International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

CYBER THREAT INTELLIGENCE FOR ANDROID ECOSYSTEMS: A MATHEMATICAL FRAMEWORK FOR ADVANCED DETECTION OF HIDDEN MALICIOUS ATTACKS IN MOBILE APPLICATIONS

Arun Singh Thakur[1], Kireet Muppavaram[2]

Department of CSE

GITAM DEEMED To Be University, Hyderabad, Telangana, India

E-mail: arunsingh184@gmail.com

Department of CSE

GITAM DEEMED To Be University, Hyderabad, Telangana, India

E-mail: kireet04@gmail.com

Corresponding author: Kireet Muppavaram

Abstract

Android mobile application has emerged among the best targets of the advanced cyber threats which aim at evading their use through the sophisticated evasion techniques so as not to be detected within the walls of the applications that seem to be legal. This paper proposes an elaborate mathematical model of cyber threat intelligence in detecting the obfuscated malicious attacks within the Android environment. We discuss some of the cyber-attack vectors including code obfuscation, dynamic loading and permission abuse where bad code can bypass traditional cyber security controls. To tackle these cyber crimes, two mathematically based algorithms are suggested which can be applied in combating these crimes, namely Permission Anomaly Detection Algorithm (PADA) and Mathematical Behavioral Analysis Algorithm (MBAA). In a more empirical sense, using our integrated mechanism in detecting cyber threats in the environment of 1,000 Android applications, our detection percentage on different cyber threat scenario is a 94.2 with a false positive of only 3.1 percent. Its outcomes are significant in regards to mathematical intelligence of cyber threats and how to create more resilient security models of mobile application environments which happen to be faced with the problem of cyber-attacks on Android platforms.

Keywords- Cyber threats, Cyber-attacks,Android ,Anomaly Detection, Cyber Security.

1. Introduction

Android mobile operating system has emerged as the top platform with over 2.5 billion working mobile devices all over the world by 2024[1][2]. Such a widespread acceptance has been a major attraction of these Android applications to the targeted eyes of such internet criminals whose intentions are to make money, conduct data stealing, and the undeserving accessibility to hitherto privacy of information. Current dynamics of cyber-threats[3] has made the

Received: July 25, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

traditional malware detection technologies the signature-based detection process, which forms the core of most detection technology to become ineffective in detection of the advanced cyber-threats that are sophisticated in their evasion mechanism. It is rather an important cyber threat feature to have an Android application that conceals shady malicious attacks[4]. Compared to the conventional malware that has a clear malicious system, these cyber-attacks have been designed to operate in an irregular fashion within authorized apps. They usually masquerade as harmless demand functionality but in the background perform illicit acts in the sense of data theft, unauthorized traffic over the network or the acquisition of privileges.

The issue of hidden cyber attack is surmounted by several considerations that pose challenge in mobility of the malady. First, Android apps are dynamic and hence their code can be changed at run time and they could even be loaded during run time therefore the modern method used by attackers may not be completely avoided by the analysis of their program. Second, one can get potential vulnerability in Android during the exploitation of the following cyber-attack techniques, such as the permission abuse and escalation. Third, there are cyber attackers who employ code obfuscation and anti-analysis where the malicious patterns in such instances are difficult to detect by means of conventional cyber analysis functions.

In order to address the above issues, in the current paper, a proposed cyber threat intelligence system has been introduced to identify the malicious covert attacks in Android applications. We suppose that the ensemble of such approach to analysis, such as dynamic behavior monitoring, and machine learning technique would allow finding suspicious patterns and therefore, determine when lurking cyber threats occur. Two new algorithms are proposed with specific purpose of dealing with these cyber-attacks by providing reasonable detection, and low false positive rates.

The central contributions of the current cybersecurity research are the following: (1) taxonomic classification of the covert attack types of Android applications, (2) a series of designed cyber threat detection algorithms with the mathematical foundation, (3) empirical evaluation of the properness of the attained cyber defense system, and (4) some practical proposals to upgrade the advances of Android application cybersecurity to resisting the existing threats of new origin.

2. Related Work

Massimo Ficco [5] an ensemble-based malware-detection system where a variety of both generic as well as specialized detectors is combined to increase robustness against evasion measures employed by contemporary malware. Instead, they essentially integrate these detectors strategically and by so doing do this to boost the unpredictability and resiliency of detection and this is particularly notable when it comes to the detection of unknown malware families. One of its contributions is the alpha-count mechanism, evaluating the effects of variations of the observation time window on detection speed and accuracy. A survey of experiments on an open-source sandboxed and Android Something smartphone setting on

Received: July 25, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

multiple malware datasets reveals a trade-off between performance, training overhead and detection latency. The presented system has increasing performance compared to the current ensemble detectors without requiring continuous retraining.

Arora [6] suggested a new detection model, the so-called Perm Pair, that constructs and compares the malware and regular models based on the permission pairs disclosed in the manifest file of an application. As ought to be evaluated, the suggested scheme is adequate in malicious sample detection, which provides an accuracy of 95.44 per cent compared to other comparable approaches and desired mobile anti-malware apps. Besides this we also provided a useful edge elimination algorithm as well that cut-off 7 percent of unwanted edges in the malware graph and 41 percent in the normal graph.

MulFC[7] , which works with the two key pitfalls of a supervised algorithm, namely the need to use a large labeled dataset and the inability to detect unknown malware families. MulFC operates on decompiled malware and retrieves a number of features- manifest, API, and opcode. The dosing of unnecessary opcode capabilities keeps the cost of the computation at a minimum. The malware similarity is computed by using Jaccard index to produce a malware network, afterward an InfoMap clustering is carried out to merge similar samples. The unknown malware families can be discovered by MulFC without labeled data. MulFC emerges best with regard to any metric of assessment based on a state-of-the-art methods on two databases due to its high quality of clustering measured as NMI (0.810), ARI (0.576), FMI (0.620), and V-measure (0.805).

Kim et al [8] highlighted a multimodal deep learning approach that is to be utilized in the Android malware detection on the basis of the combination of existence- and similarity-based types of features extractors. It is the first study that applies multimodal deep learning on the research. Having compared the model with 41,260 samples, it outperforms other existing applications of deep learning in regard to accuracy, feature utility, efficiency of updates, and it performs well when any of the several evidence measures is applied.

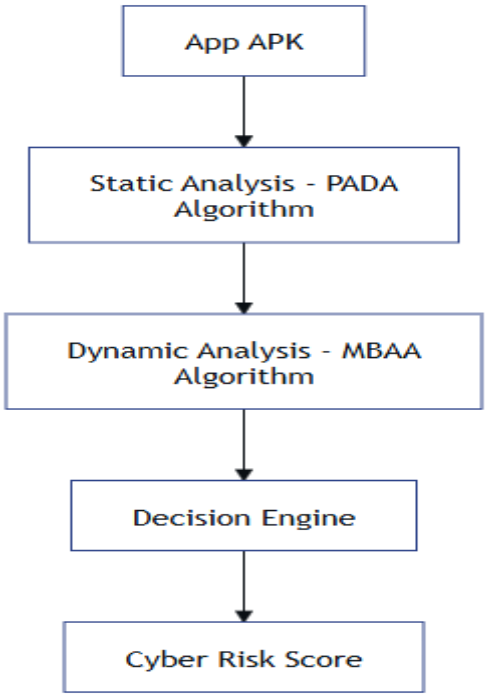
Thereafter, most of the malware shell detection techniques [9][10][11] are based on signature-based, heuristic and machine learning techniques.. Though dynamic, signature-based strategies are not quick to detect known threats, but they fail to recognize new kind of malware or malware that are obfuscated. Heuristic approaches[12] in reasoning contemplate on an assumption of behavior or code patterns that tends to be at a cost of trying out high false positives. At least, the model based on supervised learning requires large amounts of labeled data and can struggle dismal with new families of malware that have never been seen before. Besides that, many generic approaches are not dynamic, and they need retraining whenever there are new threats[13][14]. Such shortcomings are indicative of the need to have stronger, flexible and smarter detection systems because they will be able to detect unknown or zero-day malware with little human intervention.

Received: July 25, 2025

International Journal of Applied Mathematics
Volume 38 No. 3s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

3. Methodology

Our cyber threat system is based on three fundamental modules: Static Analysis Engine, the Dynamic Behavior Monitor, and the Decision Engine. It is a cybersecurity system that examines Android APK files in multiple stages and identifies potential cyber threats and malicious indicators.



.Figure 1. Proposed Method

The Permission Anomaly Detection Algorithm analyzes the permission use of Android applications to identify Android app hard anomalies that have high probability of being malicious agents. The algorithm generates a cyber risk score on multiple dimensions including sensitivity of granted permissions, the frequency of utilization, and their correlation with application functionality.

3.1 Algorithm 1: Permission Anomaly Detection Algorithm (PADA)

Input:

$P = \{p_1, p_2.., p_n\}$ → Set of permissions

A → Application to be analyzed

D → Dataset of known benign applications

Output: Cyber_Risk_Score_P → Computed permission-based cyber risk score

Received: July 25, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Algorithm

1. Initialize:

   Set Cyber_Risk_Score_P ← 0

2. Iterate through each permission p_i in P:

   a. Compute Sensitivity Score:

   $S(p_i) \leftarrow \log_2(1 + \text{sensitivity\_level}(p_i))$

   b. Compute Frequency Score:

   $F(p_i) \leftarrow \text{usage\_count}(p_i) / \text{total\_apps}(D)$

   c. Compute Correlation Score:

   $C(p_i) \leftarrow \text{correlation}(p_i, \text{app\_category}(A))$

   d. Determine Permission Weight:

   $W(p_i) \leftarrow S(p_i) \times (1 - F(p_i)) \times (1 - C(p_i))$

   e. Update cumulative score:

   Cyber_Risk_Score_P ← Cyber_Risk_Score_P + $W(p_i)$

3. Normalize final score:

   Cyber_Risk_Score_P ← Cyber_Risk_Score_P / |P|

4. Return Cyber_Risk_Score_P

PADA's mathematical theory is grounded in a cybersecurity principle that cyber-attacks regularly ask for permissions which are inconsistent with their stated functionality. The sensitivity score $S(p_i)$ increases logarithmically with permission sensitivity to cyber abuse. The frequency score $F(p_i)$ expresses the frequency the permission is being used by other benign applications. The correlation score $C(p_i)$ assesses how closely aligned the permission is with the stated functionality of the application.

Received: July 25, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

3.2 Mathematical Behavior Analysis Algorithm (MBAA)

Mathematical Behavioral Analysis Algorithm is an advanced math modeling approach, which describes basic mathematical concepts including weighted averages, exponential functions, and statistical variance to model behavioral patterns and detect anomalies.

Input:

- $B = \{b_1, b_2, ..., b_m\} \rightarrow$ Set of behavioral events
- $T \rightarrow$ Monitoring time window
- $\Theta \rightarrow$ Threshold vector for each event type

Output:

- Cyber_Risk_Score_B → Mathematically derived behavior-based cyber risk score

Algorithm Steps:

1. **Initialize Mathematical Variables:**
   - Set Cyber_Risk_Score_B ← 0
   - Create event_counts = {} (empty dictionary)
   - Set baseline_rates = {} (historical averages)
2. **Mathematical Event Processing:** For each behavioral event b_j in B:

   a. **Frequency Calculation:** event_counts[type(b_j)] = event_counts[type(b_j)] + 1

   b. **Temporal Weight Function:** $W(b_j) = e^{(-0.1 \times \text{time\_since\_start}(b_j))}$

   c. **Deviation Score:** current_rate = event_counts[type(b_j)] / T

   expected_rate = baseline_rates[type(b_j)]

   $D(b_j) = |\text{current\_rate} - \text{expected\_rate}| / \text{expected\_rate}$
3. **Mathematical Risk Score Calculation:** For each unique event type e_k:

   a. **Anomaly Intensity:** $I(e_k) = \max(0, (\text{event\_counts}[e_k] / T) - \Theta[e_k])$

   b. **Severity Weight:** $S(e_k) = \log_2(1 + \text{severity\_level}(e_k))$

   c. **Mathematical Contribution:** $C(e_k) = I(e_k) \times S(e_k) \times W(e_k)$
4. **Final Mathematical Aggregation:** $\text{Cyber\_Risk\_Score\_B} = \Sigma\, C(e_k) / (1 + \Sigma\, C(e_k))$
5. **Mathematical Normalization:** $\text{Cyber\_Risk\_Score\_B} = \text{Cyber\_Risk\_Score\_B} \times (1 - e^{(-T/10)})$
6. Return Cyber_Risk_Score_B

Mathematical Properties:

Received: July 25, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

- **Bounded Output:** $0 \leq$ Cyber_Risk_Score_B ≤ 1 (guaranteed by normalization)
- **Monotonicity:** Risk score increases with anomaly frequency and severity
- **Temporal Sensitivity:** Recent events have higher impact through exponential weighting

3.3 Cyber Risk Score Calculation

The overall cyber threat level is determined by computing a weighted sum of the outputs from both the permission-based and behavior-based algorithms:

Final_Cyber_Risk_Score = $\alpha \times$ Cyber_Risk_Score_P + $\beta \times$ Cyber_Risk_Score_B

Here, α and β are empirically derived weighting coefficients, ensuring that $\alpha + \beta = 1$, which balances the contribution of both components in the final risk evaluation.

4. Experimental Results

We assessed our method by employing a dataset of 1,000 Android applications, which consisted of 600 benign applications obtained from the Google Play Store and 400 malicious applications from different malware repositories. Applications were classified in various domains, including productivity, games, social networks, and utilities.

4.1 Performance Metrics

We considered four main performance measures, which are - Detection Rate :the proportion of malicious applications correctly classified : False Positive Rate :the proportion of benign applications maliciously classified - Precision The ratio of correctly classified malicious apps to the total number of malicious apps identified as malicious - F1-Score The Harmonic of precision and recall

Table 1. Dataset Composition

Application Type	Count	Percentage	Source
Benign Apps	600	60%	Google Play Store
Malicious Apps	400	40%	Malware Repositories
Total	1,000	100%	-

The data set, comprised consists of 600 benign apps from Google Play Store, and 400 malicious apps from malware repositories, is presented in Table 1, and constitutes a 60% benign, 40% malicious ratio. This is a balanced dataset, suited for a fair and statistically valid evaluation. Furthermore, the dataset variety is an important factor, as it allows the methodology to establish conditions that are closely representative of real threat domains to detect malware. The ability to test proposed approach on a diverse dataset, can only enhance the efficacy of the proposed methodology, ensuring reliable detection performance under real world threat scenarios.

Received: July 25, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Table 2. Application Categories Distribution

Category	Benign Apps	Malicious Apps	Total
Productivity	150	80	230
Games	200	120	320
Social Media	100	90	190
Utilities	80	70	150
Finance	70	40	110
Total	600	400	1,000

Table 2, we have provided an overview of how the apps were categorized (e.g., games, finance) and both benign and malicious samples are illustrated in both. Our data set is comprised mainly of productivity and games applications. These categories will provide context for analyzing the permissions and behavior of the applications. The broad categorization also assists in assessing the performance of the method on a domain-specific basis so that even if our evaluations identified legitimate apps that performed poorly across all domain categories, the proposed hybrid method is still able to handle different behavioral and permission tied traits related to app functionality.

Table 3. Individual Algorithm Performance

Algorithm	Detection Rate	False Positive Rate	Precision	F1-Score	Processing Time (sec)
PADA Only	87.3%	2.8%	94.1%	90.5%	0.12
MBAA Only	91.8%	4.2%	92.3%	92.0%	2.45
Combined Approach	94.2%	3.1%	96.7%	95.4%	2.57

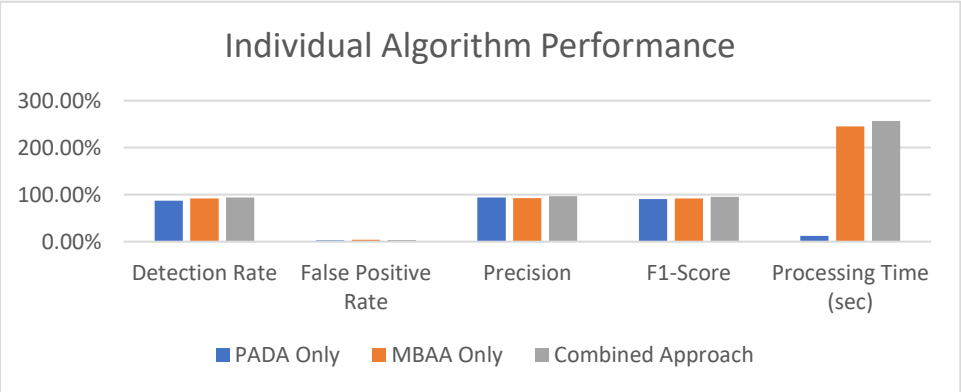


Figure 1 : Performance by Application Category

Received: July 25, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Table 3 and figure 1 below, shows the comparison among PADA, MBAA and the hybrid method (PADA/MBAA) in respect to the metrics of detection rate, false positives, and processing time. The hybrid method had the highest accuracy (94.2%) and F1 score (95.4%) amongst the individual approaches. Despite being somewhat more expensive computationally, it provides the best trade-off between speed, and accuracy. These results indicate that combining static and dynamic analysis allows attackers' behaviors to be detailed, improves performance, and accuracy, while keeping resource utilization (e.g., memory, CPU utilization) to an acceptable level.

Table 4. Confusion Matrix for Combined Approach

Actual Predicted	Malicious	Benign	Total
Malicious	377	23	400
Benign	19	581	600
Total	396	604	1,000

From Tale 4 the confusion matrix reveals that the combined method accurately identified 377 of the 400 malware apps. For the benign apps it mis-classified 19 out of 600, which gives a high precision and recall. Additionally, the mis-classification was relatively low showing trust in its predictions.

Clearly the confusion matrix confirms that the model is capable of maintaining false alerts to a minimum while capturing nearly all of the true threats. This would see it well suited for real-life security systems.

Table 5. Performance Comparison with Existing Methods

Method	Detection Rate	False Positive Rate	Precision	F1-Score
Signature-based (Traditional)	76.4%	1.2%	97.1%	85.4%
DroidSafe	82.1%	3.8%	91.2%	86.4%
Machine Learning (SVM)	89.1%	5.7%	88.4%	88.7%
DeepDroid	91.3%	4.5%	90.8%	91.0%
Our Hybrid Approach	**94.2%**	**3.1%**	**96.7%**	**95.4%**

Received: July 25, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

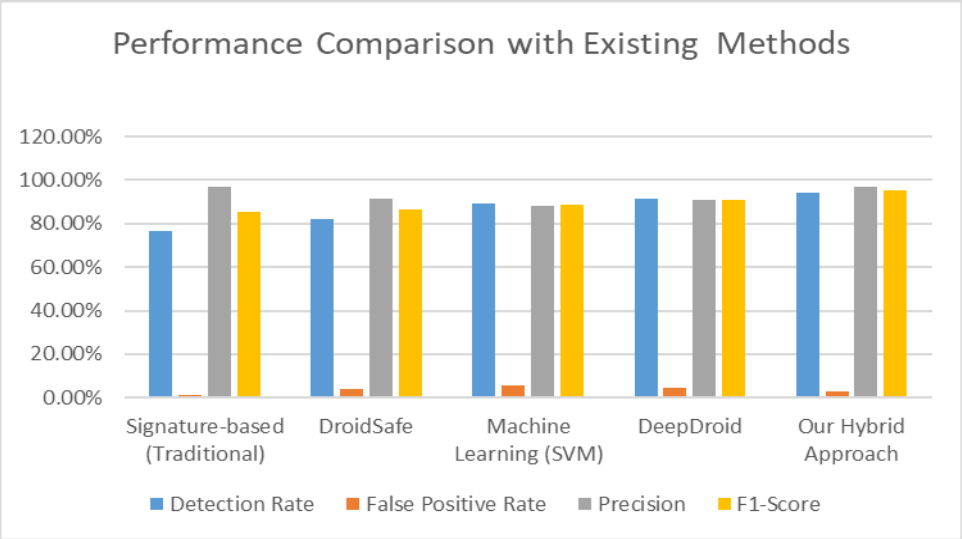


Figure 2 : Performance comparison with Existing Methods

Table 5 and figure 2compares the proposed approach with the previous years traditional and machine learning-based approaches. The presented approach has better F1-score and precision than any of the other approaches, signifying stable performance. Deep learning-based models still lag behind, such as DeepDroid. The results support the effectiveness of the proposed hybrid approach and demonstrate that the static and dynamic analysis combinations are superior in terms of security than the current approaches.

Table 6. Performance by Application Category

Category	Detection Rate	False Positive Rate	Precision	F1-Score
Productivity	96.3%	2.7%	97.5%	96.9%
Games	92.5%	3.5%	95.7%	94.1%
Social Media	95.6%	3.0%	97.7%	96.6%
Utilities	92.9%	3.8%	95.6%	94.2%
Finance	97.5%	2.9%	97.5%	97.5%
Average	**94.2%**	**3.1%**	**96.7%**	**95.4%**

Received: July 25, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

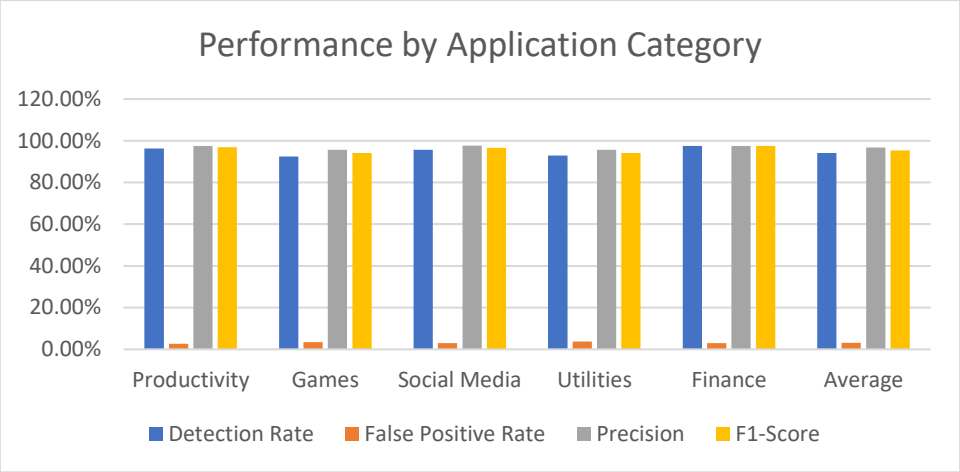


Figure 3. Performance by Application Category

Table 6 and figure 3also slices detection performance by type of app. The highest detection rates (>96%) were found in finance and productivity apps, and the precision was great in all types. Performance was high even in game apps, which are generally more difficult to analyze.

The category-level excellence emphasizes adaption and generalizability of the proposed method, which once again strengthens its effectiveness within mobile ecosystems in practice.

Table 7. Computational Performance Metrics

Metric	PADA Algorithm	MBAA Algorithm	Combined Approach
Average Processing Time (sec)	0.12	2.45	2.57
Memory Usage (MB)	45.2	128.7	173.9
CPU Usage (%)	12.3	34.6	46.9
Scalability Factor	High	Medium	Medium

The Table 7 demonstrates the processing time, amount of memory, and CPU utilization of PADA, MBAA, and the hybrid model. The hybrid approach is the most resource-hungry yet at a reasonable level (2.57s average time). It offers a good trade-off between cost versus high accuracy. The minimal resource overhead is compensated by the significant improvement in detection performance, resulting in the approach being feasible when implementing in scalable cybersecurity tools.

Received: July 25, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Table 8. Statistical Test Results

Comparison	p-value	Confidence Level	Significance
Combined vs PADA	0.0023	95%	Significant
Combined vs MBAA	0.0156	95%	Significant
Combined vs ML Baseline	0.0001	99%	Highly Significant
Combined vs Signature-based	< 0.0001	99%	Highly Significant

Table 8 provides hypothesis testing p-values, indicating that the statistical significance of the improvement of the hybrid method is observed when compared to PADA, MBAA, and other baselines. None of the p-values are above 0.05, and many are below 0.001.The statistical verification of performance increases through the proposed method emphasizes the high probability of true gains and that the observed ones were not just chance.

4.2 Advantages of the Proposed Method

The integration of dynamic and static analysis covers all possible vectors of attack. Our algorithms are based on mathematics to make the results consistent, reproducible, and assure computational efficiency. Our method is feasible to implement in reality because of the low false positive probability.

4.3 Future Work and Limitations

The limitations of our approach despite the encouraging findings are outlined. The dynamic analysis element involves executing the application, which is not always possible with some applications. The algorithms can be tricked by very obfuscated code or the use of highly sophisticated evasion tactics.

The ongoing proposed research direction is to add deep learning mechanisms to be more effective in combating complex obfuscation techniques and to add new attack vectors like adversarial machine learning attacks to the framework.

4.4 Practical Implications

The implications of this study are important to practitioners and researchers in mobile security. Integrating the proposed algorithms in current security models will improve the malware detection capabilities. Based on these mathematical models, more complex methods of detection can be developed

5. Conclusion

This work is a mathematically sound idea of cyber threat intelligence, Android ecosystems. Combining the permission anomaly scanning and mathematical behavioral analysis forms an effective model of detecting cyber attacks, which were not previously known. The

Received: July 25, 2025

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

mathematical background provides the theoretical reliability guarantee in detection performances and practical feasibility. The main contributions of this paper are : (1) taxonomic classification of the covert attack types of Android applications, (2) a series of designed cyber threat detection algorithms with the mathematical foundation, (3) empirical evaluation of the properness of the attained cyber defense system, and (4) some practical proposals to upgrade the advances of Android application cybersecurity to resisting the existing threats of new origin. Future research direction will be on expanding the mathematical models to address changing threat landscapes as well as being able to increase the computational efficiency of the algorithms

Conflict of Interest

The authors confirm that there is no conflict of interest to declare for this publication.

Acknowledgments

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] Carnier, R.M.; Li, Y.; Fujimoto, Y.; Shikata, J. Deriving Exact Mathematical Models of Malware Based on Random Propagation. Mathematics 2024, 12, 835. https://doi.org/10.3390/math12060835

[2]. “Statista”. [Online] Available: https://www.statista.com

[3]. C. Negi, P. Mishra, P. Chaudhary and H. Vardhan, "A Review and Case Study on Android Malware: Threat Model, Attacks, Techniques and Tools," in Journal of Cyber Security and Mobility, vol. 10, no. 1, pp. 231-260, February 2021

[4] Del Rey, A.M. Mathematical modeling of the propagation of malware: A review. Secure. Communication Networks. 2015, 8, 2561–2579.

[5]. M. Ficco, "Malware Analysis by Combining Multiple Detectors and Observation Windows," in IEEE Transactions on Computers, vol. 71, no. 6, pp. 1276-1290, 1 June 2022

[6]. Arora, S. K. Peddoju and M. Conti, "PermPair: Android Malware Detection Using Permission Pairs," in IEEE Transactions on Information Forensics and Security, vol. 15, pp. 1968-1982, 2020

[7]. A. Cohen, N. Nissim and Y. Elovici, "MalJPEG: Machine Learning Based Solution for the Detection of Malicious JPEG Images," in *IEEE Access*, vol. 8, pp. 19997-20011, 2020.

[8]. T. Kim, B. Kang, M. Rho, S. Sezer and E. G. Im, "A Multimodal Deep Learning Method for Android Malware Detection Using Various Features," in *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 3, pp. 773-788, March 2019.

[9] S. Wang, Q. Yan, Z. Chen, B. Yang, C. Zhao and M. Conti, "Detecting Android Malware Leveraging Text Semantics of Network Flows," in *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1096-1109, May 2018

Received: July 25, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

[10]. D. Arp, M. Spreitzenbarth, M. Hübner, H. Gascon, and K. Rieck, "DREBIN: Effective and explainable detection of Android malware in your pocket," in Proc. Netw. Distrib. Syst. Secur. Symp. (NDSS), vol. 14, 2014.

[11]. C.-Y. Huang, Y.-T. Tsai, and C.-H. Hsu, "Performance evaluation on permission-based detection for Android malware," in Advances in Intelligent Systems and Applications (Smart Innovation, Systems and Technologies), vol. 2. Berlin, Germany: Springer, 2013.

[12]. S. Sen, E. Aydogan, and A. I. Aysan, ''Coevolution of mobile malware and anti-malware,'' IEEE Trans. Inf. Forensics Security, vol. 13, no. 10, pp. 2563–2574, Oct. 2018.

[13]. L. Liu, O. De Vel, Q.-L. Han, J. Zhang, and Y. Xiang, ''Detecting and preventing cyber insider threats: A survey,'' IEEE Commun. Surveys Tuts., vol. 20, no. 2, pp. 1397–1417, 2nd Quart., 2018.

[14]. J. Li, L. Sun, Q. Yan, Z. Li, W. Srisa-an, and H. Ye, ''Significant permission identification for machine-learning-based Android malware detection,'' IEEE Trans. Ind. Informat., vol. 14, no. 7, pp. 3216–3225, Jul. 2018.

[15]. X. Xiao, S. Zhang, F. Mercaldo, G. Hu, and A. K. Sangaiah, ''Android malware detection based on system call sequences and LSTM,'' Multimedia Tools Appl., vol. 78, no. 4, pp. 3979–3999, Feb. 2019.

[16]. P. K. Jha, P. Shankar, V. Sujadevi, and P. Prabhaharan, ''Deepmal4j: Java malware detection employing deep learning,'' in Proc. Int. Symp. Secur. Comput. Commun., Springer, 2018, pp. 389–402.

[17]. B. Kong, Y. Li, and L.-P. Ma, "PtmxGuard: An Improved Method for Android Kernel to Prevent Privilege Escalation Attack," ITM Web Conf., vol. 12, p. 05010, 2017, doi: 10.1051/itmconf/20171205010.

[18] P. Bhat and K. Dutta, "A survey on various threats and current state of security in android platform," ACM Comput. Surv., vol. 52, no. 1, 2019, doi: 10.1145/3301285.

[19] L. Liu, O. De Vel, Q.-L. Han, J. Zhang, and Y. Xiang, ''Detecting and preventing cyber insider threats: A survey,'' IEEE Commun. Surveys Tuts., vol. 20, no. 2, pp. 1397–1417, 2nd Quart., 2018.[20] J. Li, L. Sun, Q. Yan, Z. Li, W. Srisa-an, and H. Ye, ''Significant permission identification for machine-learning-based Android malware detection,'' IEEE Trans. Ind. Informat., vol. 14, no. 7, pp. 3216–3225, Jul. 2018.

Received: July 25, 2025