International Conference on Innovation in Engineering and Technology (ICIET) 23-24 December, 2019

A Systematic Review on Hybrid Analysis using Machine Learning for Android Malware Detection

Asadullah Hill Galib
Institute of Information Technology
University of Dhaka
bsse0712@iit.du.ac.bd

B M Mainul Hossain
Institute of Information Technology
University of Dhaka
mainul@iit.du.ac.bd

***Abstract*—Android is the most ubiquitous mobile operating system nowadays. It's prevalence also provokes the humongous growth of Android malware. Primarily researchers have focused on static and dynamic analysis using machine learning techniques to detect Android malware. But, multifarious evasion techniques by the shrewd malware authors have made those techniques limited and ineffective. Therefore, recent researchers have shifted their focus on discovering an effective strategy to fight against. Hybrid analysis: a fusion of static and dynamic analysis would be a good candidate for that as it prevails over the individual drawbacks of static and dynamic analysis with the cost of complexity. According to research, hybrid analysis has many opportunities as well as challenges. This work aims at presenting a thorough and systematic review of hybrid analysis using machine learning techniques for Android malware detection. It encompasses the leading researches on hybrid analysis: their contributions, strengths, and weaknesses. This work also discusses the challenges, opportunities and future directions of hybrid analysis for Android malware detection.**

***Index Terms*—Android Malware Detection, Hybrid Analysis, Machine Learning**

I. Introduction

Android is the most prevalent mobile operating system (OS) currently: 72.23% of total mobile OS is Android [1]. With the enormous growth of the Android system [2], Android malware also has grown significantly as well as upgraded its nature and activities [2]. On average 12,000 new malware instances are found per day [3]. To defend against that malware phenomenon, researchers emphasized on Android malware detection to ensure Android mobile application security.

To detect Android malware, there are three approaches: Static Analysis, Dynamic Analysis, and Hybrid Analysis. The static analysis uses the static features of the Android application such as Permissions, API Calls, etc. The dynamic analysis investigates the dynamic behavior of the application running on an emulated environment or on a real device. These dynamic features/behaviors include System Calls, Network Traffic, etc. Hybrid analysis tends to incorporate both the static and dynamic approaches into a common ground.

Static and dynamic analyses have their own limitations. Currently, malware authors are too smart to evade these detection techniques. They use many evasion techniques to evade the analysis. For static analysis, commonly used evasion techniques by the malware authors are data obfuscation, control flow obfuscation, encryption, reflection, dynamically loaded code, repackaging, etc. [4]. For dynamic analysis, anti-analysis, mimicry, data obfuscation, misleading information flows and function in-directions, etc. are used as evasion techniques [4]. Besides, limited code coverage lessens the effectiveness of the dynamic analysis.

As static and dynamic analysis have their weaknesses individually, combining both analyses into a common ground would be helpful. The hybrid analysis approach integrates both static and dynamic analyses to mitigate their weaknesses. Though hybrid analysis is complex enough, it is effective and feasible according to related research. But, comparatively a few works have been performed in hybrid analysis. Researchers nowadays focuses on it because of its effectiveness and potential.

Though there exists some reviews on Android malware detection, none of them focused on hybrid analysis using machine learning. For instance, Tam et al. [4] depicted the evolution of Android malware and analysis techniques, but they did not give too much focuses on hybrid analysis. Qamar et al. [5] presented an all-inclusive review on mobile malware, though they nearly overlooked the hybrid analysis approach. Baskaran et al. [6] covered hybrid analysis in their Android malware detection review in parallel with static and dynamic analysis. Naway et al. [7] focused on deep learning techniques and Feizollah et al. [8] investigated feature selection for analysis in their reviews. None of them provided an in-depth investigation of hybrid analysis.

Due to hybrid analysis approach's huge potential and importance in Android malware detection, a brief review of the existing researches on hybrid analysis is necessary. In this work, we provide a comprehensive and systematic review of the hybrid analysis approach in Android malware detection, analyzed the existing works: their strengths and weaknesses and discussed challenges, opportunities and future directions in this regard.

To be specific, this work makes the following contributions:

1) To the best of our knowledge, this is the first review of the existing works on the hybrid analysis approach and an analysis of their pros and cons.
2) This work presents the significance of hybrid analysis

over static analysis and dynamic analysis by analyzing their weaknesses and limitations.
3) This work provokes a discussion about the challenges, opportunities and future directions of hybrid analysis.

II. Background

A. Android Malware

Android malware is an application running on the Android OS that implicitly or explicitly performs malicious activities. It includes viruses, worms, ransomware, spyware, and other malicious applications. It tends to cause - disrupting normal functioning, taking access controls, leaking information, root exploitation, manipulating data, private content exposed, phishing, disruption of services, etc. [5].

Moreover, malware is growing exceedingly to keep pace with the immense growth of Android applications. In each month, on average almost 10 million new malware is introduced [9]. New malware is found in every 10 seconds [10]. The most alarming fact is that nowadays noxious malware authors also aware of the malware detection system and they use many novels and crafty evasion techniques to avoid detection.

B. Detection Techniques

Researchers generally analyze Android malware with the following three approaches: Static Analysis, Dynamic Analysis and Hybrid Analysis.

In static analysis, various static features are extracted from source code and meta-data. If the source code is not available, reverse engineering is applied to reproduce the source code. According to the static features, a detection model is built using machine learning techniques to classify Android malware. Researchers used Androguard, ApkTool, Appknox, DroidMat, etc. tools for static analysis. According to the existing research [11]–[15] in static analysis, the most used static features are as follows: Permissions, Intents, Instructions, Hardware Usage Analysis, Meta-data, Intents, API Calls, Intents, Suspicious Files, and Potentially Dangerous Functions and Methods.

Dynamic analysis deals with the dynamic features/behaviors of an application. To track the dynamic behaviors of an application, the application is to be run/executed in an emulated environment or on a real device. A detection model is also built here according to the dynamic features. Researchers used Droidbox, Marvin, Cuckoo Sandbox, AppsPlayground, DroidLogger, etc. tools for dynamic analysis. According to research [16]–[19], the most used dynamic features are: System Calls, Network Traffic, Running Services, File Operations, Network Operations, and Phone Events.

Hybrid Analysis incorporates both static and dynamic features for detecting Android malware. As it deals with both static and dynamic features, it is computationally more complex. Andrubis, AndroData, etc. are used by the researchers for hybrid analysis.

C. Limitations of Static and Dynamic Analysis

Static Analysis faces many troubles such as data obfuscation, control flow obfuscation, encryption, reflection, dynamically loaded code, repackaging, etc. [4] by the shrewd malware authors.

On the other hand, Dynamic Analysis also has some drawbacks. To evade dynamic analysis, the anti-analysis technique is used frequently by malware authors to detect virtual machines or emulated environments. If the application detects emulated environments in advance, they will act as a benign application. By doing so, dynamic analysis might fail to detect Android malware. Besides, malware authors use mimicry, data obfuscation, misleading information flows and function indirections, etc. to evade dynamic analysis [4]. The biggest weakness of dynamic analysis is limited code coverage: covering all paths is not feasible when investigating the dynamic behavior of an application.

III. Hybrid Analysis using Machine Learning

Hybrid analysis integrates both static and dynamic features for effectiveness. Firstly, it seeks to extract the static and dynamic features of Android applications. After that, those extracted static and dynamic features are combined to build a detection model. Finally, according to the static and dynamic features, a detection model is built using machine learning techniques to classify Android malware. Figure 1 depicts the common methodology of that approach.

By incorporating static and dynamic approaches into a common ground, hybrid analysis leads to more complexity in Android malware detection. The detection process is more likely to take more time and effort. Though the hybrid approach might be more effective for Android malware detection than the static or dynamic approach, accomplishing a viable malware detection technique is challenging.

As the hybrid approach is the combination of static and dynamic approaches, this approach can overcome the individual weakness as well as can accumulate the advantages of them. Thereby, the hybrid approach strengthens the detection process with the cost of time and complexity. Hybrid methods can also increase robustness, monitor edited apps, increase code coverage and find vulnerabilities [4].

IV. Methodology

To build up a systematic literature review, we have followed a state-of-the-art guideline presented by Kitchenham and Stuart [20]. According to the guideline, Developing a review protocol is compulsory to shape a systematic review. The review protocol includes:

- The rationale for the review
- Research questions
- Search strategy
- Study selection criteria
- Study selection procedures
- Study quality assessment
- Data extraction
- Data synthesis

It is not mandatory to follow all the steps given by the protocol. Only relevant steps should be done, other steps could be overlooked. Details of each step taken in our work are described in the following subsections.

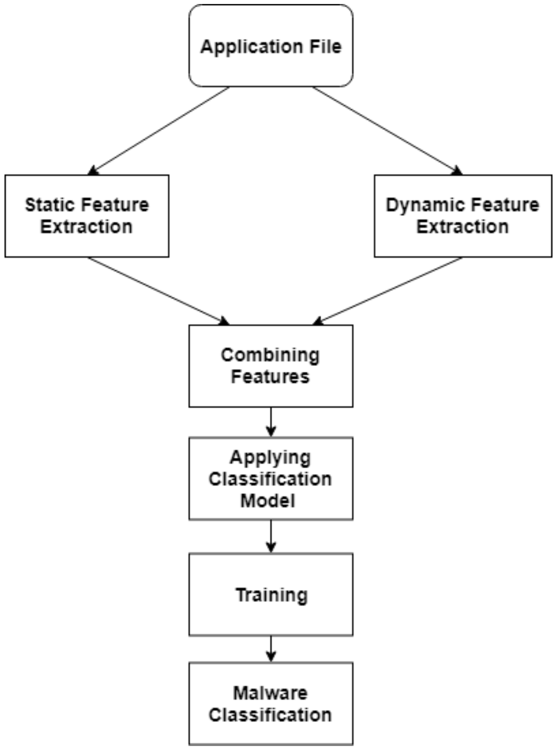

Fig. 1. Hybrid Analysis using Machine Learning

A. The Rationale for the Review

Hybrid analysis using machine learning for Android malware detection is a promising research domain because the weaknesses of static and dynamic analysis approach have lessened their effectiveness. Though there are a few researches so far in this domain, the potentiality of this domain needs a brief review of the existing literature.

B. Research Questions

Identifying the research questions is the most key part of a systematic review. To present a systematic review, we have identified the following research questions:

1) What are the static and dynamic features used in hybrid analysis using machine learning?
2) What are the most common dataset sources of the existing literature?
3) Which machine learning algorithms are most frequently used in the existing researches?
4) Which evaluation metrics are most widely used in the existing literature?
5) What are the evaluation results of the existing researches?
6) What are the limitations of the existing literature?

C. Study Selection Criteria

From the search results, we have defined the inclusion and exclusion criteria as follows:

1) Inclusion Criteria:

- Journal, Conference Proceedings of hybrid analysis using machine learning
- Date (year) of publication: 2012-2019
- Most recent version of the research paper

2) Exclusion Criteria:

- Research that uses - hybrid keyword, but not directed to the hybrid analysis
- Research that incorporates hybrid analysis, but not employing machine learning techniques
- Research that lacks a well-defined methodology and unambiguous contributions

D. Study Quality Assessment

We have scrutinized the selected papers for bias, internal validity, and external validity. Though there is no consensus about the interpretation of - quality, the CRD Guidelines [21] and the Cochrane Reviewers Handbook [22] suggest that quality correlates insofar as the study minimizes bias and maximizes internal and external validity [20].

E. Data Extraction

To keep track of the extracted data/information, a data extraction form have been maintained. Table I depicts the contents of the form.

TABLE I
DATA EXTRACTION FORM

Data Item	Value	Remarks
Paper ID		
Paper Name		
Author Name		
Publishing Year		
Publisher		
Journal/Conference Name		
Research Question 1		
Research Question 2		
Research Question 3		
Research Question 4		
Research Question 5		

V. SYSTEMATIC LITERATURE REVIEW

In the following section, we have resolved the research questions and presented an inclusive systematic review of the consequential researches in hybrid analysis. At first, the first four research questions are resolved according to the existing researches. Then each of the existing researches on hybrid analysis is discussed briefly; the last two research questions are sorted out in that part.

Permissions and API Calls as static features and System Calls as dynamic features are most frequently used in the existing researches.

The most common dataset according to the existing researches are Drebin and Android Malware Genome Project. Besides, most researches use the Google Play Store and local app stores to collect benign applications. ContagioDump, VirusTotal, VirusShare, etc. sources are also used for malware samples.

Support Vector Machine (SVM) is the most frequently used machine learning algorithm in the existing research. Besides, Naive Bayes, Random Forest, J48, Logistic Regression, etc. are also common in the existing researches.

Accuracy, True Positive Rate (TPR), False Positive Rate are the most common evaluation metrics according to the existing researches.

One of the state-of-the-art work in hybrid analysis, Marvin [23] employed a lot of static and dynamic features to detect Android malware. It extracted Permissions, Intents, Suspicious Files, API Calls, Developers Certificate, etc. as static features and File Operations, Network Operations, Phone Events, Dynamically Loaded Code, etc. as dynamic features. It used SVM and Linear Classifiers (Regularized Logistic Regression) to build a detection model where Linear Classifiers can detect more accurately but SVM is faster comparatively. For labeled test data, Marvins performance is sound enough as its accuracy to detect malware is 98.24 % with less than 0.04% false-positive rate. But for previously unseen malware, its accuracy is close to 90%. Besides, to avoid the obsolescence of its classification model in the future, it presented a retraining strategy. Though Marvin considers a lot of features, it overlooked system-level events such as System Calls: an integral part of the behavioral aspects (dynamic features).

Mobile-SandBox [24] used Permissions, Services, Receivers, Intents, Potentially dangerous functions, and methods as static features and investigated Native Code (Native API Calls) and Network Traffic as dynamic features to classify malware. It lacks in performance as it did not provide any solid performance metrics.

Samadroid [25] presented an on-device malware detection architecture which ensures the resource efficiency by reducing memory overhead of local devices. It used a subset of Drebin's [11] features (6 out of 8) as static features and 10 predefined System Calls as dynamic features. Its accuracy is almost 98% with a false positive rate of 0.1%. Though it incorporated System Call into its feature space and outperform Drebin [11], it used the old dataset. Thereby it might fail to fight against recent malware as malware behavior changes frequently over time. It also overlooked any additional dynamic features.

Kapratwar et al. [26] used Permissions and System Calls for hybrid analysis. Its performance (AUC) is significantly better for static features in comparison with dynamic features. But it used a small (200 apps) and old dataset and overlooked other static and dynamic features.

Hadm [27] incorporated Deep Neural Network for feature extraction from a set of static and dynamic features. It exhibited that combining advanced features derived by deep learning with the original static and dynamic features provides consequential returns. It achieved 94.7% accuracy with a false positive rate of 1.8% while with the original features the best accuracy is 93.5%. An improvement of 1.2% with the cost of complexity.

Dhanya et al. [28] used Permissions as static and API Calls as a dynamic feature. Separability assessment Criteria is used for feature selection in this research. Using the 77 selected features and four different machine learning algorithms (Naive Bayes, SVM, J48 & Random Forest), they evaluated their work. Their performance regarding F-measure, precision, and recall is dubitable as they used Drebin, an outdated and limited dataset. Besides they did not consider any other features except Permissions and API Calls.

Liu et al. [29] proposed a hybrid malware detecting scheme for Android where Permissions and API Calls are used as static features and System Calls used as dynamic features. Their scheme's detection accuracy is from 93.33% to 99.28% according to experimental results. Though they considered only a small feature-set and their dataset is also limited.

Table II depicts the literature overview of hybrid analysis using machine learning.

VI. Discussion

In this section, we have pointed out the opportunities, challenges, limitations and future directions of hybrid analysis.

A. *Lack of Research*

As mentioned before, there is not enough research in hybrid analysis, though it is a promising and effective approach in Android malware detection. Researchers have to emphasize in this regard. A lot of opportunities and research directions are available right now. Researchers' enthusiastic focus on this field would have been beneficial to fight against the escalating malware authors community, more research work is essential.

B. *Dataset Inadequacy*

Malware is growing enormously, but there does not exist any up-to-date dataset for the researchers. Previously stated, almost 10 million new malware are found each month [18]. But most of the dataset used in research is dated and obsolete nowadays. Thereby, their performance in Android malware detection is doubtful considering the vast population of the new malware. Dataset inadequacy is a vital factor as the dataset is responsible for the evaluation of any research. So, the Android malware dataset has to be updated on a regular basis to assure the effectiveness of the new research and to justify the feasibility of the existing research.

C. *Exploring New Feature*

Most of the existing research dealt with some common features such as Permissions, API Calls, System Calls, File Operations, Network Operations etc. But it would be possible that there exists more distinguishable features to detect Android malware. In this regard, Talha et al. [30] revealed many unknown characteristics of Android malware, however it did not integrate any machine learning technique to detect Android malware. They revealed that over-privileged permissions is one of the characteristics of malware. Besides, they uncovered that malware's average number of incoming and outgoing connections, the average size of download and upload, the average number of INTERNET_CLOSE action are distinguishable features with respect to benign applications. Looking for more discernible features might create new opportunities in Android malware detection.

TABLE II
SYSTEMATIC LITERATURE OVERVIEW OF HYBRID ANALYSIS USING MACHINE LEARNING

Ref.	Publishing Year	Static Features	Dynamic Features	Dataset Source	Dataset Size	Algorithms	Metrics	Values	Limitation
Marvin [23]	2015	Permissions, Intents, Suspicious Files, API Calls, Developer's Certificate etc.	File Operations, Network Operations, Phone Events, Dynamically Loaded Code etc.	Google Play Store, VirusTotal, Genome Project, Contagio	150,000 apps (135,000 benign, 15,000 malware)	SVM and Linear Classifier (Regularized Logistic Regression)	Accuracy, FPR	98.24%, <0.04%	Overlooking system-level events such as System Calls
Mobile-SandBox [24]	2013	Permissions, Services, Receivers, Intents, Potentially Dangerous Functions and Methods	Native Code (Native API Calls) and Network Traffic	Asian markets and Google Play Store	40,000 apps				Lacking in performance as no solid performance metrics given
Samadroid [25]	2018	Permissions, API Calls, Intents, App Components	System Calls (10)	Drebin	5,560 apps	SVM, Naive Bayes, Decision Tree and Random Forest	Accuracy, TPR, FPR	91.6%∼ 98.97%, 81.1%∼ 98.5%, 0.03%∼ 7.8%	Overlooking many dynamic features; using limited and old dataset
Kapratwar et al. [26]	2017	Permissions	System Calls	Google Play Store, VirusTotal, Drebin	200 apps (103 benign, 97 malware)	Nave Bayes, J48 & Random Forest, Simple Logistic, IBk	AUC	0.5844∼ 0.9660	Overlooking many static and dynamic features; using small and old dataset
Hadm [27]	2016	Permissions, API Calls, Intents	System Call Sequences	Google Play and VirusShare	5888 apps (4002 benign, 1886 malware)	Deep Neural Network, SVM, Hierarchical Multiple Kernel Learning	Accuracy, FPR	94.7%, 1.8%	Higher complexity with respect to accuracy gains
Dhanya et al. [28]	2019	Permissions	API Calls	Drebin	400 apps (200 benign, 200 malware)	Nave Bayes, SVM, J48 & Random Forest	F-measure, Precision, Recall	0.71%∼ 0.975, 74.7%∼ 97.6%, 72.5%∼ 97.5%	Using limited and old dataset; considering few features
Liu et al. [29]	2016	Permissions	System Calls	Gnome Project, Wandoujia App Market	1000 apps (1000 benign, 1000 malware)	SVM, KNN	ACC, TPR, FPR	93.33%∼ 99.28%, 94.59%∼ 99.47%, 0.20%∼ 11.01%	Using limited dataset, considering few features

D. Better Performance

Hybrid analysis exhibits better performance on average than the typical static and dynamic approaches and induces a lot of opportunities. By taking those opportunities and overcoming the challenges ahead, hybrid analysis would be a vanguard for Android malware detection in the future.

E. New Malware Family

As existing malware behavior is decoded by the existing tool or research outcome, malware authors update existing malware families and create new malware families frequently to evade detection. Their behaviors are mostly unfamiliar to the typical detection system. They try to trick existing detection systems by introducing new behavior as well as exhibiting benign behavior. So researchers should consider this issue carefully to ensure security. *How do we detect new malware families effectively* - would be a promising research question.

F. Reducing Complexity

Since the hybrid approach combines static and dynamic approaches, its overall complexity is higher with respect to time, cost and effort. *How do we reduce the complexity of hybrid analysis* - would be a potential direction for future researchers.

VII. Conclusion

Detecting Android malware effectively and feasibly is one of the crucial challenges of this fast-growing digital world. The hybrid analysis technique has the capability and can offer a sound direction in this field. By exploring this field, researchers have already published several research. This work tends to highlight those research by providing a thorough and systematic review of them. It encompasses the static features, dynamic features, dataset, algorithms, metrics considered in those research. It also focuses on the individual strengths and limitations of them. Besides it points out the specific challenges, limitations and future directions in the hybrid analysis technique. By doing so, this research seeks to contribute to academia as well as raise concern for Android mobile application security.

References

[1] N. G., "Android: Market share other stats [infographic]," Jul 2019.

[2] B. Popper, "Google announces over 2 billion monthly active devices on android," May 2017.

[3] C. Lueg, "Cyber attacks on android devices on the rise," Nov 2018.

[4] K. Tam, A. Feizollah, N. B. Anuar, R. Salleh, and L. Cavallaro, "The evolution of android malware and android analysis techniques," *ACM Computing Surveys (CSUR)*, vol. 49, no. 4, p. 76, 2017.

[5] A. Qamar, A. Karim, and V. Chang, "Mobile malware attacks: Review, taxonomy future directions," *Future Generation Computer Systems*, vol. 97, 03 2019.

[6] B. Baskaran and A. Ralescu, "A study of android malware detection techniques and machine learning," 2016.

[7] A. Naway and Y. Li, "A review on the use of deep learning in android malware detection," *arXiv preprint arXiv:1812.10360*, 2018.

[8] A. Feizollah, N. B. Anuar, R. Salleh, and A. W. A. Wahab, "A review on feature selection in mobile malware detection," *Digital investigation*, vol. 13, pp. 22–37, 2015.

[9] "Malware statistics trends report: Av-test," Apr 2019.

[10] "Android malware threat profile," 2018.

[11] D. Arp, M. Spreitzenbarth, M. Hubner, H. Gascon, K. Rieck, and C. Siemens, "Drebin: Effective and explainable detection of android malware in your pocket.," in *Ndss*, vol. 14, pp. 23–26, 2014.

[12] S. Y. Yerima, S. Sezer, G. McWilliams, and I. Muttik, "A new android malware detection approach using bayesian classification," in *2013 IEEE 27th international conference on advanced information networking and applications (AINA)*, pp. 121–128, IEEE, 2013.

[13] E. Mariconti, L. Onwuzurike, P. Andriotis, E. De Cristofaro, G. Ross, and G. Stringhini, "Mamadroid: Detecting android malware by building markov chains of behavioral models," *arXiv preprint arXiv:1612.04433*, 2016.

[14] M. Ghorbanzadeh, Y. Chen, Z. Ma, T. C. Clancy, and R. McGwier, "A neural network approach to category validation of android applications," in *2013 International Conference on Computing, Networking and Communications (ICNC)*, pp. 740–744, IEEE, 2013.

[15] Q. Jerome, K. Allix, R. State, and T. Engel, "Using opcode-sequences to detect malicious android applications," in *2014 IEEE International Conference on Communications (ICC)*, pp. 914–919, IEEE, 2014.

[16] L. K. Yan and H. Yin, "Droidscope: Seamlessly reconstructing the {OS} and dalvik semantic views for dynamic android malware analysis," in *Presented as part of the 21st {USENIX} Security Symposium ({USENIX} Security 12)*, pp. 569–584, 2012.

[17] B. Amos, H. Turner, and J. White, "Applying machine learning classifiers to dynamic android malware detection at scale," in *2013 9th international wireless communications and mobile computing conference (IWCMC)*, pp. 1666–1671, IEEE, 2013.

[18] W.-C. Wu and S.-H. Hung, "Droiddolphin: a dynamic android malware detection framework using big data and machine learning," in *Proceedings of the 2014 Conference on Research in Adaptive and Convergent Systems*, pp. 247–252, ACM, 2014.

[19] W. Enck, P. Gilbert, S. Han, V. Tendulkar, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth, "Taintdroid: an information-flow tracking system for realtime privacy monitoring on smartphones," *ACM Transactions on Computer Systems (TOCS)*, vol. 32, no. 2, p. 5, 2014.

[20] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," 2007.

[21] K. S. Khan, G. Ter Riet, J. Glanville, A. J. Sowden, J. Kleijnen, *et al.*, *Undertaking systematic reviews of research on effectiveness: CRD's guidance for carrying out or commissioning reviews*. No. 4 (2n, NHS Centre for Reviews and Dissemination, 2001.

[22] J. P. Higgins and S. Green, *Cochrane handbook for systematic reviews of interventions*, vol. 4. John Wiley & Sons, 2011.

[23] M. Lindorfer, M. Neugschwandtner, and C. Platzer, "Marvin: Efficient and comprehensive mobile app classification through static and dynamic analysis," in *2015 IEEE 39th annual computer software and applications conference*, vol. 2, pp. 422–433, IEEE, 2015.

[24] M. Spreitzenbarth, F. Freiling, F. Echtler, T. Schreck, and J. Hoffmann, "Mobile-sandbox: having a deeper look into android applications," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, pp. 1808–1815, ACM, 2013.

[25] S. Arshad, M. A. Shah, A. Wahid, A. Mehmood, H. Song, and H. Yu, "Samadroid: a novel 3-level hybrid malware detection model for android operating system," *IEEE Access*, vol. 6, pp. 4321–4339, 2018.

[26] A. Kapratwar, F. Di Troia, and M. Stamp, "Static and dynamic analysis of android malware.," in *ICISSP*, pp. 653–662, 2017.

[27] L. Xu, D. Zhang, N. Jayasena, and J. Cavazos, "Hadm: Hybrid analysis for detection of malware," in *Proceedings of SAI Intelligent Systems Conference*, pp. 702–724, Springer, 2016.

[28] K. D. T. Gireesh Kumar, "Efficient android malware scanner using hybrid analysis," *International Journal of Recent Technology and Engineering(TM)*, vol. 7, pp. 76–80, 2019.

[29] Y. Liu, Y. Zhang, H. Li, and X. Chen, "A hybrid malware detecting scheme for mobile android applications," in *2016 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 155–156, IEEE, 2016.

[30] A. T. Kabakus and I. A. Dogru, "An in-depth analysis of android malware using hybrid techniques," *Digital Investigation*, vol. 24, pp. 25–33, 2018.