of testing information is expanded. For various estimating (N) of malware tests, the expectation precision of malware discovery runs up to 98.7% with N = 100. The general recognition precision of the SVC is more than 85% for unspecific versatile malware.

Santos et al. [27] proposed another strategy to identify obscure malware families. This model depends on the recurrence of the presence of opcode groupings. Moreover, they depicted a system to mine the importance of each opcode and evaluate the recurrence of each opcode grouping. Furthermore, they provided experimental approval that this new strategy is fit for recognizing obscure malware.

B. Behavior Based Approaches

Yuan et al. [31] presented a deep learning method to connect the components from the static investigation with elements from the dynamic investigation of Android applications. In addition, they actualized an Android malware detection engine based on the deep-learning method (Droid Detector) that can consequently distinguish whether a file has a malicious behavior or not. With a large number of Android applications, they tested Droid Detector and play out an in-depth examination of the elements that deep learning basically adventures to portray malware completely. The outcomes appear that deep learning is appropriate for characterizing Android malware and particularly compelling with the accessibility of additional preparation information. Droid Detector can accomplish 96.76% detection accuracy, which traditional machine learning methods.

Mohaisen et al. [30] proposed, a computerized and conduct based malware examination and marking framework called AMAL that addresses shortcomings of the current frameworks. AMAL comprises of two sub-frameworks, AutoMal and MaLabel. AutoMal gives instruments to gather low granularity behavioral curios that portray malware utilization of the document framework, memory, organize, what's more, registry, and does that by running malware tests in virtualized situations. On the other hand, MaLabel utilizes those ancient rarities to make delegate highlights, utilize them for building classifiers prepared by physically screened preparing tests, and utilize those classifiers to characterize malware tests into families comparable in conduct. AutoMal additionally empowers unsupervised learning, by executing various bunching calculations for tests gathering. An assessment of both AutoMal and MaLabel in view of medium-scale (4000 specimens) and expansive scale datasets (more than 115,000 samples) collected and broke down via AutoMal shows AMAL's adequacy in precisely describing, ordering, and gathering malware tests. MaLabel accomplishes an exactness of 99.5% and review of 99.6% to confident relations demand, and more than 98% of accuracy and evaluation for unsupervised classification.

Eskandari et al. [34] presented a novel hybrid approach, HDM-Analyzer, is displayed which takes points of interest of dynamic and static investigation techniques for rising pace while protecting the precision at a sensible level. HDM-Analyzer can foresee the dominant part of basic leadership focuses on using the factual data which is assembled by element investigation; along these lines, they have no any performance overhead. The fundamental commitment of this paper is taking exactness preferred standpoint of the element investigation and consolidating it into static examination keeping in mind the end goal to enlarge the precision of static investigation. Truth be told, the execution overhead has been endured in learning stage; hence, it does not force on highlight extraction stage which is performed in examining operation. The exploratory outcomes illustrate that HDM-Analyzer accomplishes better general exactness and time many sided quality than static and element investigation strategies.

Boukhtouta et al. [32] presented the issue of fingerprinting perniciousness of activity with the end goal of recognition and arrangement. This research pointed first at fingerprinting perniciousness by utilizing two approaches: Deep Packet Inspection (DPI) and IP bundle headers arrangement. To this end, we consider malignant activity created from element malware examination as movement perniciousness ground truth. In light of this supposition, they exhibited how these two methodologies are utilized to recognize what's more, attribute maliciousness to the various threat. In this work, we concentrate the positive and negative angles for Deep Packet Review and IP bundle headers order. They assessed every approach in view of its recognition and attribution precision and additionally their level of multifaceted nature. The results of both methodologies have demonstrated promising outcomes as far as discovery; they are great possibility to constitute a collaboration to expand or prove recognition frameworks as far as runtime speed and grouping exactness.

Ming et al. [35] have presented a substitution attacks to cover comparable practices by harming behavior-based specifications. The key strategy for the attacks is to supplant a

system call dependence graph to its semantically identical variations so that the comparable malware tests confidential unique family end up being characteristic. Accordingly, malware investigators need to put more endeavors into reconsidering the similar samples which may have been examined sometime recently. They distill general attacking strategies by mining more than 5200 malware tests' behavior specifications and execute a compiler-level model to automate replacement attacks. By evaluating on the real malicious examples, the effectiveness of the proposed method to obstruct several behavior based malware analysis tasks, such as clustering and malware comparison. Finally, they discussed likely countermeasures to support current malware protection.

Ding et al. [33] proposed an affiliation mining strategy based on API calls to recognize malware. To expand the identification speed of the Objective-Oriented association (OOA) mining, distinctive methodologies are exhibited: to enhance the govern quality, criteria for API determination are proposed to expel APIs that can't get to distinctly visit things; to discover affiliation decides that have solid segregation control, we characterize the manage utility to assess the affiliation runs; and to enhance the location exactness, a characterization strategy in view of numerous affiliation guidelines is embraced. The trials demonstrate that the proposed systems can essentially enhance the running velocity of OOA. In our investigations, the time cost for information mining is decreased by 32%, and the time cost for arrangement is decreased by 50%.

Norouzi et al. [39] have proposed distinctive classification techniques with a specific end goal to recognize malware in light of the element and conduct of each malware. A dynamic investigation technique has been exhibited for recognizing the malware features. A recommended program has been introduced for changing over a malware behavior executive history XML document to an appropriate WEKA instrument input. To represent the execution proficiency and preparing information and test, the authors apply the proposed ways to deal with a genuine contextual investigation information set utilizing WEKA instrument. The evaluation results described that the availability of the proposed data mining approach. In addition, their proposed data mining methodology is more proficient for identifying malware and behavioral classification of malware can be helpful to recognize malware in a behavioral antivirus.

Galal et al. [40] proposed a behavior-based features model that defines malicious action exhibited by malware example. To remove the proposed model, the authors first perform dynamic examination on a generally late malware dataset inside a controlled virtual environment and capture traces of API calls conjured by malware examples. The traces are then generalized into high-level features refer to as actions. The proposed method is evaluated using some famous classification methods such as random forests, decision tree and SVM. The experimental results show that the classifiers attain high precision and satisfactory results in the detection of malware variants.

Miao et al. [35] presented a bilayer conduct reflection strategy in light of the semantic examination of dynamic API sequences. Operations on touchy framework assets and complex practices are disconnected in an interpretable way at various semantic layers. At the lower layer, crude API calls are joined to extract low-layer practices by means of information reliance investigation. At the higher layer, low-layer practices are further joined to build more intricate high-layer practices with great interpretability. The separated low-layer furthermore, high-layer practices are at last inserted into a high dimensional vector space. Henceforth, the disconnected practices can be specifically utilized by numerous prominent machine learning calculations. In addition, to handle the issue that considerate projects are not satisfactorily examined or malware and amiable projects are seriously imbalanced, an enhanced one-class bolster vector machine (OCSVM) named OC-SVM-Neg is proposed which makes utilization of the accessible negative examples. The trial comes about demonstrate that the proposed include extraction technique with OC-SVM-Neg beats double classifiers on the false caution rate and the speculation capacity.

Nikolopoulos and Polenakis [37] have proposed a graph-based model which using relations between gatherings of system-calls, distinguishes whether an unknown software sample is malicious or benign, and classifies a malevolent software to one of a set of an arrangement of known malware families. All the more correctly, clients used the System-call Dependency Graphs (or, for short, ScD-graphs), acquired by traces captured through dynamic taint investigation. The authors planed their model to be safe against strong changes applying our recognition and arrangement systems on a weighted coordinated graph, to be specific Group Relation Graph, or Gr-graph for short, coming about because of ScD-graph subsequent to gathering disjoint subsets of its vertices. For the discovery procedure, the authors proposed the Delta-comparability metric, and for the procedure of classification, they proposed the SaMe-similitude and NP-similarity measurements comprising the SaMe-NP closeness. At last, they evaluated their model for malware recognition and classification demonstrating its possibilities against malicious software measuring its identification rates and classification accuracy.

Sheen et al. [38] have considered Android-based malware for examination and an adaptable recognition component is planned to utilize multi-feature collaborative decision fusion (MCDF). The distinctive features of a malicious record like the consent-based features and the API call based features are considered keeping in mind the end goal to give a superior discovery via preparing a gathering of classifiers and combining their choices utilizing collective approach in view of likelihood hypothesis. The execution of the proposed model is evaluated on a gathering of Android-based malware including diverse malware families and the outcomes demonstrate that the presented approach give a superior execution than best in class troupe plans accessible.

V. Conclusion

This paper presented a literature review of different data mining techniques for android malware detection. Signature and behavior based approaches are investigated in the paper. The paper also reviewed the various classification algorithms, data analysis methods, dataset size and Accuracy of the proposed work. The DPIM approach gives the maximum accuracy as 99.6% and 86%is the minimum accuracy for the DMDAM method (refer Table 1 & 2). From the experiments we observed that SVM classification algorithm gives highest

malware detection rate as 29%. J48 and Decision Tree gives accuracy as 17% and 14% respectively. Other Classification techniques provide less that 10%. Finally, we have seen that 30% of the signature-based and 65% of the behavior-based malware detection approaches have used the dynamic data analysis method. To protect the devices, the antivirus software is developed in world wide. But in the study reveals that the static approach is less efficient in detecting the malicious contents that are loaded dynamically from remote servers. The dynamic approach is efficient but they time consuming processes. Finally we can conclude that whether we using signature-based or behavior-based techniques, the hybrid approach with SVM classifier will address the limitations of existing static and dynamic approaches.

Table 1: Review of selected Signature based Approaches

Method	Concept	Classification Techniques	Method of Data Analysis	Dataset Size	Accuracy (%)
APMD	API malware detection (APMD) [19]	Naive Bayes and Decision tree, SVM	Dynamic	7000	95
BAM	Hybrid malware detection with binary associative memory [12]	MLP, SVM, Naïve Bayes, J48	Hybrid	52,183	98.6
DBScan	Hybrid pattern based text mining approach [45]	ANN, malicious sequential Pattern based Malware Detection	Hybrid	8000	98.89
Droid	Droid malware detection [43]	SVM	Dynamic	7000	98
DroidNative	Android malware detector with control flow patterns [37]	Droid, CFGO-IL	Static	3158	93.57
FPM	Frequent pattern mining (FPM) [33]	Minimal contrast frequent subgraphs	Static	2083	92
MKLDroid	A multi-view context-aware approach to Android malware detection [44]	Multiple Kernel Learning, SVM	Static	6056	98.05
MobA	Mobile android [20]	SVM	Hybrid	2500	98.7
MOED	Multi-objective evolutionary detection (MOED) [26]	Multi-objective evolutionary by GA	Static	9383	95.15
OpCode	Graph malware detection [3]	Graph-SVM	Dynamic	6671	88
Opcode	Opcode sequences [27]	K-nearest neighbors and SVM	Hybrid	2000	92.9
PMD	Polymorphic Malware Detection (PMD) [21]	K-means	Dynamic	2876	99
SAAM	Symbolic aggregate approximation for mal-wares (SAAM) [23]	Naive Bayes and SVM	Dynamic	8100	95.83
SHMD	Signature and Heuristic-based malware detection [28]	SVM, J48, KNN, Decision Tree and Random Tree	Hybrid	500	99.81
SigPID	Significant permission identification android malware detection (SigPID) [15]	SVM	Dynamic	5494	94
SMD	Smartphone malware detection (SMD) [22]	K-means artificial immune system	Hybrid	1300	89.8
SOMM	Service-Oriented mobile malware detection (SoMM) [24]	Naive Bayes and Decision Tree	Hybrid	3000	97.3

SPM	Sequential pattern mining (SMP) [25]	All-Nearest-Neighbor, KNN, SVM J48	Hybrid	3200	95.2
SVDD	N-grams malware detection [20]	SVM	Dynamic	658	97

Table 2: Review of selected Behavior based Approaches

Method	Concept	Classification Techniques	Method of Data Analysis	Dataset Size	Accuracy (%)
ABM	Android based malware [38]	J48, SVM, IBk, Naïve Bayes	Static	2000	98.91
AMAL	AMAL: automated malware analysis [30]	Decision trees	Dynamic	2086	98
AMCS	Android Malware Characterization and Detection [31]	Deep belief networks	Hybrid	1860	96.76
AMD	Android malware detection [38]	Evolving neuro fuzzy inference system	Dynamic	500	90
AMP	Android malware detection [18]	Multilayer perceptron	Dynamic	734	97
BBA	Bilayer behavior abstraction [35]	SMV, Naïve Bayes, Decision tree, Logistic regression	Dynamic	17,000	94
CloudIntell	Feature extraction method in cloud [14]	Decision tree, SVM and Boosting	Static	15000	99.5
DBM	Behavioral malware [39]	Regression, SVM, J48	Dynamic	7000	98.3
DeepAM	Deep learning malware detection [11]	DeepAM	Dynamic	2000	98
DeepFlow	Deep-learning malware detection [42]	Naive Bayes, PART, Logistic Regression, SVM and MLP	Hybrid	11000	95.05
DFAMD	Data flow android malware detection [51]	KNN, LR, BN	Static	2200	97.66
DMDAM	Android malware detection [9]	Random forest	Dynamic	170	86
DPIM	Deep Packet Inspection for malware [32]	BoostedJ48, J48, Naïve Bayesian and SVM	Dynamic	4560	99.6
HAM	Hybrid analysis malware [34]	Bayesian network, Naïve Bayes, Layy K0 Stare	Hybrid	3000	95.27
MAPI	Malicious code based on API [40]	Decision tree, SVM and Random Forest	Dynamic	2000	96.89
Mspec	Malware specifications [36]	System call dependency graph	Dynamic	5200	92
OOM	Objective Oriented malware [33]	Multiple association rule	Hybrid	8000	97.2
QDFG	Graph mining in malware detection [17]	Graph search	Dynamic	6994	96
SCCMD	So-called compression based malware detection [17]	k-NN, QDA, LDA, SVN, Decision Trees and Random Forest	Dynamic	7507	99.3

SDMS	Security dependency network for malware detection [41]	No read down and no write up	Dymanic	7257	93.92
SyCM	System-call malware [37]	SaMe-NP	Dynamic	2667	95.9

REFERENCES

[1]. https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide, last access March 2019

[2]. http://gs.statcounter.com/os-market-share/mobile/worldwide, last access April 2019

[3]. https://securelist.com/mobile-malware-evolution-2018/89689/, last access December 2018

[4]. "Android and Security - Official Google Mobile Blog." [Online]. Available: http://googlemobile.blogspot.in/2012/02/android-andsecurity.html. [Accessed: 28-Oct-2015].

[5]. "Root Exploits." [Online]. Available: http://www.selinuxproject.org/~jmorris/lss2011_slides/caseforse android. pdf. [Accessed: 15-Dec-2018].

[6]. Y. Zhou, Z. Wang, W. Zhou, and X. Jiang, "Hey, You, Get Off of My Market: Detecting Malicious Apps in Official and Alternative Android Markets," Proc. 19th Annu. Netw. Distrib. Syst. Secur. Symp., no. 2, pp. 5–8, 2012.

[7]. Y. Zhou and X. Jiang, "Dissecting Android Malware: Characterization and Evolution," 2012 IEEE Symp. Secur. Priv., no. 4, pp. 95–109, 2012.

[8]. "Riskware | Internet Security Threats." [Online]. Available: http://usa.kaspersky.com/internet-security-center/threats/riskware#.Vm5IUp97IU. [Accessed: 15-Dec-2018].

[9]. Bhattacharya A, Goswami RT (2017) DMDAM: data mining based detection of android malware. In: Mandal JK, Satapathy SC, Sanyal MK, Bhateja V (eds) Proceedings of the first international conference on intelligent computing and communication springer Singapore, Singapore, pp 187–194.

[10]. Pektaş A, Acarman T (2017) Classification of malware families based on runtime behaviors. J Inf Secur Appl 37:91–100. https://doi.org/10.1016/j.jisa.2017.10.005.

[11]. Ye Y, Chen L, Hou S, Hardy W, Li X (2017) DeepAM: a heterogeneous deep learning framework for intelligent malware detection. Knowl Inf Syst. https://doi.org/10.1007/s10115-017-1058-9.

[12]. Chowdhury M, Rahman A, Islam R (2018) Malware analysis and detection using data mining and machine learning classification. In: Abawajy J, Choo K-KR, Islam R (eds) International conference on applications and techniques in cyber security and intelligence: applications and techniques in cyber security and intelligence. Springer International Publishing, Cham, pp 266–274

[13]. Palumbo P, Sayfullina L, Komashinskiy D, Eirola E, Karhunen J (2017) A pragmatic android malware detection procedure. Comput Secur 70:689–701. https://doi.org/10.1016/j.cose.2017.07.013.

[14]. Siddiqui M, Wang MC, Lee J (2008) A survey of data mining techniques for malware detection using file features. In: Proceedings of the 46th annual southeast regional conference on xx. 2008. ACM

[15]. Sun L, Li Z, Yan Q, Srisa-an W, Pan Y (2016) SigPID: significant permission identification for android malware detection. In: 2016 11th international conference on malicious and unwanted software (MALWARE), pp 1–8

[16]. Boujnouni ME, Jedra M, Zahid N (2015) New malware detection framework based on N-grams and support vector domain description. In: 2015 11th international conference on information assurance and security (IAS), pp 123–128

[17]. Wuechner T, Cislak A, Ochoa M, Pretschner A (2017) Leveraging compression-based graph mining for behavior-based malware detection. IEEE Trans Dependable Secur Comput. https://doi.org/10.1109/tdsc.2017.2675881

[18]. Bhattacharya A, Goswami RT (2017) Comparative analysis of different feature ranking techniques in data mining-based android malware detection. In: Satapathy SC, Bhateja V, Udgata SK, Pattnaik PK (eds) Proceedings of the 5th international conference on frontiers in intelligent computing: theory and applications: FICTA 2016, Volume 1.Springer Singapore, Singapore, pp 39–49

[19]. Fan CI, Hsiao HW, Chou CH, Tseng YF (2015) Malware detection systems based on API log data mining. In: 2015 IEEE 39th annual computer software and applications conference, pp 255–260

[20]. Wang P, Wang Y-S (2015) Malware behavioural detection and vaccine development by using a support vector model classifier. J Comput Syst Sci 81:1012–1026. https://doi.org/10.1016/j.jcss.2014.12.014

[21]. Fraley JB, Figueroa M (2016) Polymorphic malware detection using topological feature extraction with data mining. In: SoutheastCon 2016, pp 1–7

[22]. Wu B, Lu T, Zheng K, Zhang D, Lin X (2014) Smartphone malware detection model based on artificial immune system. China Commun 11:86–92. https://doi.org/10.1109/CC.2014.7022530

[23]. Bat-Erdene M, Park H, Li H, Lee H, Choi MS (2017) Entropy analysis to classify unknown packing algorithms for mal-ware detection. Int J Inf Secur 16(3):227–248. https://doi.org/10.1007/s10207-016-0330-4

[24]. Cui B, Jin H, Carullo G, Liu Z (2015) Service-oriented mobile malware detection system based on mining strategies. Pervasive Mob Comput 24:101–116. https://doi.org/10.1016/j.pmcj.2015.06.006

[25]. Fan Y, Ye Y, Chen L (2016) Malicious sequential pattern mining for automatic malware detection. Expert Syst Appl 52:16–25. https://doi.org/10.1016/j.eswa.2016.01.002

[26]. Martín A, Menéndez HD, Camacho D (2016) MOCDroid: multi-objective evolutionary classifier for Android malware detection. Soft Comput 21:7405–7415. https://doi.org/10.1007/s00500-016-2283-y

[27]. Santos I, Brezo F, Ugarte-Pedrero X, Bringas PG (2013) Opcode sequences as representation of executables for data-mining-based unknown malware detection. Inf Sci 231:64–82. https://doi.org/10.1016/j.ins.2011.08.020

[28]. Rehman Z-U, Khan SN, Muhammad K, Lee JW, Lv Z, Baik SW, Shah PA, Awan K, Mehmood I (2017) Machine learning-assisted signature and heuristic-based detection of malwares in Android devices. Comput Electr Eng. https://doi.org/10.1016/j.compeleceng.2017.11.028

[29]. Altaher A (2016) An improved Android malware detection scheme based on an evolving hybrid neuro-fuzzy classifier (EHNFC) and permission-based features. Neural Comput Appl 28:4147–4157. https://doi.org/10.1007/ s00521-016-2708-7

[30]. Mohaisen A, Alrawi O, Mohaisen M (2015) AMAL: high-fidelity, behavior-based automated malware analysis and classification. Comput Secur 52:251–266. https://doi.org/10.1016/j.cose.2015.04.001

[31]. Yuan Z, Lu Y, Xue Y (2016) Droiddetector: android malware characterization and detection using deep learning. Tsinghua Sci Technol 21:114–123. https://doi.org/10.1109/TST.2016.7399288

[32]. Boukhtouta A, Mokhov SA, Lakhdari N-E, Debbabi M, Paquet J (2016) Network malware classification com-parison using DPI and flow packet headers. J Comput Virol Hacking Tech 12:69–100. https://doi.org/10.1007/ s11416-015-0247-x

[33]. Ding Y, Yuan X, Tang K, Xiao X, Zhang Y (2013) A fast malware detection algorithm based on objective-oriented association mining. Comput Secur 39(Part B):315–324. https://doi.org/10.1016/j.cose.2013.08.008

[34]. Eskandari M, Khorshidpour Z, Hashemi S (2013) HDM-Analyser: a hybrid analysis approach based on data mining techniques for malware detection. J Comput Virol Hacking Tech 9:77–93. https://doi.org/10.1007/ s11416-013-0181-8

[35]. Miao Q, Liu J, Cao Y, Song J (2016) Malware detection using bilayer behavior abstraction and improved one-class support vector machines. Int J Inf Secur 15:361–379. https://doi.org/10.1007/s10207-015-0297-6

[36]. Ming J, Xin Z, Lan P, Wu D, Liu P, Mao B (2016) Impeding behavior-based malware analysis via replacement attacks to malware specifications. J Comput Virol Hacking Tech 13:193–207. https://doi.org/10.1007/s11416-016-0281-3

[37]. Nikolopoulos SD, Polenakis I (2016) A graph-based model for malware detection and classification using system-call groups. J

Comput Virol Hacking Tech 13:29–46. https://doi.org/10.1007/s11416-016-0267-1

[38]. Sheen S, Anitha R, Natarajan V (2015) Android based malware detection using a multifeature collaborative decision fusion approach. Neurocomputing 151(Part 2):905–912. https://doi.org/10.1016/j.neucom.2014.10.004

[39]. Norouzi M, Souri A, Samad Zamini M (2016) A data mining classification approach for behavioral malware detection. J Comput Netw Commun 2016:9. https://doi.org/10.1155/2016/8069672

[40]. Galal HS, Mahdy YB, Atiea MA (2016) Behavior-based features model for malware detection. J Comput Virol Hacking Tech 12:59–67. https://doi.org/10.1007/s11416-015-0244-0

[41]. Mao W, Cai Z, Towsley D, Feng Q, Guan X (2017) Security importance assessment for system objects and malware detection. Comput Secur 68:47–68. https://doi.org/10.1016/j.cose.2017.02.009

[42]. Dali Z, Hao J, Ying Y, Wu D, Weiyi C (2017) DeepFlow: deep learning-based malware detection by mining Android application for abnormal usage of sensitive data. In: 2017 IEEE symposium on computers and communications (ISCC), pp 438–443

[43]. Li Z, Sun L, Yan Q, Srisa-an W, Chen Z (2017) DroidClassifier: efficient adaptive mining of application-layer header for classifying android malware. In: Deng R, Weng J, Ren K, Yegneswaran V (eds) Security and privacy in communication networks: 12th international conference, securecomm 2016, Guangzhou, China, October 10–12, 2016, Proceedings. Springer International Publishing, Cham, pp 597–616.

[44]. Narayanan A, Chandramohan M, Chen L, Liu Y (2017) A multi-view context-aware approach to Android malware detection and malicious code localization. Empir Softw Eng. https://doi.org/10.1007/s10664-017-9539-8.

[45]. Malhotra A, Bajaj K (2016) A hybrid pattern based text mining approach for malware detection using DB Scan. CSI Trans ICT 4:141–149. https://doi.org/10.1007/s40012-016-0095-y.

[46]. "Number of available Android applications - AppBrain." [Online]. Available: http://www.appbrain.com/stats/number-of-android-apps. [Accessed: 28-Oct-2018].