UNIVERSITI PUTRA MALAYSIA

ANDROID MALWARE DETECTION WITH ENSEMBLE OF ANDROIDMANIFEST FEATURES

MOHAMMAD SALEHI

FSKTM 2019 18

ANDROID MALWARE DETECTION WITH ENSEMBLE OF ANDROIDMANIFEST FEATURES

By

MOHAMMAD SALEHI

Thesis submitted to Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, in fulfillment of the requirements for the Master of Information Security

JUNE 2019

COPYRIGHT PAGE

DEDICATIONS

I would like to dedicate this thesis to my beloved parents who supported me constantly and did not let me give up whenever I was close to it. Also, my supportive family and friends which all the time were there for me. Lastly, to my lecturers who besides Information Security courses, taught me valuable lessons about life which I'll always remember them.

Abstract of thesis presented to the Senate of University Putra Malaysia in fulfillment of the requirement for the degree of Master of Information Security

Android Malware Detection with Ensemble of AndroidManifest Features

By

MOHAMMAD SALEHI

JUNE 2019

Supervisor: Dr. Mohd Yunus Sharum

Faculty: Faculty of Computer Science and Information Technology

The popularity of Android Operating System rose gradually in the past years. Android becomes the first choice of the users in the second quarter of 2019 with more than 75 percent of worldwide market share. Furthermore, most of the users are keeping their personal information on their mobile devices. Consequently, Android is the main target of attackers on mobile and portable devices. In order to protect users' privacy and data, numerous researches have been done with different approaches. There are two main methods for analyzing and investigating applications. The first one is a static analysis which is the most common method that extracts static features from Android Package (APK) files. AndroidManifest features are extracted from APK files for analyzing malware in this research. The second method is the dynamic analysis that collects data while operating the application in an isolated environment. Mostly, machine learning techniques are used in researches for classifying unknown samples. This study comes with a new framework which

is named as a Composite of AndroidManifest Features (CAMF) to detect Android malware. In the proposed framework, three different static features are extracted like, requested permissions, hardware features, and intent-filters. A single merged feature vector is created from the feature matrix of each static feature. This vector is used as input data to our supervised machine learning models. As a result, CAMF framework minimizes the number of features to 141. Hence, it reduced the false negative rate to 1.201 percent in comparison to the previous study which is nearly 5 percent in their string feature analyzes.

Abstrak tesis yang dikemukakan kepada Universiti Putra Malaysia sebagai memenuhi keperluan untuk ijazah Sarjana Keselamatan Maklumat

Pengesanan Perisian Android Dengan Sambungan Ciri Statik

Oleh

MOHAMMAD SALEHI

JUN 2019

Penyelia: Dr. Mohd Yunus Sharum

Fakulti: FAKULTI SAINS KOMPUTER DAN TEKNOLOGI MAKLUMAT

Populariti Sistem Operasi Android semakin meningkat dalam beberapa tahun yang lalu. Android menjadi pilihan pertama pengguna pada suku kedua pada tahun 2019 dengan lebih daripada 75 peratus bahagian pasaran di seluruh dunia. Tambahan lagi, Majoriti pengguna akan menyimpan maklumat peribadi mereka pada peranti mudah alih. Akibatnya, system operasi Android menjadi sasaran utama bagi penyerang untuk menyerang peranti mudah alih. Bagi melindungi privasi dan data pengguna, pelbagai penyelidikan telah dilakukan dengan kaedah yang berlainan. Terdapat dua kaedah utama untuk menganalisis dan menyiasat lebih mendalam berkenaan aplikasi Sistem Operasi Android. Kaedah pertama adalah analisis statik yang merupakan kaedah paling umum mengekstrak ciri statik dari fail Pakej Android (APK). Dalam penyelidikan ini ciri Android Manifest diekstrak dari fail APK untuk menganalisis malware. Kedua adalah kaedah menganalisis dinamik yang

mengumpul data semasa aplikasi operasi dalam persekitaran terpencil. Kebanyakan teknik pembelajaran mesin digunakan dalam penyelidikan untuk mengklasifikasikan sampel yang tidak diketahui. Kajian ini datang dengan dapatan kajian yang baru yang dinamakan sebagai Composite of Android Manifest Features (CAMF) untuk mengesan malwares Android. Berdasarkan Kaedah CAMF yang dicadangkan, ia merangkumi tiga ciri statik diekstrak berbeza seperti, kebenaran yang diminta, ciri perkakasan dan penapis niat. Satu ciri vektor yang digabungkan dicipta dari ciri matriks pada setiap ciri statik. Vektor ini digunakan sebagai data kemasukan di model pembelajaran mesin untuk tujuan pemantauan. Hasil dari kaedah CAMF ia dapat meminimumkan bilangan ciri vector kepada 141. Oleh itu, hasil dapatan kajian mendapati kaedah ini dapat mengurangkan kadar negatif palsu kepada 1.201 peratus berbanding dengan kajian sebelumnya yang hampir 5 peratus dalam analisis ciri rentetan mereka.

.

ACKNOWLEDGMENT

First of all, I thank God for his blessings and mercy. During the last two years, I had a chance to be part of UPM community and living in Malaysia. At the same time, I met many students from different countries and learned a lot about different cultures which broaden my horizon. I learn so many things from great lectures especially security and information technology. Next, I would like to dedicate this appreciation to my parents Fereydoun and Zahra who always give me endless moral support and encouragement. Also, my appreciation goes to my supervisor, Dr.Mohd Yunus Sharum who is patient in helping me and giving guidelines for improvements. I am very thankful toward his effort and sharing of knowledge to support me until the end of this project. Not to be forgotten, thanks to my fellow friends who are not tired of giving an opinion and constructive comments that able to improve my project.

APPROVAL FORM

This thesis was submitted to the Senate of Universiti Putra Malaysia and has been accepted as fulfillment of the requirement for degree of Master Information Security. The members of the Supervisory Committee were as follows:

Dr. Mohd Yunus Sharum

Faculty of Computer Science and Information Technology

Universiti Putra Malaysia

(Supervisor)

Date: June 2019

DECLARATION FORM

Declaration by graduate student

I hereby confirm that:

- This thesis is my original work
- Quotations, illustrations and citations have been duly referenced
- This thesis has not been submitted previously or concurrently for any other degree at any other institutions
- Intellectual property from the thesis and copyright of thesis are fully owned by Universiti Putra Malaysia (UPM)
- Written permission must be obtained from supervisor and Deputy Vice-Chancellor (Research and Innovation) before thesis is published in book form.
- There is no plagiarism or data falsification/fabrication in the thesis, and scholarly integrity was upheld as according to Rule 59 in Rules 2003 (Revision 2013-2014). The thesis has undergone plagiarism detection software.

Signature: _____ Date: _____

Name and Matric No: MOHAMMAD SALEHI (GS50383)

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

This introduction briefly explained about research background, the importance of security in Android environment, techniques which are used for detecting malware, problem statement, research objective, research scope, and thesis structure.

1.1 BACKGROUND INFORMATION

The popularity of the mobile phone is soared since they become more powerful and easy to use in daily life. The statistical stats that in 2018, 58% of visits of sites get from the mobile which illustrates that people are more willing to using their smartphones for surfing on the Internet instead of personal computers (PCs).(Enge, 2019)

In the fourth quarter of 2018, the number of worldwide PC shipment dropped by 4.3 percent compared to the fourth quarter of 2017(Gartner, 2019). It's clear that people are more willing to buy devices which be more portable. Thus, smartphones become their first choice.

There are several different operating systems in the market for a mobile device such as Android, iOS, KaiOS, Windows, and Samsung. Base on the report of web traffic analysis website, Statcounter, Android mobile operation system, with more than 75 percent of the worldwide market share in the second quarter of 2019, is the first choice of the users(Statcounter, 2019). Most users keep

and store their personal information in their devices that attract attackers to develop their malicious applications (malapps) for the Android platform. Kaspersky Lab detected 5,730,916 mobile malicious installation packages in 2017 (Roman Unuchek, 2018). Android is an open source operating system and the user are able to install the application from third-party markets. Some third party markets are not using any mechanism in detecting malware. Thus, malware is more likely to be installed by users. Mobile malware is divided into several categories according to their behavior and goal.

1.1.1 MALWARE

There are two kinds of malware in the Android system. The first one is self-propagating which is used several different strategies to automatically installed on victims' phone like worms. The second one uses social engineering techniques to manipulate the user to install the application (apps) manually. Herein, mobile malware is divided base on their behaviors such as Adware, mobile Botnet, Scareware, Mobile spyware, mobile worm, mobile Trojans and mobile virus(Yan & Yan, 2018).

1.1.2 DETECTION METHODS

Hence, many research has been done in this area since for the first time Microsoft Lab used code analysis and monitoring behavior to detect Malware applications in 2008. Until now has been used many approaches to detect malicious apps, which static and dynamic approach are two main approaches in this area (Zachariah, Akash, Yousef, & Chacko, 2017). Android application

package has many features which the researcher can use for analyzing the applications. In(Feizollah, Anuar, Salleh, & Wahab, 2015) mobile malware features are divided into four different categories for detection. The first group is Static features which extract from Android packaging (APK) file. Also, there are various features, namely requested permissions, Java code, network address, intent filters, string features, and hardware components. Static features are widely used by researchers for increasing the detection rate. The AndroidManifest.xml file contains necessary information which causes the operating system cannot run the codes without it and every application should have the manifest file in its root directory. In this research, for detecting malware applications, we used three different features in AndroidManifest files.

The second category, dynamic features, is defined as the behavior of the application while it is running on the device. The features which are related to the operating system and network are used for investigation. System calls along with network traffic are two main features that recent works used them since application needs resources and services to be able to run on the device. Most application request network connection to receive and send data, receive updates or leaking personal information of users to the attackers. Additionally, user interaction is another feature which has been used beside system component for analyzing CPU and memory usage and running process of the mobile device. Hybrid features are defined as a combination of static and dynamic features that used simultaneity to get more accuracy in detection. In the fourth category, the researcher used Android application metadata that user can review the prior installation of the application such as description of

apps, rate of them in google play, the requested permission additional developer information.

In Figure 1.1 there is the taxonomy of Android feature selection for detecting attacks:

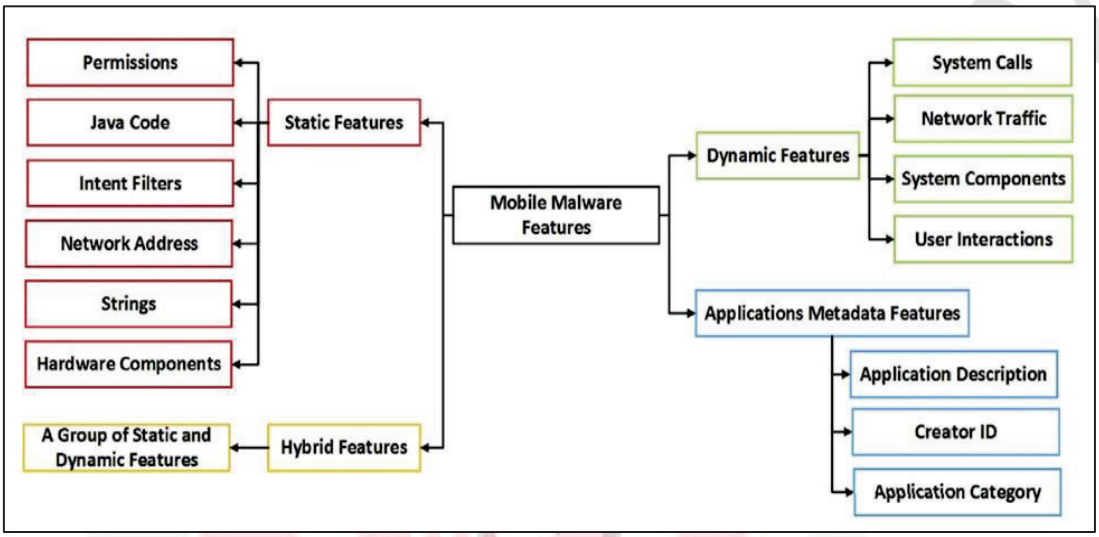

Figure 1.1 Taxonomy of mobile malware features (Feizollah et al. 2015)

In March 2018, Google play store was placed at 3.3 million apps(Google, 2017a), one of a most common method for malware detection is using rule and signature-based approach, while, the detection rate of this method is declined since it depends on fingerprint database (Fang et al. , 2014). Therefore, the learning-based approach has been used to detect malware from benign application automatically. The most popular method in automated learning is using Machine Learning Technique. Supervised and unsupervised learning are used for detecting malware. In a supervised method, the model is trained based on labeled data while in unsupervised, the data are clustered based on similarity (Martinelli et al., 2017). In view of such condition, Deep Learning

allowed using multiple layers to learn data which can implement multiple levels of abstraction (LeCun et al. , 2015).

Refer to the (Makridakis et al., 2018) Machine Learning methods computational cost is so high compare than statistical ones. Also, they are not the panacea that would increase detection accuracy automatically.

Here we are using the supervised method and three different classification models, Support Vector Machine, Random Forest and K-Neatest neighbor, are used for detecting the malware. There are different evaluation measurements that are used in android malware detection. The confusion matrix is usually used in detection systems. The result of the experiment is shown in four different elements by a confusion matrix (Davis & Goadrich, 2006).

The false negative (FN) which represent the number of malware which wrongly classified as the normal application is the most important element in the confusion matrix. Thus, we focused on beside reduce the false negative rate in our classification model.

1.2 PROBLEM STATEMENT

The design and structure of malicious application are going to be more complex compare than previous (Riad & Ke, 2018). Therefore, with just extracting a single feature, it is ambiguous to get a high percentage in detection rate (J. Li et al., 2018). Nowadays, researchers extracting several static features to increase their detection rate (Feizollah et al., 2017)(D. Li et al. , 2018). However, Wang et al. used six different static features which three of them are extracted from manifest file and other three features from .dex. They

specify those features as string features. Just number of the features they extracted from the .dex file is more than 34,000 features (Wang et al., 2018). Although they could get a high percentage of accuracy with sting features, the number of malware which wrongly classified as benign is an unforeseen event.

As they mentioned, some of the malware samples do not have enough suspicious features which caused they wrongly classified as a benign application. Clearly, 19 samples form 389 malware applications misclassified by their string features which cause false negative rate reach to just under 5 percent. In malware detection systems, the false negative rate is the most important metric.

1.3 RESEARCH OBJECTIVE

The main objective of this research is, improving the detection rate with just using the AndroidManifest features. With the use of AndroidManifest features, we are able to reduce the number of our features in our detection model to avoid some malware with the low number of suspicious features be wrongly classified in benign batch. As a result, the rate of our false negative is reduced simultaneously in the proposed model.

1.4 RESEARCH SCOPE

In Android malware detection has different phases. There are two main analyzing methods for detecting malware. The first one is static analysis which uses static features and the second one is dynamic analysis.

Static analysis is used as our method for detecting malware and we used AndroidManifest features as our static features. AndroidManifest.xml file is one of APK component that each application should have it and the operating system use it to run the app codes. The requested permission, hardware features, and intent-filter are extracted from the manifest file. All the feature matrix are merged into a single feature vector to be used on our machine learning models. the machine learning models are used as a classifier to detect malware sample in our unknown dataset.

1.5 THESIS STRUCTURE

This thesis contains six chapters including Introduction, Literature Review, Methodology, Implementation, Result and Discussion, and Conclusion is defined as a final chapter.

Introduction chapter briefly explains the reasons and our motivation for researching on the Android environment, different types of features are used by researchers. Besides that, our problem statement, research objective, and research scope are defined.

Chapter two, explain about the history of mobile malware, Android operating system and stricter of APK files. The security features which is used on the Android environment is discussed. Additionally, the different type of malware is introduced to provide an overview of the types of malware the proposed method tries to detect. In the end, based on the different analysis method, several cutting edge articles' method and the result are provided. Also in the bar charts, their results are compared.

In chapter three, we defined our research methodology and its requirements. The proposed framework that contains five different phases is fully covered and each phase is explained in details. The features are extracted from AndroidManifest.xml file are introduced beside different approaches which are used for generating feature matrixes. Moreover, classification models and evaluation metrics are discussed.

Chapter four includes different parts of framework implementation. It explained in details about the tools and libraries which is used for extracting and merging

features. It also shows the sample of codes as well as data extracted from APK files. Different techniques are used for training and testing the machine learning models is described clearly.

Chapter five represents the results of the proposed framework in details. The result of every single AndroidManifest feature is shown in a single table. Also, the result of merged features is compared in different machine learning models. Finally, the result of the proposed framework is compared with other techniques.

For sum up, in chapter 6 we summarized the objective and result for the research and future work is discussed in details.

REFERENCE

Androgurd. (2018). Welcome to Androguard's documentation! — Androguard 3.4.0 documentation. Retrieved June 11, 2019, from https://androguard.readthedocs.io/en/latest/#

ApkTool. (2011). Apktool - A tool for reverse engineering 3rd party, closed, binary Android apps. Retrieved June 14, 2019, from https://ibotpeaches.github.io/Apktool/

Armando, A., Merlo, A., Migliardi, M., & Verderame, L. (2012). Would you mind forking this process? A denial of service attack on android (and some countermeasures). In *IFIP Advances in Information and Communication Technology* (Vol. 376 AICT, pp. 13–24). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-30436-1_2

Arp, D., Spreitzenbarth, M., Hübner, M., Gascon, H., & Rieck, K. (2014). Drebin: Effective and Explainable Detection of Android Malware in Your Pocket. In *Proceedings 2014 Network and Distributed System Security Symposium*. Reston, VA: Internet Society. https://doi.org/10.14722/ndss.2014.23247

Arshad, S., Shah, M. A., Wahid, A., Mehmood, A., Song, H., & Yu, H. (2018). SAMADroid: A Novel 3-Level Hybrid Malware Detection Model for Android Operating System. *IEEE Access*, *6*, 4321–4339. https://doi.org/10.1109/ACCESS.2018.2792941

Arslan, R. S., Doğru, İ. A., & Barişçi, N. (2019). Permission-Based Malware Detection System for Android Using Machine Learning Techniques. *International Journal of Software Engineering and Knowledge Engineering*, *29*(01), 43–61. https://doi.org/10.1142/S0218194019500037

Au, K. W. Y., Zhou, Y. F., Huang, Z., & Lie, D. (2012). PScout. In *Proceedings of the 2012 ACM conference on Computer and communications security - CCS '12* (p. 217). New York, New York, USA: ACM Press. https://doi.org/10.1145/2382196.2382222

AVG.ThreatLabs. (2013). Android Dowgin. Retrieved from http://www.avgthreatlabs.com/virus-and-malware-information/info/androiddowgin/

Beal, V. (2018). What is Mobile Application? Webopedia Definition. Retrieved June 4, 2019, from https://www.webopedia.com/TERM/M/mobile_malware.html

Bläsing, T., Batyuk, L., Schmidt, A.-D., Camtepe, S. A., & Albayrak, S. (2010). An Android Application Sandbox system for suspicious software

detection. In *2010 5th International Conference on Malicious and Unwanted Software* (pp. 55–62). IEEE. https://doi.org/10.1109/MALWARE.2010.5665792

Booz, J., Mcgiff, J., Hatcher, W. G., Yu, W., Nguyen, J., & Lu, C. (2018). Tuning Deep Learning Performance for Android Malware Detection. *2018 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, 140–145.

Damshenas, M., Dehghantanha, A., Choo, K.-K. R., & Mahmud, R. (2015). M0Droid: An Android Behavioral-Based Malware Detection Model. *Journal of Information Privacy and Security*, *11*(3), 141–157. https://doi.org/10.1080/15536548.2015.1073510

Davis, J., & Goadrich, M. (2006). The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning - ICML '06* (pp. 233–240). New York, New York, USA: ACM Press. https://doi.org/10.1145/1143844.1143874

Dini, G., Martinelli, F., Saracino, A., & Sgandurra, D. (2012). MADAM: A Multi-level Anomaly Detector for Android Malware (pp. 240–253). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-33704-8_21

Enge, E. (2019). Mobile vs Desktop Traffic in 2019. Retrieved June 2, 2019, from https://www.stonetemple.com/mobile-vs-desktop-usage-study/

ESET. (2016). *The Rise of Android Ransomware Document version: 1.0 The Rise of Android Ransomware 2 Contents*. Retrieved from https://www.welivesecurity.com/wp-content/uploads/2016/02/Rise_of_Android_Ransomware.pdf

Fang, Z., Han, W., & Li, Y. (2014). Permission based Android security: Issues and countermeasures. *Computers & Security*, *43*, 205–218. https://doi.org/10.1016/J.COSE.2014.02.007

Feizollah, A. (2017). *A MALWARE ANALYSIS AND DETECTION SYSTEM FOR MOBILE DEVICES*. *FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY UNIVERSITY OF MALAYA KUALA LUMPUR*. Retrieved from http://studentsrepo.um.edu.my/8139/

Feizollah, A., Anuar, N. B., Salleh, R., Suarez-Tangil, G., & Furnell, S. (2017). AndroDialysis: Analysis of Android Intent Effectiveness in Malware Detection. *Computers & Security*, *65*, 121–134. https://doi.org/10.1016/j.cose.2016.11.007

Feizollah, A., Anuar, N. B., Salleh, R., & Wahab, A. W. A. (2015). A review on feature selection in mobile malware detection. *Digital Investigation*,

13, 22–37. https://doi.org/10.1016/j.diin.2015.02.001

Felt, A. P., Chin, E., Hanna, S., Song, D., & Wagner, D. (2011). Android permissions demystified. In *Proceedings of the 18th ACM conference on Computer and communications security - CCS '11* (p. 627). New York, New York, USA: ACM Press. https://doi.org/10.1145/2046707.2046779

Feng, P., Ma, J., Sun, C., & Ma, Y. (2018). A Novel Dynamic Android Malware Detection System With Ensemble Learning. *IEEE Access*, *6*(c), 30996–31011. https://doi.org/10.1109/ACCESS.2018.2844349

Gartner. (2019). Gartner Says Worldwide PC Shipments Declined 5.7 Percent in Third Quarter of 2016. Retrieved from https://www.gartner.com/en/newsroom/press-releases/2019-01-10-gartner-says-worldwide-pc-shipments-declined-4-3-perc

Gascon, H., Yamaguchi, F., Arp, D., & Rieck, K. (2013). Structural detection of android malware using embedded call graphs. *Proceedings of the 2013 ACM Workshop on Artificial Intelligence and Security - AISec '13*, 45–54. https://doi.org/10.1145/2517312.2517315

Google. (2016a). Google Online Security Blog: Android Security 2015 Annual Report. Retrieved June 3, 2019, from https://security.googleblog.com/2016/04/android-security-2015-annual-report.html

Google. (2016b). Hardware components| Android Developers. Retrieved June 5, 2019, from https://developer.android.com/guide/topics/manifest/uses-feature-element

Google. (2017a). • Google Play Store: number of apps 2009-2017 | Statistic. Retrieved September 29, 2018, from https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/

Google. (2017b). Intents and Intent Filters | Android Developers. Retrieved June 5, 2019, from https://developer.android.com/guide/components/intents-filters

Google. (2018). Permissions overview | Android Developers. Retrieved June 5, 2019, from https://developer.android.com/guide/topics/permissions/overview#permission_enforcement

Gunasekera, S. (2012). *Android Apps Security*. *Android Apps Security*. Berkeley, CA: Apress. https://doi.org/10.1007/978-1-4302-4063-1

Harrison, O. (2018). Machine Learning Basics with the K-Nearest Neighbors

Algorithm. Retrieved June 14, 2019, from https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761

Hido, S., & Kashima, H. (2009). A Linear-Time Graph Kernel. In *2009 Ninth IEEE International Conference on Data Mining* (pp. 179–188). IEEE. https://doi.org/10.1109/ICDM.2009.30

Jang, J., Kang, H., Woo, J., Mohaisen, A., & Kim, H. K. (2016). Andro-Dumpsys: Anti-malware system based on the similarity of malware creator and malware centric information. *Computers & Security*, *58*, 125–138. https://doi.org/10.1016/j.cose.2015.12.005

Jensen, R., & Shen, Q. (2008). *Computational intelligence and feature selection : rough and fuzzy approaches*. IEEE Press. Retrieved from https://books.google.com.my/books?hl=en&lr=&id=rgDDCeY-COkC&oi=fnd&pg=PR5&dq=jensen,+r.,+%26+shen,+q.+(2008).+computational+intelligence+and+feature+selection:+rough+and+fuzzy+approaches.&ots=9QhN9GpSeO&sig=Jf5XCS1lUHDNjZl3oMv9dFSOHzc#v=onepage&q=jensen%2C r.%2C %26 shen%2C q. (2008). computational intelligence and feature selection%3A rough and fuzzy approaches.&f=false

KasperSky. (2016). Mobile malware evolution 2015. Retrieved June 3, 2019, from https://securelist.com/mobile-malware-evolution-2015/73839/

Lashkari, A. H., Kadir, A. F. A., Taheri, L., & Ghorbani, A. A. (2018). Toward Developing a Systematic Approach to Generate Benchmark Android Malware Datasets and Classification. In *2018 International Carnahan Conference on Security Technology (ICCST)* (pp. 1–7). IEEE. https://doi.org/10.1109/CCST.2018.8585560

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Leonid Grustniy. (2018). The Rotexy Trojan: banker and blocker. Retrieved June 4, 2019, from https://www.kaspersky.com/blog/rotexy-banker-blocker/24733/

Li, D., Wang, Z., & Xue, Y. (2018). DeepDetector: Android Malware Detection using Deep Neural Network. In *2018 International Conference on Advances in Computing and Communication Engineering (ICACCE)* (pp. 184–188). IEEE. https://doi.org/10.1109/ICACCE.2018.8441737

Li, J., Sun, L., Yan, Q., Li, Z., Srisa-An, W., & Ye, H. (2018). Significant Permission Identification for Machine-Learning-Based Android Malware Detection. *IEEE Transactions on Industrial Informatics*, *14*(7), 3216–3225. https://doi.org/10.1109/TII.2017.2789219

Liu, H., & Motoda, H. (2008). *Computational methods of feature selection*. Chapman & Hall/CRC. Retrieved from https://dl.acm.org/citation.cfm?id=1207436

Liu, Y., Zhang, Y., Li, H., & Chen, X. (2016). A hybrid malware detecting scheme for mobile Android applications. In *2016 IEEE International Conference on Consumer Electronics (ICCE)* (pp. 155–156). IEEE. https://doi.org/10.1109/ICCE.2016.7430561

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). Statistical and Machine Learning forecasting methods: Concerns and ways forward. *PLOS ONE*, *13*(3), e0194889. https://doi.org/10.1371/journal.pone.0194889

Martinelli, F., Mercaldo, F., & Saracino, A. (2017). BRIDEMAID. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security - ASIA CCS '17* (pp. 899–901). New York, New York, USA: ACM Press. https://doi.org/10.1145/3052973.3055156

Pandita, R., Xiao, X., Yang, W., Enck, W., & Xie, T. (2013). WHYPER : Towards Automating Risk Assessment of Mobile Applications W HYPER : Towards Automating Risk Assessment of Mobile Applications. In *USENIX Security Symposium* (pp. 527–542). Retrieved from https://www.usenix.org/conference/usenixsecurity13/technical-sessions/presentation/pandita

Riad, K., & Ke, L. (2018). RoughDroid: Operative Scheme for Functional Android Malware Detection. *Security and Communication Networks*, *2018*, 1–10. https://doi.org/10.1155/2018/8087303

Roman Unuchek. (2018). Mobile malware evolution 2017. Retrieved from https://securelist.com/mobile-malware-review-2017/84139/

Sahin, D. O., Kural, O. E., Akleylek, S., & Kilic, E. (2018). New results on permission based static analysis for Android malware. In *2018 6th International Symposium on Digital Forensic and Security (ISDFS)* (pp. 1–4). IEEE. https://doi.org/10.1109/ISDFS.2018.8355377

Sarma, B. P., Li, N., Gates, C., Potharaju, R., Nita-Rotaru, C., & Molloy, I. (2012). Android permissions. In *Proceedings of the 17th ACM symposium on Access Control Models and Technologies - SACMAT '12* (p. 13). New York, New York, USA: ACM Press. https://doi.org/10.1145/2295136.2295141

Statcounter. (2019). Mobile Operating System Market Share Worldwide | StatCounter Global Stats. Retrieved June 2, 2019, from http://gs.statcounter.com/os-market-share/mobile/worldwide

statista. (2019). • Google Play Store: number of apps 2018 | Statista.

Retrieved June 10, 2019, from https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/

TrendMicro. (2012). *A Brief History of Mobile Malware*. Retrieved from https://countermeasures.trendmicro.eu/wp-content/uploads/2012/02/History-of-Mobile-Malware.pdf

Wang, W., Gao, Z., Zhao, M., Li, Y., Liu, J., & Zhang, X. (2018). DroidEnsemble: Detecting Android Malicious Applications with Ensemble of String and Structural Static Features. *IEEE Access*, *6*, 1–1. https://doi.org/10.1109/ACCESS.2018.2835654

Wei, F., Li, Y., Roy, S., Ou, X., & Zhou, W. (2017). Deep Ground Truth Analysis of Current Android Malware. Bonn, Germany. Retrieved from http://amd.arguslab.org/.

Yan, P., & Yan, Z. (2018). A survey on dynamic mobile malware detection. *Software Quality Journal*, *26*(3), 891–919. https://doi.org/10.1007/s11219-017-9368-4

Zachariah, R., Akash, K., Yousef, M. S., & Chacko, A. M. (2017). Android malware detection a survey. In *2017 IEEE International Conference on Circuits and Systems (ICCS)* (pp. 238–244). IEEE. https://doi.org/10.1109/ICCS1.2017.8325997