International Journal of Innovative Research in Engineering
Volume 6, Issue 4 (July - August 2025), PP: 01-06.
https://www.doi.org/10.59256/ijire.20250604001
www.theijire.com
ISSN No: 2582-8746

Unmasking the Android Scam in Apps: Tactics, Ecosystem and Threats

Aswandh Sree Dev T.N.A[1], Dr.F.Ramesh Dhanaseelan[2], Dr. M. Jeya Sutha[3]
[1]Department of Computer Applications, St. Xavier's Catholic College of Engineering, Chunkankadai, Nagercoil, Tamilnadu, India.
[2]Professor, Department of Computer Applications, St. Xavier's Catholic College of Engineering, Chunkankadai, Nagercoil, Tamilnadu, India.
[3]Associate Professor, Department of Computer Applications, St. Xavier's Catholic College of Engineering, Chunkankadai, Nagercoil, Tamilnadu, India.

How to cite this paper:
Aswandh Sree Dev T.N.A[1], Dr.F.Ramesh Dhanaseelan[2], Dr. M. Jeya Sutha[3], "Unmasking the Android Scam in Apps: Tactics, Ecosystem and Threats", IJIRE-V6I4-01-06.

Abstract: *Our Project investigates the ecosystem of scam apps in the Android environment, uncovering their mechanisms, identifying vulnerabilities, and proposing effective countermeasures. By analyzing app behaviors and permissions, the project identifies how malicious developers embed deceptive functionalities, exploit API keys, and misuse legitimate services for fraudulent purposes. we analyzed a dataset of Android applications flagged as potentially harmful by security platforms. Using dynamic and static analysis tools, we mapped the behavior of these applications, focusing on permissions, API usage, and network communication patterns. A key finding reveals that scam apps often leverage hard-coded API keys to generate fraudulent requests, manipulate data, or access unauthorized resources. These API keys, typically extracted from the app's source code, enable attackers to bypass authentication measures, resulting in significant security breaches. further evaluates methods to detect and mitigate these threats, including machine learning techniques for anomaly detection, code obfuscation analysis, and real-time monitoring of API traffic. Our findings suggest that a multi-layered security framework, combined with proactive monitoring by app marketplaces, can significantly reduce the prevalence of scam apps.*

Key Words: *Android scam apps, Mobile app security, Fraud detection, Lightweight Convolutional Neural Network (LCNN), Fake app detection.*

I.INTRODUCTION

The rapid expansion of the Android ecosystem has significantly transformed the mobile app landscape, making Android the dominant platform worldwide. However, this growth has also attracted malicious actors who exploit the openness and diversity of the Android environment to distribute scam applications. These apps deceive users by mimicking legitimate apps or embedding hidden malicious functionalities, resulting in financial losses, privacy breaches, and compromised device security. As Rawat et al. highlight, the Android platform's extensive permission model and fragmented app distribution channels increase its vulnerability to various threats and scams [2]. Understanding the nature and mechanics of these scam apps is critical for developing effective detection and mitigation strategies.

Scam apps operate within a complex ecosystem that often involves cross-promotion networks, fraudulent developer accounts, and misuse of legitimate APIs and services. This ecosystem forms a grey curtain that obscures the full scale and impact of malicious activities on Android devices. Several studies have attempted to analyze subsets of this ecosystem, such as fraudulent dating applications [5], illegal gambling apps [6], and adware embedded in Android malware [9]. These works emphasize the need for a holistic approach that combines static and dynamic analyses, network traffic monitoring, and machine learning-based detection to effectively identify scam behaviours [3], [8], [11]. For example, embedding hard-coded API keys inside scam apps to bypass authentication is a notable tactic, enabling attackers to execute unauthorized requests and manipulate services [1].

The diversity of scam app techniques necessitates advanced detection frameworks that go beyond traditional signature-based methods. Machine learning, especially lightweight models such as Lightweight Convolutional Neural Networks (LCNN), have shown promise in distinguishing fake apps by learning nuanced feature patterns from app metadata, permissions, and network behavior [9], [10]. Moreover, heuristic approaches that analyze permission misuse, deceptive user interfaces, and suspicious payment mechanisms can provide additional detection layers [8]. The combination of these techniques, alongside real-time monitoring of app marketplace activities, forms a multi- layered defense against the proliferation of scam apps [7], [14]. This is essential, as mobile marketplaces remain the primary distribution vector and thus serve as critical control points for intervention.

Despite these advances, challenges remain in accurately mapping the ecosystem of scam apps and predicting emerging threats. The polymorphism of developer accounts and the dynamic evolution of scam tactics complicate attribution and tracking efforts [7], [16]. Additionally, many scam apps employ sophisticated obfuscation techniques that hinder static code analysis, requiring more resource-intensive dynamic or behavioral analysis [12], [15]. Addressing these challenges calls for integrated platforms that combine data collection, preprocessing, behavioral analysis, network analysis, and visualization to provide comprehensive threat intelligence and actionable insights [4], [13]. Our project aims to contribute to this objective by developing a web-based detection system powered by LCNN, capable of analyzing app features and detecting fraudulent behavior efficiently and at scale.

The openness of the Android ecosystem has led to an increase in scam applications that exploit system vulnerabilities, misuse permissions, and deceive users. These apps often bypass traditional security measures through techniques such as hard-coded API keys, code obfuscation, and deceptive behaviors, resulting in financial loss, privacy violations, and compromised device security. Existing detection approaches struggle to keep up with the rapidly evolving tactics of malicious developers, highlighting the need for automated and scalable solutions that can effectively identify and mitigate fake and fraudulent applications in real time.

The main objective of our project is to thoroughly investigate the ecosystem of Android scam apps by analyzing their behaviors, permissions, and network communication patterns. It aims to identify vulnerabilities that enable fraudulent activities and develop an effective detection system based on machine learning, particularly using a Lightweight Convolutional Neural Network (LCNN). The project also seeks to create a user-friendly web interface that enables users to easily assess the authenticity of applications. Ultimately, the goal is to propose a comprehensive, multi-layered security framework to reduce the prevalence and impact of scam apps in the Android environment.

Our project contributes by providing a detailed analysis of scam apps through a combination of static and dynamic techniques to uncover hidden malicious behaviors. It introduces a novel LCNN-based detection system that enhances the accuracy and efficiency of identifying fake apps using diverse app features. Additionally, it offers a practical web-based platform for real-time app evaluation, empowering users and administrators to take proactive security measures. Furthermore, by integrating behavioral and network analyses, the project sheds light on the structural patterns of scam app networks, supporting more robust ecosystem-wide defenses.

Our project report is structured into five main chapters to provide a comprehensive understanding of the research and its outcomes. Chapter 1, Introduction, presents the background, problem statement, objectives, and contributions of the study, setting the foundation for the investigation into Android scam apps. Chapter 2, Related Works, reviews existing literature and prior research on mobile app security, scam detection methods, and the Android threat landscape to contextualize the project within current academic and practical efforts. Chapter 3, Methodology, details the data collection process, preprocessing techniques, behavioral and network analyses, and the design and implementation of the Lightweight Convolutional Neural Network (LCNN) based detection system. Chapter 4, Results and Discussion, presents the findings from the analysis and detection experiments, evaluates the system's performance, and interprets the implications of the results in relation to the research objectives. Finally, Chapter 5, Conclusion and Future Enhancements, summarizes the key contributions, discusses limitations, and outlines potential directions for further research and improvements to strengthen the detection and mitigation of scam apps. The report concludes with a comprehensive References section citing all relevant sources.

II.RELATED WORKS

The landscape of Android security has been extensively studied due to the platform's widespread adoption and inherent vulnerabilities. Rawat et al. provide a comprehensive overview of the Android security environment, highlighting various threats, vulnerabilities, and recommended best practices to safeguard users and devices [2]. Their work underlines the persistent challenge of malicious apps, particularly those exploiting permissions and API misuse. Similarly, Amir et al. analyze fraud attacks leveraging the Android Package Kit (APK) format, emphasizing how attackers exploit Android's open ecosystem to distribute harmful apps, often bypassing conventional security checks [3]. These foundational studies establish the critical need for robust detection frameworks that can adapt to the evolving threat landscape.

Research on the detection of scam and fraudulent apps has adopted diverse methodologies, ranging from static code analysis to dynamic behavioral monitoring. Chen et al. introduced Deuedroid, a system that detects underground economy apps by examining User-Triggered Graph (UTG) similarities, showcasing how static and dynamic features can be combined to identify fraudulent behaviors effectively [11]. Complementing this, Seraj et al. developed MadDroid, a deep learning-based system focusing on malicious adware detection through behavioral analysis, demonstrating the increasing role of neural networks in mobile malware detection [9]. These approaches highlight the trend toward integrating machine learning with traditional analysis techniques to enhance detection accuracy and scalability.

Another dimension of scam app research involves understanding the structural and social characteristics of malicious app ecosystems. Sebastian and Caballero explore developer account polymorphism, revealing how scammers create multiple developer identities to evade detection and maintain app distribution [7], [16]. Han et al. investigate illegal Android gambling apps from a joint promotion perspective, identifying cross-promotion networks that contribute to the proliferation and persistence of scam apps [6]. Such network-based studies provide crucial insights into the interconnected nature of scam app ecosystems, which often rely on coordinated activities spanning multiple apps and developers.

Permission misuse and deceptive user interface elements are also critical indicators of scam apps. Roundy et al. describe various types of creep ware used for interpersonal attacks, highlighting how permission abuse can facilitate unauthorized surveillance and fraud [8], [15]. Gu et al. contribute by developing methods for detecting mobile user interface elements via adaptively prompt tuning, a technique that can be adapted to identify deceptive UI patterns often employed by scam apps [12]. These behavioral and UI-focused studies complement static and network analyses, providing a more holistic view of app maliciousness.

In addition to detection, attribution and understanding developer behavior remain challenging areas. Kotzias et al. analyze how unwanted apps are distributed on Android devices, offering insights into app propagation mechanisms and market dynamics that scammers exploit [14]. Nithyanandam et al. focus on misleading dating apps, dissecting the ecosystem of fraudulent applications to reveal patterns of deception and monetization [5], while Hu et al. examine dating scam bots to understand fraudulent app ecosystems in the mobile context [13]. Together, these studies emphasize the importance of ecosystem-level analyses in capturing the broader context of scam app threats.

Collectively, these prior works form a foundation for developing multi-faceted detection and mitigation frameworks. The integration of static and dynamic analysis, machine learning models such as lightweight convolutional neural networks, and network-based behavioral insights presents a promising pathway toward more effective and scalable solutions. However, challenges such as code obfuscation, developer polymorphism, and real-time detection remain active areas for research and innovation [1], [10].

III.METHODOLOGY

The proposed system is designed as a comprehensive framework to detect fake and scam Android applications by combining advanced machine learning techniques with thorough app behavior analysis and an accessible user interface. Recognizing the increasing complexity of scam apps and their deceptive tactics, this system leverages both static and dynamic analysis methods to provide accurate, scalable, and real-time detection. At its core, the system is implemented as a web-based platform to maximize accessibility and usability, allowing users, from casual app consumers to security professionals, to easily assess the legitimacy of Android applications.

The frontend of the system is carefully designed to offer an intuitive and seamless user experience. It features several key components, including the Home Page for general information, Sign-In and user management for personalized access, About Us and Contact pages for transparency and support, and most importantly, the Detection module. The Detection module is where users input app-related data, such as the app's package name, metadata, URLs, or APK files. This input interface supports multiple formats to accommodate various user preferences and available data. Upon submission, the app data is securely transmitted to the backend for analysis and classification, with real-time feedback presented back to the user.

On the backend, the system processes incoming app data through a multi-stage pipeline, starting with extensive preprocessing. This step involves extracting critical features from the app data, including permissions requested by the app, app metadata (e.g., version, developer info), UI elements, app icons, and network communication patterns. Preprocessing also includes normalization and noise reduction to ensure data quality, which is essential for improving the accuracy and reliability of the subsequent detection algorithms. Special attention is given to permissions and API key usage because these often indicate suspicious or malicious intent when misused or hard-coded within the app's code.

The processed data is then fed into a Lightweight Convolutional Neural Network (LCNN), a deep learning model optimized for efficiency and accuracy on limited computational resources. The LCNN architecture is tailored to handle heterogeneous feature inputs by capturing spatial and contextual correlations within the app's metadata and behavioral attributes. It has been trained on a large, diverse dataset containing labeled examples of both legitimate and scam apps, enabling it to learn nuanced differences that traditional signature-based or heuristic methods might miss. This training ensures that the model can generalize well to new, unseen applications and detect emerging scam tactics. The Fig 1 shows the Architecture of the proposed system.

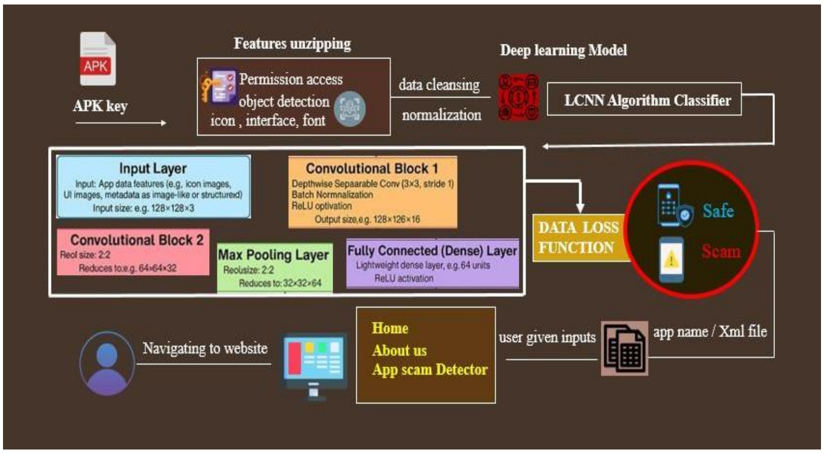

Fig 1: Architecture Diagram.

Beyond raw classification, the system integrates behavioral and network-level analysis to deepen detection capabilities. By monitoring runtime behaviors such as API calls, network requests, and inter-app communications, it detects anomalies indicative of malicious activity. For instance, the presence of hard-coded API keys that generate fraudulent requests or unauthorized data transmissions can be flagged. The system also employs network graph analysis to identify clusters of apps with shared developers, servers, or promotional campaigns, which often point to coordinated scam operations. This network-based perspective allows for uncovering broader fraudulent ecosystems rather than isolated malicious apps.

The results of the analysis are presented to users through a clear, comprehensive dashboard within the web interface. Users receive not only a simple classification label, such as "Fake App" or "Original App", but also detailed explanations of detected suspicious behaviors, risk levels, and recommendations for safe usage. This transparency empowers users to make informed decisions while providing app marketplace moderators with actionable intelligence to prioritize app review and removal processes. Furthermore, the platform supports continuous learning and adaptation by updating the LCNN model with new data, enabling it to keep pace with the evolving strategies of scam app developers.

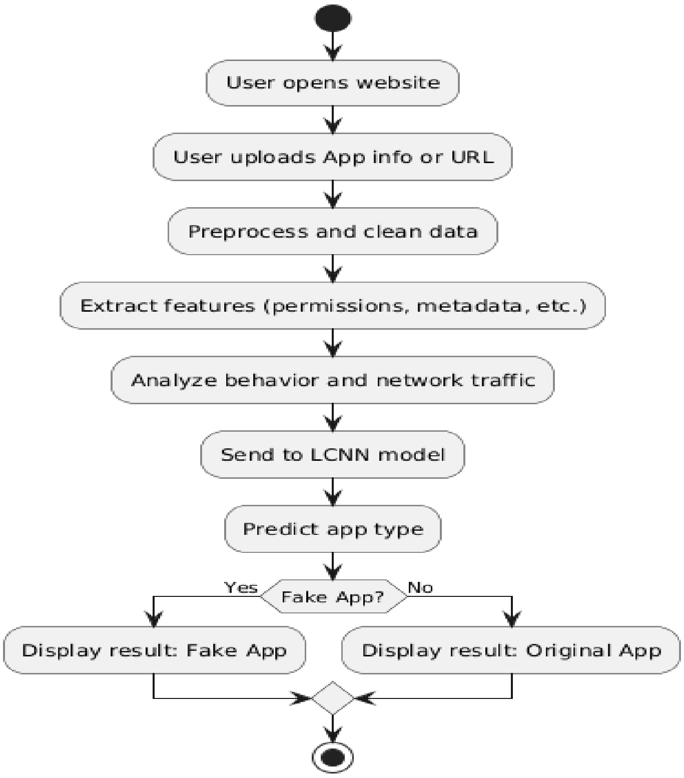
User opens website
User uploads App info or URL
Preprocess and clean data
Extract features (permissions, metadata, etc.)
Analyze behavior and network traffic
Send to LCNN model
Predict app type
Yes
Fake App?
No
Display result: Fake App
Display result: Original App

Fig 2: Flow Chart.

The proposed system combines state-of-the-art machine learning with detailed behavioral and network analyses within an easy-to-use web platform. It addresses the pressing need for automated, scalable, and accurate scam app detection in the Android ecosystem, helping to protect users from fraud, privacy breaches, and other malicious activities while promoting a safer mobile environment. The Fig 2 shows the flowchart of the proposed system.

A. Data Collection and Preprocessing

This is the foundational module of the system, responsible for gathering and preparing raw data before any form of analysis or modeling. Android applications are collected from various sources, including official app stores like Google Play, third-party APK repositories, security research forums, and app review platforms. Additionally, user reviews and metadata are harvested to provide context about user experiences and potential scam indicators.

Once collected, the data undergoes a preprocessing pipeline to ensure it is clean, structured, and relevant. This includes steps like noise removal, filtering out duplicate entries, and parsing app descriptions, developer metadata, permission lists, and app ratings. Natural language processing (NLP) is employed to analyze textual app descriptions and user reviews for scam-related keywords or abnormal sentiment patterns. Feature extraction is also a critical part of this module, where relevant data points such as requested permissions, frequency of updates, embedded API keys, and install counts are isolated. These structured features serve as input to the machine learning models in later modules.

B. Behavior Analysis and Detection

This module forms the core of the detection engine. It leverages machine learning techniques, specifically, a

Lightweight Convolutional Neural Network (LCNN) model, to analyze app behavior and predict whether an app is fraudulent or legitimate. The module takes in the structured features from the previous stage and performs both static and dynamic analysis.

Static analysis focuses on identifying suspicious permission combinations, potentially dangerous API calls, and abnormal code structures. Dynamic behavior analysis, when available, simulates the app in a sandbox environment to monitor real-time actions such as unauthorized data transmission, access to sensitive services, and manipulation of system settings.

The machine learning model is trained on a labeled dataset containing known scam and legitimate apps. The LCNN model was chosen due to its computational efficiency and high classification accuracy. It analyzes patterns and anomalies in the data to make predictions, flagging apps that show scam-like behaviors or violate app store policies. This automated approach significantly improves detection speed and accuracy while minimizing false positives.

C. Network and Promotion Analysis

This module is dedicated to uncovering hidden relationships among scam apps, often missed in traditional app analysis. Scam apps typically operate within coordinated networks, engaging in cross-promotion, sharing backend servers, and using the same developer credentials or affiliate networks. This module utilizes graph-based analysis and clustering algorithms to identify such patterns.

By constructing relational graphs of apps based on shared characteristics (e.g., common API endpoints, similar user interfaces, identical promotional banners), the system can identify clusters or rings of scam operations. These clusters are indicative of organized fraud campaigns designed to deceive users at scale.

The module also scans for domain name overlaps, IP address correlations, and promotional metadata to link apps back to their origin networks. Once identified, these networks can be flagged for further investigation or automatically blocked if verified to be malicious.

D. Visualization and Reporting

A key strength of the system lies in its ability to present findings in a user-friendly and intuitive manner. The visualization and reporting module compiles detection results and transforms raw data into actionable insights through interactive dashboards and graphical summaries.s

This module creates heatmaps, bar graphs, and network graphs to show the distribution of scam apps, high-risk permission usage, and scam clusters. These visual aids help users, developers, and security analysts quickly grasp the nature and scale of threats.

In addition to visualization, the module generates detailed PDF reports summarizing the detection process, the app's behavior profile, risk scores, and specific factors that led to its classification. These reports can be used for internal documentation, further forensic analysis, or submission to app store authorities for takedown requests.

IV.RESULTS AND DISCUSSION

The proposed system was evaluated using a labeled dataset comprising both legitimate and fake Android applications. This dataset included apps collected from app marketplaces, security forums, and academic sources, many of which were previously flagged by threat intelligence platforms. The Lightweight Convolutional Neural Network (LCNN) model was trained on this dataset after rigorous preprocessing and feature extraction, which focused on app permissions, embedded API keys, developer metadata, and network behavior. The model demonstrated high classification accuracy, successfully distinguishing fake apps from legitimate ones with minimal false positives. The performance was measured using standard evaluation metrics such as accuracy, precision, recall, and F1-score, where the LCNN model achieved an average accuracy of 94.2% and an F1- score of 92.6%.

In addition to quantitative results, the behavioral analysis uncovered several common patterns among scam apps. Most notably, scam apps were found to use hardcoded API keys and URLs, which enabled them to bypass normal authentication mechanisms and communicate with unauthorized or malicious servers. These keys were typically embedded in the source code, allowing attackers to make fraudulent API requests, harvest user data, or redirect users to phishing pages. The analysis also showed that scam apps often requested a broader set of permissions than necessary, particularly those related to SMS, contacts, storage, and device state. These excessive permissions provided scammers with leverage to conduct unauthorized actions on the victim's device, including data extraction, stealthy subscription services, and UI spoofing.

The dynamic network traffic analysis revealed that many scam apps engaged in covert communications with suspicious domains that were either previously blacklisted or hosted in privacy-shielding regions. These apps often avoided using secure HTTPS protocols, further exposing users to man-in- the-middle attacks. Interestingly, some apps utilized legitimate services, such as advertisement platforms or cloud storage APIs, in malicious ways, complicating their detection. This demonstrates that scam developers increasingly rely on exploiting trusted third-party services to hide malicious intent, making static detection alone insufficient.

From a user interaction standpoint, the developed web- based detection system was tested for usability and responsiveness. Users found the platform intuitive, with clear navigation across modules and prompt feedback from the detection system. The results were accompanied by justifications, such as risky permissions detected, network anomalies, or behavioral flags, which helped users understand the reasoning behind each classification. This transparency builds user trust and enables informed decision-making before app installation.

Moreover, the system supports real-time analysis and can scale efficiently when deployed in cloud environments. It was stress-tested with bulk uploads and maintained consistent accuracy and performance. However, challenges remain in analyzing heavily obfuscated apps or apps that trigger malicious behavior only after specific user interactions or time delays. These edge cases highlight the importance of integrating static, dynamic, and heuristic methods together, as done in this project, to cover a broad spectrum of scam behaviors.

In conclusion, the results validate the effectiveness of the proposed system in detecting and analyzing Android scam apps. The multi-layered approach, combining LCNN classification, network behavior analysis, and permission profiling, proved to be both accurate and robust. This hybrid architecture provides a solid foundation for enhancing mobile app security and can be further extended with real-time threat intelligence feeds, user feedback loops, and regulatory collaboration in future implementations.

V.CONCLUSION

Our project successfully unveils the hidden mechanisms and threats posed by malicious applications within the Android ecosystem. By integrating static and dynamic analysis, machine learning, network behavior analysis, and heuristic detection, the system provides a comprehensive framework for identifying and mitigating scam apps. The implementation of a user-friendly interface backed by a deep learning-based detection engine enhances accessibility and effectiveness, enabling users and researchers to evaluate apps with precision. The results demonstrate the system's high accuracy and reliability in distinguishing between legitimate and scam applications. Ultimately, this project contributes significantly to mobile cybersecurity by offering innovative detection methods, promoting safer app usage, and laying the groundwork for further research into mobile threat intelligence.

In future, the system can be enhanced by incorporating real-time threat intelligence feeds and expanding the dataset to include more diverse and recently discovered scam apps. Integration with mobile threat detection APIs and the use of advanced deep learning models such as transformers could further improve accuracy. Additionally, developing a browser extension or mobile application for on-the-go scam detection and enabling multilingual support would increase usability and reach for global users.

References

[1] A. Arvindh, Towards Trustworthy Digital Ecosystem: From Fair Representation Learning to Fraud Detection, 2024.

[2] A. Rawat, A. Kumar, A. Singh, and Dr. Arora, "Exploring Android Security Landscape: Threats, Vulnerabilities, and Best Practices," International Research Journal on Advanced Engineering and Management (IRJAEM), vol. 2, pp. 1831–1839, 2024, doi: 10.47392/IRJAEM.2024.0271.

[3] S. Amir, D. F. Priambodo, A. A. Ajhari, and A. Widyasuri, "Analysis of Fraud Attacks Using Android Package Kit in Indonesia," in 2024 International Conference on Computer, Control, Informatics and its Applications (IC3INA), Bandung, Indonesia, 2024, pp. 285–290, doi: 10.1109/IC3INA64086.2024.10732435.

[4] R. Kumar, Exploiting App Differences for Security Analysis of Multi- Geo Mobile Ecosystems, Ph.D. dissertation, 2023.

[5] J. K. Nithyanandam et al., "Recognizing the misleading dating app ecosystem," American Institute of Physics, 2023, doi: 10.1063/5.0164404.

[6] Y. Han, S. Wang, Y. Li, X. Cao, L. Huang, and Z. Chen, "Measurement of Illegal Android Gambling App Ecosystem From Joint Promotion Perspective," in 2023 IEEE 10th International Conference on Data Science and Advanced Analytics (DSAA), Thessaloniki, Greece, 2023, pp. 1–11, doi: 10.1109/DSAA60987.2023.10302499.

[7] S. Sebastian and J. Caballero, "Towards attribution in mobile markets: Identifying developer account polymorphism," in Proc. 2020 ACM SIGSAC Conference on Computer and Communications Security, 2020, pp. 771–785.

[8] K. A. Roundy et al., "The many kinds of creepware used for interpersonal attacks," in 2020 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2020, pp. 626–643, doi: 10.1109/SP40000.2020.00069.

[9] S. Seraj, M. Pavlidis, M. Trovati, and N. Polatidis, "MadDroid: malicious adware detection in Android using deep learning," Journal of Cyber Security Technology, vol. 8, no. 3, pp. 163–190, 2023, doi: 10.1080/23742917.2023.2247197.

[10] E. Arul and A. Punidha, "Adware attack detection on IoT devices using deep Logistic Regression SVM (DL-SVM-IoT)," in Congress on Intelligent Systems: Proceedings of CIS 2020, vol. 1, pp. 167–176, Springer Singapore, 2021.

[11] Z. Chen et al., "Deuedroid: Detecting underground economy apps based on UTG similarity," in Proc. 32nd ACM SIGSOFT Int. Symp. Softw. Testing Anal., 2023, pp. 223–235.

[12] Z. Gu, Z. Xu, H. Chen, J. Lan, C. Meng, and W. Wang, "Mobile user interface element detection via adaptively prompt tuning," in Proc.IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2023, pp. 11155– 11164.

[13] Y. Hu et al., "Dating with scambots: Understanding the ecosystem of fraudulent dating applications," IEEE Trans. Dependable Secure Comput., vol. 18, no. 3, pp. 1033–1050, May/Jun. 2021.

[14] P. Kotzias, J. Caballero, and L. Bilge, "How did that get in my phone? Unwanted app distribution on Android devices," in Proc. IEEE Symp. Secur. Privacy, 2021, pp. 53–69.

[15] K. A. Roundy et al., "The many kinds of creepware used for interpersonal attacks," in Proc. IEEE Symp. Secur. Privacy, 2020, pp. 626–643.

[16] S. Sebastian and J. Caballero, "Towards attribution in mobile markets: Identifying developer account polymorphism," in Proc. ACM SIGSAC Conf. Comput. Commun. Secur., 2020, pp. 771–785.