JPPIPA 9(7) (2023)

Jurnal Penelitian Pendidikan IPA

Journal of Research in Science Education

http://jppipa.unram.ac.id/index.php/jppipa/index

Defending Your Mobile Fortress: An In-Depth Look at on-Device Trojan Detection in Machine Learning: Systematic Literature Review

Lila Setiyani[1], Koo Tito Novelianto[2], Rusdianto Roestam[2], Sella Monica[2], Ayu Nur Indahsari[2], Amadeuz Ezrafel[2], Alinda Endang Poerwati[2], Yuliarman Saragih[3*]

[1]Information Systems Study Program, STMIK ROSMA, Karawang, Indonesia
[2]Information Study Program Informatics, Universitas President, Bekasi, Indonesia
[3]Electrical Engineering Study Program, Universitas Singaperbangsa, Karawang, Indonesia

Received: May 10, 2023
Revised: July 12, 2023
Accepted: July 25, 2023
Published: July 31, 2025

Corresponding Author:
Yuliarman Saragih
yuliarman@gmail.com

DOI: 10.29303/jppipa.v9i7.4209

Abstract: Mobile app trojans are becoming an increasingly serious threat to personal information security. They can cause severe damage by exposing sensitive and personally-identifying information to malicious actors. This paper's contribution is a comprehensive review of the attack vectors for trojan attacks, and ways to eliminate the risks posed by attack vectors and generate settlement automatically. As such, such attacks must be prevented. In this study, we explore to find how to detect the trojan attack in detail, and the way that we know in machine learning. A review is conducted on the state-of-the-art methods using the preferred reporting items for reviews and meta-analyses (PRISMA) guidelines. We review literature from several publications and analyze the use of machine learning for on-device trojan detection. This review provides evidence for the effectiveness of machine learning in detecting such threats. The current trend shows that signature-based analysis using various metadata, such as permission, intent, API and system calls, and network analysis, are capable of detecting trojan attacks before and after the initial infection.

Keywords: Machine Learning; on-device detection; PRISMA: Trojan

Introduction

The development of mobile devices has brought security challenges, reports from Weichbroth & Łysik (2020), explained that mobile devices such as Android have become one of the main targets for attackers to spread mobile malware, especially Trojan viruses. In the context of security evaluation Riadi et al. (2022) describes a trojan attack capable of stealing mobile device user credentials such as important information including system information, contacts, call logs, messages, and full access to the victim device's system directory. In a literature review conducted by Alzubaidi (2021), malware such as trojan viruses infect Android mobile devices via Google Play. The rise of cybercrimes targeting Android devices with Malware, (Saeed Jawad & Hlayel, 2022) informed one of the most popular malware of which is the Remote Access Trojan (RAT) which allows potential malicious users to control the system remotely, malware infection according to (Du et al., 2022) can be caused by social engineering, besides that malware developers use Fully Undetected (FUD) techniques this makes users unable to detect it.

Another study conducted by Zhao (2022), informs that hackers have made a lot of efforts to produce malware and find mobile device vulnerabilities, therefore an understanding of the concept of trojan malware infection, and mobile device vulnerabilities need to be understood by users. The issue of preventing cell phone viruses is very important. The development of handling mobile device vulnerabilities from trojan malware has been carried out by applying detection using machine learning. Mcdonald et al. (2021a), proposes the naïve Bayes algorithm to mine trojan

How to Cite:
Setiyani, L., Novelianto, K.T., Roestam, R., Monica, S., Indahsari, A.N., Ezrafel, A., Poerwati, A.E., & Saragih, Y. (2023). Defending Your Mobile Fortress: An In-Depth Look at on-Device Trojan Detection in Machine Learning: Systematic Literature Review. *Jurnal Penelitian Pendidikan IPA*, *9*(7), 302–308. https://doi.org/10.29303/jppipa.v9i7.4209

criminal case clues on mobile devices, it helps to detect and find viruses at the very beginning of an attack. This is reinforced by the statement from (Ullah et al., 2022) which investigated the effectiveness of four machine learning algorithms (Random Forest, Support Vector Machine, Gaussian Naïve Bayes, and K-Means) in classifying apps as malicious or benign. Seeing the importance of the issue of mobile device vulnerability to trojan malware and the potential for machine learning algorithms to support trojan detection, it is interesting to conduct a study that can provide complete insight regarding the method of how trojans infect mobile devices and their spread, factors that motivate infection, cases that arise as a result of these infections, as well as the role of machine learning algorithms in dealing with the vulnerabilities that these mobile devices have against trojans.

This paper's contribution is a comprehensive review of the attack vectors for trojan attacks, and ways to eliminate the risks posed by attack vectors and generate settlement automatically. We identify evidence from research by considering the infection mechanism of the trojan, the device, and the machine learning algorithm used to detect the trojan. The main result is a comprehensive picture of trojan malware, the mechanism of trojan infection on mobile devices along with examples of cases that occurred on victims, as well as the effectiveness of machine learning algorithms in detecting trojans. The remainder of this paper is organized as follows. In the second part, we provide an overview of trojan malware.

The third section presents the research objectives and questions, which we will answer based on the results of a systematic review. In the fourth section, we describe the research method we followed during the systematic review, and we provide details for each of the steps involved. In the fifth section, we present the results of assessing the quality of the studies included in the review and answering research questions based on data taken from the selected studies. In the sixth section, we discuss research suggestions and in the final section, we conclude.

Trojan on Mobile Device

Malware in mobile applications, based on their purpose and behavior, is divided into several categories including trojans, viruses, worms, botnets, spyware, annoying advertising tools, etc. Trojans are malware that silently performs destructive actions such as stealing user information, damaging devices, changing system settings, and loading malicious applications (Ullah et al., 2022).

Method

We conducted this research as a systematic review by following the PRISMA guidelines (Ramadhan & Setiyani, 2020). The PRISMA guidelines provide several items that need to be considered in preparing a systematic review. In this study, we will mainly focus on several key items: the problem statement, the proposed solution, and the demonstrated effectiveness. This help form the basis of our assessment. Initially, we collected recent studies on on-device trojan detection and the potential of machine learning algorithms in detecting trojans based on a few select keywords. Then, we apply eligibility criteria to the collection. We only select literature published in 2017 or later to provide an overview of the latest trend. In addition, we limit the type of literature which is only literature in the form of journals and proceedings.

Result and Discussion

Our search of the IEEE and google scholar databases resulted in a total of 6270 citations. However, from 6270 existing literature, only 197 kinds of literature whose titles had relevance to keywords that have been set, most of the literature did not focus on the trojan malware and Android device. From 197 kinds of literature, we filtered duplicate literature, and we found 4 duplicate literature, so there were 193 kinds of literature left. From 193 kinds of literature, we read abstracts and contents, filtered literature that fit or was relevant to the purpose of systematic review, and finally, we obtained 43 kinds of literature that met the requirements. We made these results a reference for conducting a systematic review.

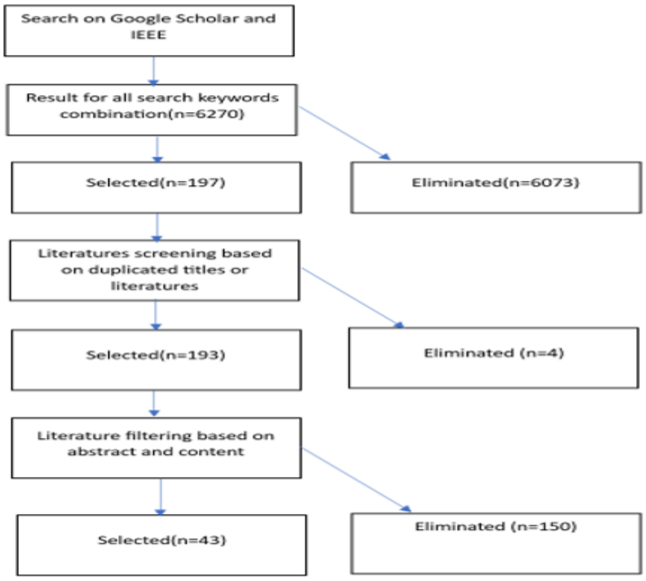
Search on Google Scholar and IEEE
Result for all search keywords combination(n=6270)
Selected(n=197)
Eliminated(n=6073)
Literatures screening based on duplicated titles or literatures
Selected(n=193)
Eliminated (n=4)
Literature filtering based on abstract and content
Selected(n=43)
Eliminated (n=150)

Figure 1. Flow process literature search based on PRISMA guidelines

The characteristics of the literature we received were a literature that was by the topic. Here, the attack vector is defined as the entry point where the malware may enter and infect the Android device. By understanding the attack vectors, special attention may be provided such that those entry points are more strictly guarded and compromising scenarios may be avoided.

Table 1. Trojan attack vectors in android

Sources	Trojan attack vectors in Androids
(Wang et al., 2017), (Huang & Kao, 2018), (T. Kim et al., 2019), (Surendran et al., 2020a), (Chen et al., 2021), (Feng et al., 2021), (Mohamad Arif, Ab Razak, Tuan Mat, et al., 2021), (Sasidharan & Thomas, 2021), (Acharya et al., 2022), (Alani & Awad, 2022), (Peng et al., 2022), (Yadav et al., 2022)	Third-party software repository
(Garcia et al., 2017), (Palumbo et al., 2017), (Huang & Kao, 2018), (Shan et al., 2018), (Qamar et al., 2019), (Feng et al., 2021), (Sasidharan & Thomas, 2021), (Acharya et al., 2022), (Alani & Awad, 2022), (Peng et al., 2022)	Official software repository
, (Qamar et al., 2019), (Pektaş & Acarman, 2020), (Feng et al., 2021), (Hou et al., 2021), (Alani & Awad, 2022), (Peng et al., 2022), (Kulkarni & Javaid, 2018)	Repackaging
(Feng et al., 2021), (Peng et al., 2022)	Malicious website
(Mohamad Arif, Ab Razak, Awang, et al., 2021)	Phishing
(Wei et al., 2017)	Other application
(Wei et al., 2017)	Dynamic payload

As shown in Table 1, there are a variety of different attack vectors. It can be observed that most infections come from downloading and installing malicious applications from the platform's official store. In this scenario, the victim is under the impression that the applications are legitimate. This is unfortunate since most layman's experience with application installation is through the store. In addition, it is also shown that the infection may happen due to phishing attempts. The victim inadvertently clicked on a link that downloads the malicious application. Although the installation must be approved, the victim may not be aware of the implication of such action. Consequently, the victim is tricked to install the application. On the other hand, the victim may be intentionally obtaining them from dubious sources with the expectation of obtaining paid applications for free. The victim may have no way of verifying the authenticity of the application and whether it has been tampered with. These scenarios show that one of the major challenges in preventing trojan infections is establishing the legitimacy of individual applications.

After we list trojan attack vectors in androids, to answer RQ2, we categorized the literature based on the methods used to eliminate or minimize a risk arising from trojan attacks in Android devices. This is performed to understand the detection methodology used in the current state-of-the-art.

Table 2. Methods used to eliminate or minimize the risk of trojan attacks in Android

Sources	Methods
(Garcia et al., 2017), (Idrees et al., 2017), (Palumbo et al., 2017), (Huang & Kao, 2018), (Shan et al., 2018), (Ma et al., 2020), (Chen et al., 2021), (Dam & Touili, 2021), (Fan et al., 2021), (Feng et al., 2021), (Mohamad Arif, Ab Razak, Tuan Mat, et al., 2021), (Mohamad Arif, Ab Razak, Awang, et al., 2021), (Rathore et al., 2021), (Sasidharan & Thomas, 2021), (Alani & Awad, 2022), (J. Kim et al., 2022), (Peng et al., 2022), (Ullah et al., 2022), (Yadav et al., 2022)	Static Analysis
(Mahindra & Singh, 2017), (Zulkifli et al., 2018), (Aminuddin & Abdullah, 2019), (T. Kim et al., 2019), (Zhou et al., 2019), (John et al., 2020), (Pektaş & Acarman, 2020), (Xie et al., 2020), (Casolare et al., 2021), (Sayed et al., 2023)	Dynamic Analysis
(Dehkordy & Rasoolzadegan, 2020), (Fiky et al., 2021), (Hou et al., 2021), (Imtiaz et al., 2021), (Liu et al., 2021), (Acharya et al., 2022), (Martinelli et al., 2017)	Hybrid Analysis

Table 2 shows three major categories of the method used. Most detection methodologies use static analysis. Applications can be scanned and analyzed for trojans or any other malware threats. This is ideal since it can be performed before the application is installed or run. However, there might be limitations to this method. The heuristics used to discriminate malware from legitimate applications may be too lax or too sensitive, causing false negatives and false positives respectively. Additionally, the malware may be a self-modifying binary that can change its characteristics before and after it's run. The dynamic analysis also has its drawbacks. It analyzes the application's runtime behavior and decides whether it's malicious. This may incur runtime performance cost that makes applications run slowly. More importantly, it can be argued that once the malware is already installed and running, it has won. Nevertheless, there is still value in performing dynamic analysis for the mitigation of such infection. As such, a hybrid approach can be employed to get the best of both methods.

Table 3. Features used to eliminate or minimize the risk of trojan attack in androids

Sources	Features
(Garcia et al., 2017), (Shan et al., 2018), (T. Kim et al., 2019), (Zhou et al., 2019), (Dehkordy & Rasoolzadegan, 2020), (Ma et al., 2020), (Pektaş & Acarman, 2020), (Surendran et al., 2020a), (Chen et al., 2021), (Dam & Touili, 2021), (Fan et al., 2021), (Feng et al., 2021), (Fiky et al., 2021), (Hou et al., 2021), (Imtiaz et al., 2021), (Liu et al., 2021), (Sasidharan & Thomas, 2021), (Acharya et al., 2022), (Alani & Awad, 2022), (J. Kim et al., 2022), (Ullah et al., 2022), (Sayed et al., 2023)	API call
(Garcia et al., 2017), (Idrees et al., 2017), (Mahindru & Singh, 2017), (Palumbo et al., 2017), (T. Kim et al., 2019), (Dehkordy & Rasoolzadegan, 2020), (Chen et al., 2021), (Fan et al., 2021), (Fiky et al., 2021), (Hou et al., 2021), (Imtiaz et al., 2021), (Liu et al., 2021), (Mohamad Arif, Ab Razak, Tuan Mat, et al., 2021), (Mohamad Arif, Ab Razak, Awang, et al., 2021), (Rathore et al., 2021), (Acharya et al., 2022), (Alani & Awad, 2022), (Peng et al., 2022), (Martinelli et al., 2017)	Permission
(Garcia et al., 2017), (Aminuddin & Abdullah, 2019), (Zhou et al., 2019), (John et al., 2020), , (Surendran et al., 2020a), (Casolare et al., 2021), (Imtiaz et al., 2021), (Acharya et al., 2022) , (Martinelli et al., 2017)	System call
(Garcia et al., 2017), (Idrees et al., 2017), (Dehkordy & Rasoolzadegan, 2020), (Fiky et al., 2021), (Imtiaz et al., 2021), (Liu et al., 2021), (Acharya et al., 2022), (Alani & Awad, 2022), (Peng et al., 2022)	Intent
(Dehkordy & Rasoolzadegan, 2020), (Chen et al., 2021), (Hou et al., 2021), (Acharya et al., 2022)	Services
(Huang & Kao, 2018), (Chen et al., 2021), (Acharya et al., 2022), (Yadav et al., 2022)	Bitmap
(Zulkifli et al., 2018), (Dehkordy & Rasoolzadegan, 2020), (Xie et al., 2020), (Acharya et al., 2022)	Network traffic
(Garcia et al., 2017), (T. Kim et al., 2019)	Native code
(Palumbo et al., 2017), (Ma et al., 2020)	Dalvik code
(Fan et al., 2021), (Hou et al., 2021)	Developer information
(Garcia et al., 2017), (Wang et al., 2017)	Dynamic class loading
, (Martinelli et al., 2017)	Communication
(Fan et al., 2021), (Hou et al., 2021)	Application source
(Martinelli et al., 2017)	Opcode n-grams
(Martinelli et al., 2017)	Rating
(Martinelli et al., 2017)	Download count
(Martinelli et al., 2017)	Developer reputation
(Palumbo et al., 2017)	String resource
(Wang et al., 2017)	Hardware and software requirements
(Peng et al., 2022)	Dalvik bytecode

Table 3 shows the individual features used in the literature. Signature-based detection is the most common method used to detect trojans. This entails creating a compact representation that can be compared against a known list of trojans and other malware. A simple hash of the application package was previously used to identify threats. But in recent studies, specific features of the application are used. These include application permission, intent, APIs, and system calls. In static analysis, these features are extracted from the application package metadata and code content. However, in dynamic analysis, the API and system calls can be accumulated or intercepted. Additionally, network packets can be intercepted to detect suspicious network activities. Regardless, it can be concluded that these features are crucial in determining whether an application is malicious or not.

Conclusion

Mobile app trojans are serious threats to information security. However, the prevention of such attacks is non-trivial. We have formulated two research questions to discover the recent trends in trojan detection and prevention. The first is “What are the attack vectors for a trojan attack?” Based on the recent literature, we have established that the current trojan vectors can be broadly classified into three categories: trojan disguised as a legitimate application, phishing campaign, and dubious application downloads. Additionally, one of the major challenges in preventing a trojan infection is in establishing the legitimacy of a given application. The second question is “How to eliminate or minimize the risk posed by the attack vectors?” We have determined that there are three main approaches: static analysis, dynamic analysis, and hybrid. We have also shown the advantages and disadvantages of each approach and concluded that a hybrid approach would complement the disadvantages. Moreover, the current trend shows that signature-based analysis using various metadata, such as permission, intent, API and system calls, and network analysis, are capable of detecting trojan attacks before and after the initial infection.

Acknowledgments
Thanks to all parties who have supported the implementation of this research. I hope this research can be useful.

Author Contributions
Conceptualization, L. S., K. T. N., R. R., S. M., A. N. I., A. E., A. E. P., Y. S.; methodology, L. S.; validation, K. T. N and R. R.; formal analysis, S. M.; investigation, A. N. I and A. E.; resources, A. E. P and Y. S.; data curation, L. S.: writing—original draft preparation, K. T. N and R. R.; writing—review and editing, S. M.: visualization, A. N. I and A. E.; supervision, A. E. P.; project administration, Y. S.; funding acquisition, L. S and Y. S. All authors have read and agreed to the published version of the manuscript.

Funding
This research was independently funded by researchers.

Conflicts of Interest
The authors declare no conflict of interest.

References

Acharya, S., Rawat, U., & Bhatnagar, R. (2022). A Low Computational Cost Method for Mobile Malware Detection Using Transfer Learning and Familial Classification Using Topic Modelling. *Applied Computational Intelligence and Soft Computing*, *2022*, 1–22. https://doi.org/10.1155/2022/4119500

Alani, M. M., & Awad, A. I. (2022). PAIRED: An Explainable Lightweight Android Malware Detection System. *IEEE Access*, *10*, 73214–73228. https://doi.org/10.1109/ACCESS.2022.3189645

Alzubaidi, A. (2021). Recent Advances in Android Mobile Malware Detection: A Systematic Literature Review. *IEEE Access*, *9*, 146318–146349. https://doi.org/10.1109/ACCESS.2021.3123187

Aminuddin, N. I., & Abdullah, Z. (2019). Android Trojan Detection Based on Dynamic Analysis. *Advances in Computing and Intelligent System, 1*(1).Retrieved from http://www.fazpublishing.com/acis/index.php/acis/article/view/4

Casolare, R., Dominicis, C. D., Iadarola, G., Martinelli, F., Mercaldo, F., & Santone, A. (2021). Dynamic Mobile Malware Detection through System Call-based Image Representation. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, *12*(1), 44–63. https://doi.org/10.22667/JOWUA.2021.03.31.044

Chen, H., Li, Z., Jiang, Q., Rasool, A., & Chen, L. (2021). A Hierarchical Approach for Android Malware Detection Using Authorization-Sensitive Features. *Electronics*, *10*(4), 432. https://doi.org/10.3390/electronics10040432

Dam, K. H. T., & Touili, T. (2021). MADLIRA: A Tool for Android Malware Detection. *Proceedings of the 7th International Conference on Information Systems Security and Privacy*, 670–675. https://doi.org/10.5220/0010339506700675

Dehkordy, D. T., & Rasoolzadegan, A. (2020). DroidTKM: Detection of Trojan Families using the KNN Classifier Based on Manhattan Distance Metric. *2020 10th International Conference on Computer and Knowledge Engineering (ICCKE)*, 136–141. https://doi.org/10.1109/ICCKE50421.2020.9303720

Du, J., Raza, S. H., Ahmad, M., Alam, I., Dar, S. H., & Habib, M. A. (2022). Digital Forensics as Advanced Ransomware Pre-Attack Detection Algorithm for Endpoint Data Protection. *Security and Communication Networks, 2022*, 1–16. https://doi.org/10.1155/2022/1424638

Fan, Y., Ju, M., Hou, S., Ye, Y., Wan, W., Wang, K., ... & Xiong, Q. (2021). Heterogeneous Temporal Graph Transformer: An Intelligent System for Evolving Android Malware Detection. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining,* 2831-2839. https://doi.org/10.1145/3447548.3467168

Feng, R., Chen, S., Xie, X., Meng, G., Lin, S.-W., & Liu, Y. (2021). A Performance-Sensitive Malware Detection System Using Deep Learning on Mobile Devices. *IEEE Transactions on Information Forensics and Security*, *16*, 1563–1578. https://doi.org/10.1109/TIFS.2020.3025436

Fiky, A. H. E., Shenawy, A. E., & Madkour, M. A. (2021). Android Malware Category and Family Detection and Identification using Machine Learning. *ArXiv preprint.* https://doi.org/arXiv:2107.01927

Garcia, J., Hammad, M., & Malek, S. (2017). Lightweight, Obfuscation-Resilient Detection and Family Identification of Android Malware. *ACM Transactions on Software Engineering and Methodology*, *26*(3), 1–29. https://doi.org/10.1145/3162625

Hou, S., Fan, Y., Ju, M., Ye, Y., Wan, W., Wang, K., Mei, Y., Xiong, Q., & Shao, F. (2021). Disentangled Representation Learning in Heterogeneous Information Network for Large-scale Android Malware Detection in the COVID-19 Era and Beyond. *Proceedings of the AAAI Conference on*

Artificial Intelligence, *35*(9), 7754–7761. https://doi.org/10.1609/aaai.v35i9.16947

Huang, T. H.-D., & Kao, H.-Y. (2018). R2-D2: ColoR-inspired Convolutional Neural Network (CNN)-based AndroiD Malware Detections. *arXiv*. Retrieved from http://arxiv.org/abs/1705.04448

Idrees, F., Rajarajan, M., Conti, M., Chen, T. M., & Rahulamathavan, Y. (2017). PIndroid: A novel Android malware detection system using ensemble learning methods. *Computers & Security*, *68*, 36–46. https://doi.org/10.1016/j.cose.2017.03.011

Imtiaz, S. I., Rehman, S. ur, Javed, A. R., Jalil, Z., Liu, X., & Alnumay, W. S. (2021). DeepAMD: Detection and identification of Android malware using high-efficient Deep Artificial Neural Network. *Future Generation Computer Systems*, *115*, 844–856. https://doi.org/10.1016/j.future.2020.10.008

John, T. S., Thomas, T., & Emmanuel, S. (2020). Graph Convolutional Networks for Android Malware Detection with System Call Graphs. *2020 Third ISEA Conference on Security and Privacy (ISEA-ISAP)*, 162–170. https://doi.org/10.1109/ISEA-ISAP49340.2020.235015

Kim, J., Ban, Y., Ko, E., Cho, H., & Yi, J. H. (2022). MAPAS: A practical deep learning-based Android malware detection system. *International Journal of Information Security*, *21*(4), 725–738. https://doi.org/10.1007/s10207-022-00579-6

Kim, T., Kang, B., Rho, M., Sezer, S., & Im, E. G. (2019). A Multimodal Deep Learning Method for Android Malware Detection Using Various Features. *IEEE Transactions on Information Forensics and Security*, *14*(3), 773–788. https://doi.org/10.1109/TIFS.2018.2866319

Kulkarni, K., & Javaid, A. Y. (2018). Opensource android vulnerability detection tools: A survey. *ArXiv Preprint ArXiv:1807.11840*. https://doi.org/10.48550/arXiv.1807.11840

Liu, Z., Wang, R., Japkowicz, N., Tang, D., Zhang, W., & Zhao, J. (2021). Research on unsupervised feature learning for Android malware detection based on Restricted Boltzmann Machines. *Future Generation Computer Systems*, *120*, 91–108. https://doi.org/10.1016/j.future.2021.02.015

Ma, Z., Ge, H., Wang, Z., Liu, Y., & Liu, X. (2020). *Droidetec: Android Malware Detection and Malicious Code Localization through Deep Learning* (arXiv:2002.03594). arXiv. http://arxiv.org/abs/2002.03594

Mahindru, A., & Singh, P. (2017). Dynamic Permissions-based Android Malware Detection using Machine Learning Techniques. *Proceedings of the 10th Innovations in Software Engineering Conference*, 202–210. https://doi.org/10.1145/3021460.3021485

Martinelli, F., Mercaldo, F., & Saracino, A. (2017). BRIDESMAID: An hybrid tool for accurate detection of Android malware. *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, 899–901. https://doi.org/10.1145/3052973.3055156

Mcdonald, J., Herron, N., Glisson, W., & Benton, R. (2021a). *Machine Learning-Based Android Malware Detection Using Manifest Permissions*. Hawaii International Conference on System Sciences. https://doi.org/10.24251/HICSS.2021.839

Mcdonald, J., Herron, N., Glisson, W., & Benton, R. (2021b). *Machine Learning-Based Android Malware Detection Using Manifest Permissions*. Hawaii International Conference on System Sciences. https://doi.org/10.24251/HICSS.2021.839

Mohamad Arif, J., Ab Razak, M. F., Awang, S., Tuan Mat, S. R., Ismail, N. S. N., & Firdaus, A. (2021). A static analysis approach for Android permission-based malware detection systems. *PLOS ONE*, *16*(9), e0257968. https://doi.org/10.1371/journal.pone.0257968

Mohamad Arif, J., Ab Razak, M. F., Tuan Mat, S. R., Awang, S., Ismail, N. S. N., & Firdaus, A. (2021). Android mobile malware detection using fuzzy AHP. *Journal of Information Security and Applications*, *61*, 102929. https://doi.org/10.1016/j.jisa.2021.102929

Palumbo, P., Sayfullina, L., Komashinskiy, D., Eirola, E., & Karhunen, J. (2017). A pragmatic Android malware detection procedure. *Computers & Security*, *70*, 689–701. https://doi.org/10.1016/j.cose.2017.07.013

Pektaş, A., & Acarman, T. (2020). Deep learning for effective Android malware detection using API call graph embeddings. *Soft Computing*, *24*(2), 1027–1043. https://doi.org/10.1007/s00500-019-03940-5

Peng, T., Hu, B., Liu, J., Huang, J., Zhang, Z., He, R., & Hu, X. (2022). A Lightweight Multi-Source Fast Android Malware Detection Model. *Applied Sciences*, *12*(11), 5394. https://doi.org/10.3390/app12115394

Qamar, A., Karim, A., & Chang, V. (2019). Mobile malware attacks: Review, taxonomy & future directions. *Future Generation Computer Systems*, *97*, 887–909. https://doi.org/10.1016/j.future.2019.03.007

Ramadhan, A., & Setiyani, L. (2020). The Analysis Of Knowledge Management Process On Software Development Process: A Systematic Review. *Dinasti International Journal of Digital Business Management*, *1*(4), 522–535. https://doi.org/10.31933/DIJDBM

Rathore, H., Sahay, S. K., Nikam, P., & Sewak, M. (2021). Robust Android Malware Detection System against Adversarial Attacks using Q-Learning. *Information Systems Frontiers*, *23*(4), 867–882. https://doi.org/10.1007/s10796-020-10083-8

Riadi, I., Aprilliansyah, D., & Sunardi, S. (2022). Mobile Device Security Evaluation using Reverse TCP Method. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control,* 289-298. https://doi.org/10.22219/kinetik.v7i3.1433

Saeed Jawad, M., & Hlayel, M. (2022). *Intelligent Cybersecurity Threat Management in Modern Information Technologies Systems. In S. Ramakrishnan (Ed.), Lightweight Cryptographic Techniques and Cybersecurity Approaches*. IntechOpen. https://doi.org/10.5772/intechopen.105478

Sasidharan, S. K., & Thomas, C. (2021). ProDroid—An Android malware detection framework based on a profile hidden Markov model. *Pervasive and Mobile Computing*, *72*, 101336. https://doi.org/10.1016/j.pmcj.2021.101336

Sayed, M. I., Saha, S., & Haque, A. (2023). Deep Learning Based Malapps Detection in Android Powered Mobile Cyber-Physical System. *2023 International Conference on Computing, Networking and Communications (ICNC)*, 443–449. https://doi.org/10.1109/ICNC57223.2023.10074208

Shan, Z., Neamtiu, I., & Samuel, R. (2018). Self-hiding behavior in Android apps: Detection and characterization. *Proceedings of the 40th International Conference on Software Engineering*, 728–739. https://doi.org/10.1145/3180155.3180214

Surendran, R., Thomas, T., & Emmanuel, S. (2020a). A TAN-based hybrid model for Android malware detection. *Journal of Information Security and Applications*, *54*, 102483. https://doi.org/10.1016/j.jisa.2020.102483

Surendran, R., Thomas, T., & Emmanuel, S. (2020b). GSDroid: Graph Signal Based Compact Feature Representation for Android Malware Detection. *Expert Systems with Applications*, *159*, 113581. https://doi.org/10.1016/j.eswa.2020.113581

Ullah, S., Ahmad, T., Buriro, A., Zara, N., & Saha, S. (2022). TrojanDetector: A Multi-Layer Hybrid Approach for Trojan Detection in Android Applications. *Applied Sciences*, *12*(21), 10755. https://doi.org/10.3390/app122110755

Wang, X., Wang, W., He, Y., Liu, J., Han, Z., & Zhang, X. (2017a). Characterizing Android apps' behavior for effective detection of malaprops at a large scale. *Future Generation Computer Systems*, *75*, 30–45. https://doi.org/10.1016/j.future.2017.04.041

Wei, F., Li, Y., Roy, S., Ou, X., & Zhou, W. (2017). Deep ground truth analysis of current Android malware. *Detection of Intrusions and Malware, and Vulnerability Assessment: 14th International Conference, DIMVA 2017, Bonn, Germany, July 6-7, 2017, Proceedings 14*, 252–276. https://doi.org/10.1007/978-3-319-60876-1_12

Weichbroth, P., & Łysik, Ł. (2020). Mobile Security: Threats and Best Practices. *Mobile Information Systems*, *2020*, 1–15. https://doi.org/10.1155/2020/8828078

Xie, J., Li, S., Yun, X., Zhang, Y., & Chang, P. (2020). HSTF-Model: An HTTP-based Trojan detection model via the Hierarchical Spatio-temporal Features of Traffic. *Computers & Security*, *96*, 101923. https://doi.org/10.1016/j.cose.2020.101923

Yadav, P., Menon, N., Ravi, V., Vishvanathan, S., & Pham, T. D. (2022). A two-stage deep learning framework for image-based Android malware detection and variant classification. *Computational Intelligence*, *38*(5), 1748–1771. https://doi.org/10.1111/coin.12532

Zhao, F. (2022). Naive Bayes Algorithm Mining Mobile Phone Trojan Crime Clues. *Mobile Information Systems*, *2022*, 1–11. https://doi.org/10.1155/2022/6262147

Zhou, Q., Feng, F., Shen, Z., Zhou, R., Hsieh, M.-Y., & Li, K.-C. (2019). A novel approach for mobile malware classification and detection in Android systems. *Multimedia Tools and Applications*, *78*(3), 3529–3552. https://doi.org/10.1007/s11042-018-6498-z

Zulkifli, A., Hamid, I. R. A., Shah, W. M., & Abdullah, Z. (2018). Android Malware Detection Based on Network Traffic Using Decision Tree Algorithm. *Recent Advances on Soft Computing and Data Mining*, *700*, 485–494. https://doi.org/10.1007/978-3-319-72550-5_46