FUDMA Journal of Engineering and Technology
ISSN: 3092-9385
https://sites.google.com/fudutsinma.edu.ng/fjet
Volume 1, Issue 2, 2025, 678-684

Performance Assessment of Android Antimalware Applications: An Experimental Approach

Lukman MOHAMMED[1*], Victor O. WAZIRI[2], Ismaila IDRIS[3], Suleiman AHMAD[4]

[1*,2,3,4]Cyber Security Science Department, Federal University of Technology, Minna, Nigeria

[1*]mohammed.pg202319198@st.futminna.edu.ng, [2]victor.waziri@futminna.edu.ng, [3]ismi.idris@futminna.edu.ng, [4]ahmads@futminna.edu.ng

Abstract

The widespread use of Android phones has made the devices the primary target for malware authors. There are many commercial antimalware tools but they are not totally effective as android users still record high false positive rate. This research evaluates the performance of five common Android antimalware tools which include Kaspersky, BitDefender, Avira, Norton and McAfee against ten ransomware samples obtained from the AndMal2017 Android dataset. The experiment was conducted using Android Studio Emulator where each of the antimalware tool was tested to ascertain the detection rate, scan time and memory usage. Experimental results indicate that BitDefender achieved the highest detection accuracy of 90% with lowest scan time of 16.2 seconds and memory usage of 146.2MB. By providing quantitative benchmarks and an emulator-based testing framework, this research contributes practical insights for both academic and industry stakeholders in mobile cybersecurity.

Keywords: *Malware, Android, Ransomware, Antimalware, detection.*

1.0 Introduction

Android has become the world's dominant mobile operating system, powering billions of devices globally [1]. Its open-source nature and extensive ecosystem have made it vulnerable to different attacks by cybercriminals through the use of malicious applications. Over the past decade, attacks involving Android malware such as spyware, adware, ransomware and banking trojans have grown exponentially [2]. This rise highlights the importance of effective antimalware tools capable of protecting users against these evolving threats. Previous studies also suggest that as the complexity of android malware continues to increase, more robust detection techniques are required [3].

In response to the rising threat landscape, many commercial antimalware tools are employed for real-time protection, heuristic scanning and behavioral analysis to safeguard Android devices [4]. Many of these antimalware tools claim high detection rates and comprehensive coverage but independent evaluations revealed that there are inconsistencies in their actual performance. Factors such as delayed signature updates, excessive memory use and poor detection of emerging threats can reduce overall protection [5]. Hence, an experimental assessment of these tools is necessary to determine their real-world effectiveness.

This study aims to assess the performance of five popular android antimalware applications through a systematic experimental approach. By testing these commercial tools against diverse malware samples, the research measures their detection rate, scan time and memory usage. Unlike most prior studies that relied on synthetic or general malware samples, this research uniquely focuses on ransomware evaluation within a controlled emulator-based Android environment. The goal is to identify strengths and weaknesses of each tool and provide evidence-based insights that can guide users in selecting reliable protection while also informing developers and researchers about areas that need further improvement.

1.1 Problem Statement

Numerous antimalware applications are available online; however, most Android users in Nigeria face challenges in identifying the most suitable applications to prevent or minimise cyberattacks [6]. In the first quarter of 2025, Cyber security landscape threat by Kaspersky Security Network ranked Nigeria and South Africa as the fourth in the world as shown in Figure 1.

Received: 23/10/2025; Revised: 30/10/2025; Accepted: 30/10/2025; Published: 15/11/2025

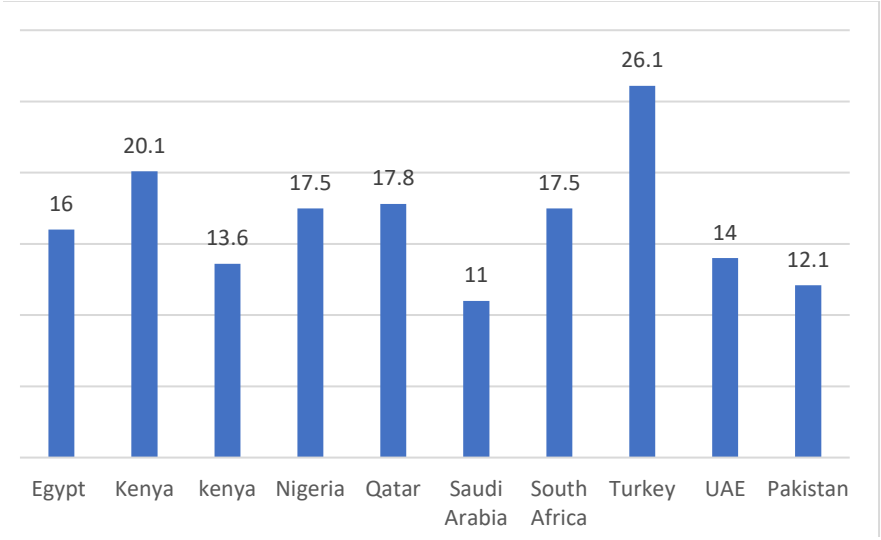

Figure 1: Cybersecurity Threat Landscape [7]

2.0 Literature Review

Arslan *et al.* [8] proposed a deep learning-based method that extracted the requested permission as features. The approach achieved an accuracy of 98.2% when compared with several traditional machine learning algorithms. Ma *et al.* [9] proposed a deep learning approach for android malware detection. Their model converted application programs to natural language sequences. They used bi-directional Long Short-Term Memory network BiLSTM for malware detection and achieved an accuracy of 97.22%. Saxe and Berlin [10] developed a deep neural network model for malware detection using binary program features extracted from executable files. Their model achieved 92 % accuracy but relied on static analysis and also lack of interpretability. It then becomes difficult to state the features that contributes to the detection. This therefore highlights the need for more comprehensive approach. Tobiyama *et al.* [11] proposed a deep learning approach for malware detection. The approach was based on static analysis of executable files where a stacked auto-encoder was used to feature extraction. Although the study demonstrated the potential of deep learning, it did not specifically focus on ransomware detection. Fang *et al.* [12] classified android applications by using a fusion algorithm based on multiple kernel learning. Features extracted from Dalvik Executable (DEX) file section were converted into RGB images and plain text and accuracy of 96% was achieved. Sahin *et al.* [13] classify android malware using multiple linear regression approach based on permission features which was applied to four datasets. They also used bagging to build the classifiers, achieving the highest accuracy of 98.53% on the second dataset when ensemble learning was used.

3.0 Methodology

3.1 Dataset and Environmental Setup

The experiment was conducted using ten ransomware samples extracted from the AndMal2017 android malware dataset. The samples include SVPENG, WANNA LOCKER, CHARGER, PLETOR, KOLER, LOCKERPIN, RANSOMBO, PORNDROID, SIMPLOCKER and JISUT. Five common antimalware applications which include Kaspersky, BitDefender, Avira, Norton and McAfee were installed and tested against the ten ransomware samples in a controlled environment using Android Studio Emulator with Android Virtual Device (AVD). The Emulator was configured to operate offline to prevent external network communication. All the installations, detections and interactions occurred within the emulator, which ran on a secure host computer with the following configuration: Windows 10 (64-bit), 4 GB RAM, Intel Core i3-2330 CPU @ 2.20 GHz, Hewlett-Packard Pavilion g6 Notebook PC.

3.2 Experimental Procedures

The following procedures were used to evaluate each antimalware application independently:

i) Install Android Studio on the host system
ii) Launch Android Studio and confirm that the Android Emulator is installed correctly.
iii) Create an Android Virtual Device (AVD)
iv) Launch the AVD from the Android Emulator to confirm successful booting
v) Install the five selected antimalware applications
vi) Take a clean snapshot of the AVD to maintain a baseline testing state.
vii) Install each ransomware sample within the AVD
viii) Observe each antimalware applications one after the other

Received: 23/10/2025; Revised: 30/10/2025; Accepted: 30/10/2025; Published: 15/11/2025

ix) Record the detection results, scan times and memory usage for each test
x) Uninstall the previous ransomware sample and restart the AVD
xi) Repeat steps 7–10 for the remaining ransomware samples until all ten have been tested.

Figure 2 illustrates the overall experimental procedures used to evaluate the antimalware tools.

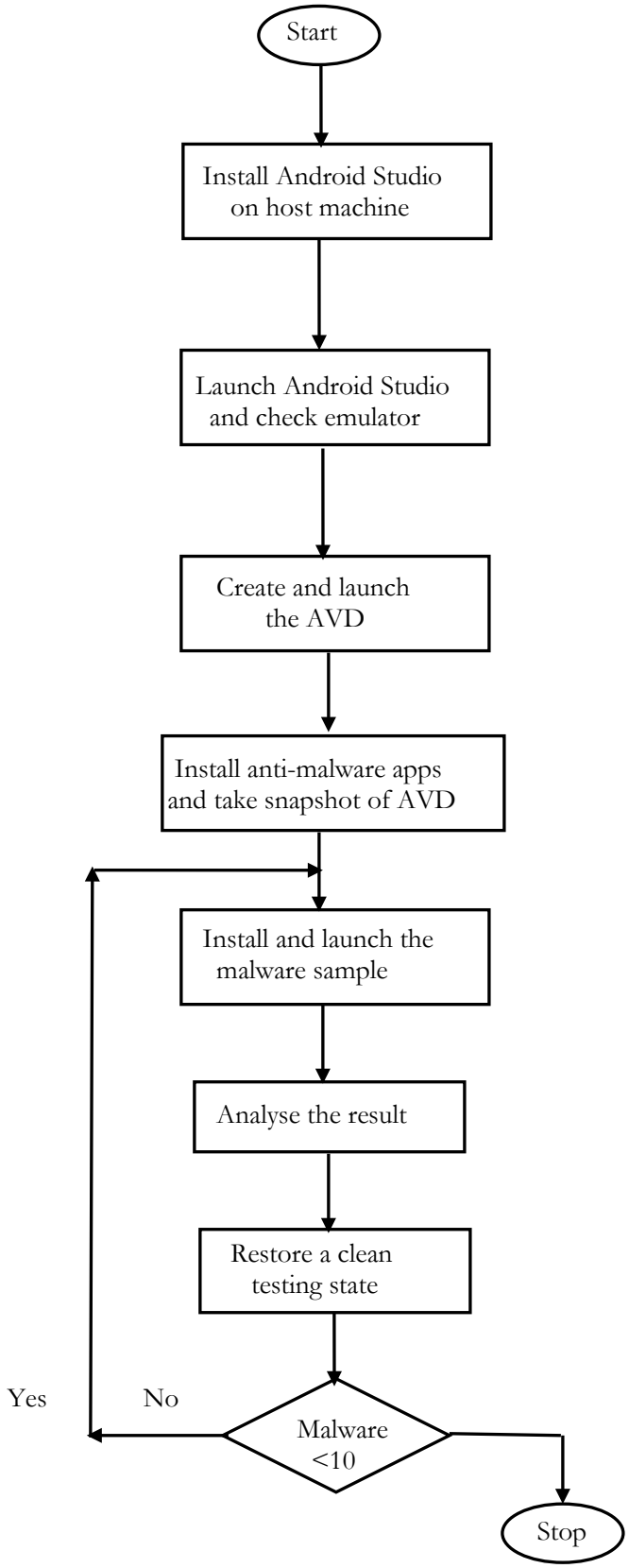

Figure 2: Experimental procedure flowchart

Received: 23/10/2025; Revised: 30/10/2025; Accepted: 30/10/2025; Published: 15/11/2025

3.3 Performance Metrics and Mathematical Formulation

The following three key metrics were employed to evaluate the performance of the selected antimalware tools:

i. Detection Rate (DR): This is the proportion of the ransomware samples correctly detected by the antimalware tools.
It is calculated as: $DR = (N_e / N_t) \times 100$ (1)
where N_e is the number of detected ransomware samples and N_t is the total tested samples.

ii. Average Scan Time (T_{avg}): This is the average time required by a tool to scan a ransomware sample. It represents the tool's efficiency.
$(T_{avg}) = \Sigma T_i/n$ (2)
where T_i = scan time for sample i and n = number of samples.

iii. Average Memory Usage (M_{avg}): This represents the average memory consumed during the scanning process, indicating the tool's resource demand.
$M_{avg} = \Sigma M_i/n$ (3)
where M_i = memory usage for sample i and n = number of samples.

3.4 Summary

The study evaluated the performance of selected antimalware tools through a structured experimental approach. Performance was quantified using detection rate, scan time and memory usage, with metrics calculated mathematically to ensure objectivity. This methodology provides a systematic framework for comparing the tools' effectiveness, efficiency and resource utilization.

4.0 Results Presentation

The detection result obtained from the experiment, scan time (in seconds) and memory usage (in megabyte) are presented in Tables 1 – 5. The analysis shows that the performance of the five antimalware applications varied in detection accuracy, scan time and memory usage. BitDefender demonstrated the best overall performance, achieving the highest detection rate, lowest scan time and memory usage. The superior detection consistency observed in BitDefender aligns with the adaptive feature-based mechanisms seen in deep learning models, suggesting that integrating machine learning into commercial tools could improve detection precision.

Table 1: Kaspersky

Malware Type	Detection Result	Scan Time (sec)	Memory Usage (MB)
SVPENG	Detected	18	210
WANNA LOCKER	Detected	21	230
CHARGER	Detected	17	190
PLETOR	Not Detected	24	205
KOLER	Detected	19	215
LOCKERPIN	Not Detected	21	183
RANSOMBO	Detected	22	221
PORNDROID	Detected	18	244
JISUT	Detected	23	217
SIMPLOCKER	Detected	21	228

Table 2: BitDefender

Malware Type	Detection Result	Scan Time (sec)	Memory Usage (MB)
SVPENG	Detected	12	122
WANNA LOCKER	Detected	18	174
CHARGER	Detected	19	156
PLETOR	Detected	17	137
KOLER	Not Detected	19	164
LOCKERPIN	Detected	11	104
RANSOMBO	Detected	17	139
PORNDROID	Detected	18	125
JISUT	Detected	18	178
SIMPLOCKER	Detected	13	163

Received: 23/10/2025; Revised: 30/10/2025; Accepted: 30/10/2025; Published: 15/11/2025

Table 3: Avira

Malware Type	Detection Result	Scan Time (sec)	Memory Usage (MB)
SVPENG	Detected	27	222
WANNA LOCKER	Not Detected	33	280
CHARGER	Not Detected	21	294
PLETOR	Detected	32	267
KOLER	Not Detected	28	245
LOCKERPIN	Detected	26	292
RANSOMBO	Detected	22	257
PORNDROID	Detected	34	259
JISUT	Detected	29	278
SIMPLOCKER	Not Detected	31	267

Table 4: Norton

Malware Type	Detection Result	Scan Time (sec)	Memory Usage (MB)
SVPENG	Detected	26	267
WANNA LOCKER	Detected	19	207
CHARGER	Detected	28	259
PLETOR	Not Detected	23	288
KOLER	Detected	27	246
LOCKERPIN	Not Detected	24	286
RANSOMBO	Detected	26	249
PORNDROID	Detected	21	289
JISUT	Detected	27	223
SIMPLOCKER	Not Detected	29	241

Table 5: McAfee

Malware Type	Detection Result	Scan Time (sec)	Memory Usage (MB)
SVPENG	Detected	33	307
WANNA LOCKER	Not Detected	29	293
CHARGER	Detected	25	248
PLETOR	Not Detected	31	289
KOLER	Detected	32	320
LOCKERPIN	Detected	27	278
RANSOMBO	Detected	25	297
PORNDROID	Detected	33	315
JISUT	Not Detected	32	322
SIMPLOCKER	Detected	28	267

The metrics values in Figures 3, 4 and 5 were derived from the mathematical formulation, computed for each tool under identical test conditions to ensure a fair comparison. As shown in Figure 3, BitDefender outperformed other antimalware tools by detecting nine out of the ten ransomware samples. In contrast, Avira ranked the lowest, detecting only six ransomware samples. This reinforces BitDefender's strong detection ability and consistency across the tested ransomware families. BitDefender achieved the lowest average scan duration of 16.2 seconds as shown in Figure 4, indicating a faster response and reduced device lag during operations. The analysis of average memory usage revealed that BitDefender was the most efficient using only 146.2 MB of memory, as shown in Figure 5.

Received: 23/10/2025; Revised: 30/10/2025; Accepted: 30/10/2025; Published: 15/11/2025

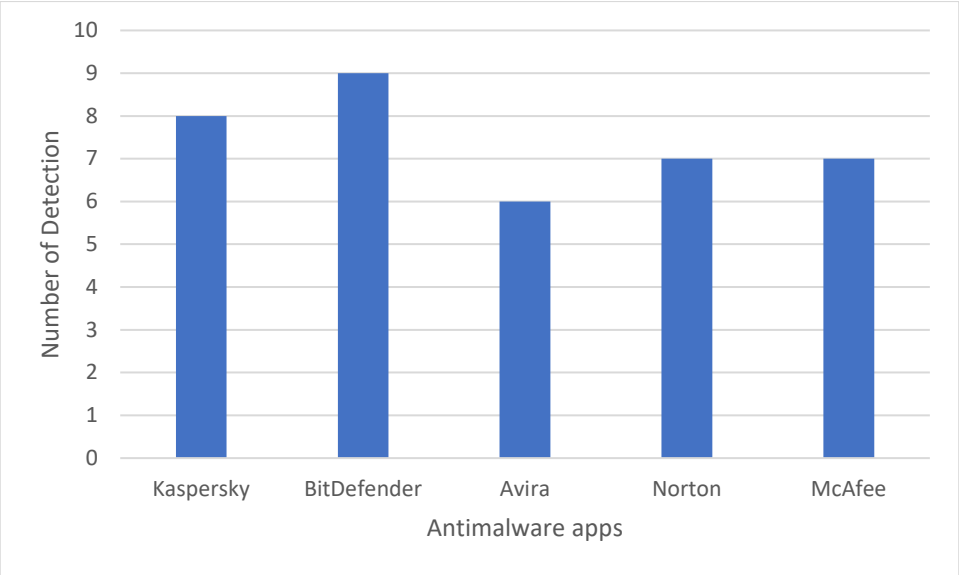

Figure 3: Malware Detection Performance

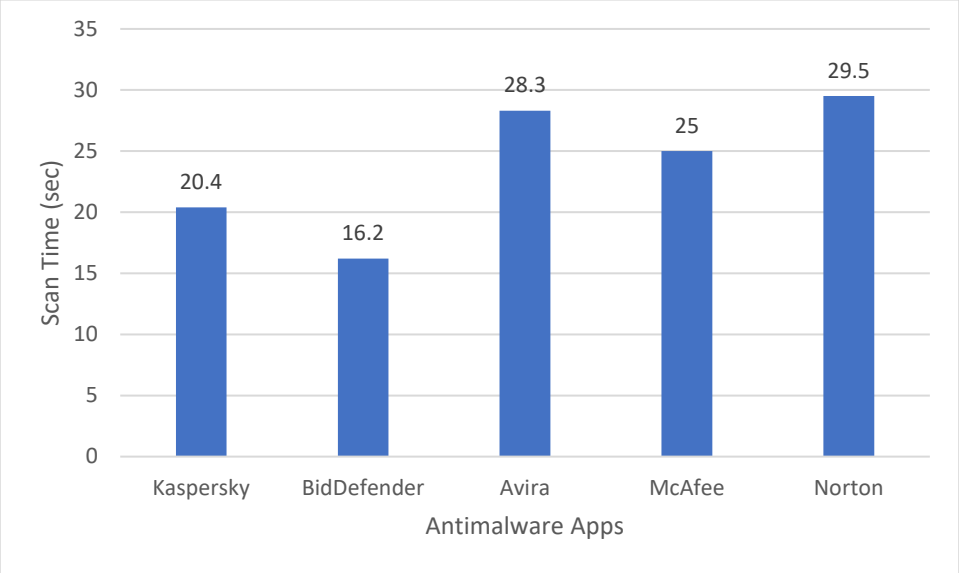

Figure 4: Average Scan Time

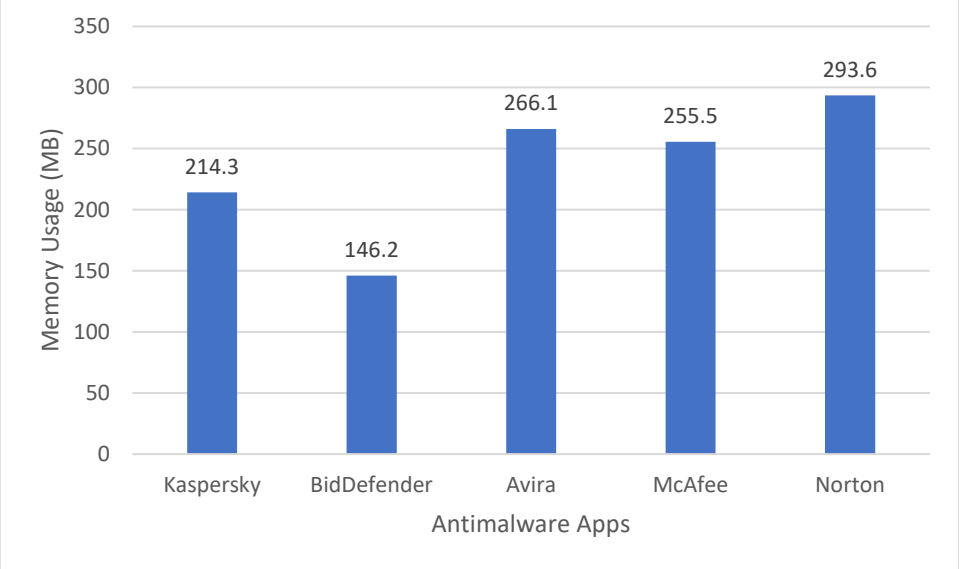

Figure 5: Average Memory Usage

Received: 23/10/2025; Revised: 30/10/2025; Accepted: 30/10/2025; Published: 15/11/2025

5.0 Conclusion

The performance assessment of the selected android antimalware tools demonstrates that not all commercial android antimalware applications provide the same level of protection. Among the five tested tools, BitDefender produced the best overall results, combining a high detection rate with the shortest scan time and lowest memory usage. Kaspersky and Norton offered balanced performance, whereas Avira and McAfee lagged in efficiency, consuming considerably more system resources. This study contributes quantitative benchmarks for the performance of five common Android antimalware tools in detecting ransomware within a controlled environment. The metrics provide reference values for future research on Android security evaluation. Future research could extend this study by using a larger dataset and benchmarking these commercial tools against deep learning-based detection frameworks to measure comparative performance improvements.

References

[1] StatCounter, "Mobile Operating System Market Share Worldwide (Sept. 2024–Sept. 2025)," *StatCounter Global Stats*, 2025. [Online]. Available: https://gs.statcounter.com/os-market-share/mobile/worldwide

[2] AV-TEST Institute, "Malware Statistics & Trends Report," *AV-TEST*, 2025. [Online]. Available: https://www.av-test.org/en/statistics/malware/

[3] A. Prasad, "AndroMD: An Android malware detection framework based on machine learning," *Heliyon*, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2590123025031068

[4] AV-Comparatives, "Mobile Security Review 2024," *AV-Comparatives*, 2024. [Online]. Available: https://www.av-comparatives.org/tests/mobile-security-review-2024/

[5] AV-TEST Institute, "Product Review: Google Play Protect (January 2024)," *AV-TEST*, 2024. [Online]. Available: https://www.av-test.org/en/antivirus/mobile-devices/android/january-2024/google-play-protect-39.2-243108/

[6] Techeconomy, "Nigeria Now Ranks Third on Mobile Malware Attacks from Fifth in 2017," *Techeconomy.ng*, Mar. 7, 2019. [Online]. Available: https://techeconomy.ng/2019/03/07/nigeria-now-ranks-third-on-mobile-malware-attacks-from-fifth-in-2017/

[7] Kaspersky, "State of Ransomware in 2025," *Securelist*, 2025. [Online]. Available: https://securelist.com/state-of-ransomware-in-2025/116475

[8] R. S. Arslan, "AndroAnalyzer: Android malicious software detection based on deep learning," *PeerJ Comput. Sci.*, 2021. [Online]. Available: https://doi.org/10.7717/peerj-cs.430

[9] Z. Ma, H. Ge, Z. Wang, Y. Liu, and X. Liu, "Droidetec: Android malware detection and malicious code localisation through deep learning," *arXiv preprint*, arXiv:2002.03594, 2020. [Online]. Available: https://arxiv.org/abs/2002.03594

[10] J. Saxe and K. Berlin, "Deep neural network-based malware detection using two-dimensional binary program features," in *Proc. 10th Int. Conf. Malicious and Unwanted Software (MALWARE)*, 2015, pp. 11–20.

[11] S. Tobiyama, Y. Yamaguchi, H. Shimada, T. Ikuse, and T. Yagi, "Malware detection with deep neural network using process behaviour," in *Proc. IEEE 40th Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, Vol. 2, 2016, pp. 577–582.

[12] Y. Fang, Y. Gao, F. Jing, and L. Zhang, "Android malware familial classification based on DEX file section features," *IEEE Access*, vol. 8, pp. 10,614–10,627, 2020, doi: 10.1109/ACCESS.2020.2965646.

[13] D. O. Sahin, S. Akleylek, and E. Kilic, "LinRegDroid: Detection of Android malware using multiple linear regression model-based classifiers," *IEEE Access*, vol. 10, pp. 14,246–14,259, 2022, doi: 10.1109/ACCESS.2022.3146363.

Received: 23/10/2025; Revised: 30/10/2025; Accepted: 30/10/2025; Published: 15/11/2025