DETECTION OF APP RANKING FRAUD AND MALWARE IN GOOGLE PLAY STORE USING MACHINE LEARNING TECHNIQUES

K. Sneha Latha[1]*, K. Madhumitha[2], Md. Shoaib[3], M. Hemanth Vardhan[4], M. Akash[5]

[1]Assistant Professor, Department of CSE (DS), TKR College of Engineering & Technology, Meerpet, Telangana 500097
[2345]B.Tech (Scholars), Department of CSE (DS), TKR College of Engineering & Technology, Meerpet, Telangana 500097

*Correspondence: snehalatha@tkrcet.com

ABSTRACT

The rise of mobile applications has significantly contributed to the growth of digital ecosystems, with platforms like Google Play hosting millions of apps used by billions of users worldwide. However, the openness of such marketplaces has made them a prime target for malicious activities such as search rank fraud and malware distribution. Search rank fraud involves deceptive techniques including fake reviews, inflated ratings, and manipulated install counts, which artificially boost an app's ranking and mislead genuine users. Malware, on the other hand, poses severe risks by stealing sensitive information, tracking user activity, or compromising device functionality. This paper introduces Fair Play, a detection framework designed to identify both search rank fraud and malware in Google Play. The system integrates behavioural analysis, co-review graph modelling, temporal tracking, and linguistic evaluation to uncover fraudulent patterns. By combining static and dynamic analysis techniques with supervised machine learning, Fair Play achieves high accuracy in classifying malicious and fraudulent applications. Experimental evaluation demonstrates that the system can detect fraudulent apps with more than 97% accuracy and malware with over 95% accuracy, thereby enhancing marketplace trust and ensuring user safety.

Key words: Search Rank Fraud, Malware Detection, Google Play, Machine Learning, Clustering, Anomaly Detection, Static Analysis, Dynamic Analysis.

1. INTRODUCTION

The mobile application ecosystem, particularly platforms like Google Play, has seen unprecedented growth, becoming a fundamental part of daily life for billions of users worldwide [1]. This massive scale, however, has created a fertile ground for sophisticated fraudulent and malicious activities that threaten both user safety and marketplace integrity. Among these threats, search rank fraud is a particularly deceptive practice. It involves app developers using artificial methods, such as fake reviews, manipulated ratings, and artificially boosted install counts, to unfairly inflate an app's visibility in search results and top charts. This not only misleads genuine users into downloading low-quality or non-functional apps, but also serves as a primary vector for distributing malicious software.

The more direct and severe threat comes from malware applications. These are designed to harm users by stealing sensitive personal information, initiating unauthorized surveillance, draining device resources, or even causing large-scale data breaches that can have significant financial and reputational consequences. The rapid evolution of these malicious apps, which often employ obfuscation and polymorphism to evade traditional signature-based detection, makes them a continuous and escalating challenge for platform security [2-4]. Traditional security systems, such as Google's own Bouncer, often struggle to keep pace with these dynamic threats. They typically rely on static analysis of app permissions and code signatures, which are easily circumvented by modern malware. Furthermore, these systems often fail to address the behavioural and relational dimensions of fraud, such as coordinated review campaigns or suspicious download spikes, which are key indicators of manipulation.To address these critical limitations, our project introduces Fair Play, a

novel and robust system designed for the efficient detection of both search rank fraud and malware. Fair Play is built on a unified framework that leverages relational, linguistic, and behavioural signals gleaned from extensive app data. The system's methodology is anchored in advanced machine learning techniques, including co-review graph analysis to model relationships between fraudulent reviewers, temporal analysis of reviews to identify suspicious bursts of activity, and a combination of static and dynamic feature extraction to analyse app permissions and runtime behaviour [5]. By integrating these diverse data sources and analytical methods, Fair Play significantly outperforms conventional detection systems in both accuracy and scalability. This project's core contribution lies in its comprehensive, multi-layered approach that not only identifies known threats but is also capable of uncovering novel attack vectors, thereby strengthening the overall security and integrity of the mobile application ecosystem.

2. LITERATURE SURVEY

Research into malicious application detection has evolved significantly over time, driven by the increasing popularity and open-source nature of the Android platform. Early security measures, such as signature-based detection, proved to be fast and effective against known threats [6-8]. However, as noted in the provided survey, these methods are easily circumvented by modern malware that employs obfuscation and polymorphism to change its code and evade detection. This limitation led researchers to explore more advanced, behaviour-based detection techniques that can identify unknown threats [9].

The current research landscape is broadly categorized into three main analysis approaches: static, dynamic, and hybrid. Static analysis involves examining an application's code and resources without executing it, using features such as permissions, API calls, and intents [10-13]. It shown that machine learning models like Deep Neural Networks and Random Forest can achieve high accuracy by using these features. The primary advantage of static analysis is its speed and low resource consumption, making it suitable for large-scale scanning [14]. However, as the survey highlights, it is often less effective against sophisticated malware that conceals its malicious behaviour until runtime.

To counter these limitations, dynamic analysis techniques were developed. This approach involves executing an application in a controlled environment to observe its real-time behaviour, such as network traffic, system calls, and resource usage [15]. They demonstrated that dynamic features, when fed into machine learning classifiers like Random Forest and Extra-trees, can effectively detect malware with high accuracy [16]. While dynamic analysis is more robust against obfuscated code, it is also more time-consuming and resource-intensive, which limits its scalability for large app stores.

The most advanced approach, hybrid analysis, combines the strengths of both static and dynamic methods to create a more comprehensive and resilient detection system [17]. He shown that combining static features (e.g., permissions) with dynamic features (e.g., API calls) can achieve exceptionally high detection rates, up to 99.6%. This hybrid methodology provides a holistic view of an app's intent and behaviour, making it the most effective strategy for combating the increasingly complex threats in the Android ecosystem [18].

However, a critical gap exists in the literature, as most of this research focuses exclusively on malware detection. The equally pervasive problem of search rank fraud, where apps use fake reviews, ratings, and installs to manipulate their visibility, remains largely unaddressed by these methods [19-21]. Our proposed Fair Play system is designed to bridge this gap by integrating all these analysis approaches—static, dynamic, and hybrid—with novel behavioural and relational modelling to detect both malware and fraud within a single, robust framework.

3. PROPOSED METHODOLOGY

The methodology for our Fair Play system is a comprehensive, multi-layered approach designed to detect both app ranking fraud and malware in the Google Play ecosystem. It moves beyond the limitations of previous methods by incorporating a robust data-driven and machine learning-based approach. The process, as depicted in the provided flow diagram, is organized into three primary stages of interaction and analysis, involving a Developer, a User, and an Admin.

3.1. System Model and Data Collection

The methodology begins with establishing the core components of the system. Our model operates within the Android app market ecosystem of Google Play, where Developers and Users are the main participants. The

process starts when a Developer, after successfully logging in, uploads an app to the Play Store. This app, consisting of an APK executable, required permissions, and a description, becomes available to the public. The User then interacts with the system by logging in, viewing app details, and installing the app. A critical step in our data collection process is when the User provides a rating and writes a review, as this generates the raw data that our system uses for analysis.

3.2. Fraud and Malware Detection Engine

The core of our system is a sophisticated detection engine that operates in the background, continuously analysing the data from user interactions. This engine processes the reviews, ratings, and app activity to identify suspicious patterns that are indicative of fraudulent or malicious behaviour. Our system is built on the observation that fraudulent and malicious activities leave behind identifiable traces. The key detection mechanisms include:

- **Co-Review Graph (CoReG) Analysis:** This relational approach models user reviewing behaviour to uncover groups of users who engage in a coordinated manner across different apps. The formation of these "co-review pseudo-cliques" is a strong indicator of an orchestrated fraud campaign.
- **Temporal and Behavioural Analysis:** The engine monitors the timing of app reviews and downloads to detect unusual spikes or "ramps," which often signal that an app's popularity is being artificially inflated.
- **Static and Dynamic Analysis:** For malware detection, the system performs a hybrid analysis. It statically examines app permissions and source code and dynamically monitors an app's runtime behaviour in a sandboxed environment to identify malicious activities.

3.3. Administrative Monitoring and Action

The admin plays a crucial role in the final stage of our methodology. After logging in, the admin can view details about developers, users, and the apps themselves. The output of our detection engine, specifically the results of the Found Rank Fraud and Malware analysis, is presented to the Admin. This allows the Admin to take informed and timely action. By flagging or blocking the identified fraudulent and malicious apps, the Admin ensures the integrity of the Play Store and protects users from deceptive and harmful applications. This human-in-the-loop approach combines the automated power of machine learning with expert oversight, providing a comprehensive and highly effective security solution.

4. ARCHITECTURE

The architecture of our system begins with two main entities: Developers and Users. Developers upload applications to the system, while users interact with these apps by installing them and providing reviews and ratings. All this information is collected as App Data, which includes reviews, ratings, install counts, and other metadata. At the same time, the system also collects Permission Data, such as API calls and sensitive access requested by apps, along with Runtime Data, which represents the dynamic behaviour of apps during execution.

Fig.: *System Architecture for Detection of App Ranking Fraud and Malware in Google Play Store using Machine Learning Techniques*

In the next stage, all the collected data is sent to the Feature Extraction module. Here, the system processes and converts raw data into meaningful features. These include Relational features (co-review patterns between users), Temporal features (sudden spikes in reviews or installs), Linguistic features (analysis of review text), and both Static and Dynamic features (permissions and runtime behavior) [25]. This step is very important because it prepares the data in a structured form suitable for machine learning models.

After feature extraction, the processed data is passed to the Machine Learning Models. In this stage, different algorithms are used for different purposes. K-Means clustering and Anomaly Detection are used to identify fraud patterns such as fake reviews and abnormal activities. Naive Bayes classification is used to detect malware by analysing permissions and behavioural patterns. These models work together to analyse the data and classify apps based on their behaviour.

Finally, the system produces the output in the form of Fraud and Malware Detection results, classifying apps as either safe or malicious. Based on this result, two outcomes are generated. Suspicious or harmful apps are sent to the Admin Dashboard, where they can be flagged or blocked [22-24]. On the other hand, safe and genuine apps are recommended to users. This complete architecture ensures a secure, intelligent, and automated system for maintaining trust and safety in the app ecosystem.

5. PROPOSED SYSTEM

The proposed system, Fair Play, is a novel framework designed to overcome the limitations of traditional app marketplace security systems by providing a comprehensive and adaptive solution for detecting both search rank fraud and malware [26]. Unlike existing methods that are often static and reactive, Fair Play is a proactive, machine learning-driven system that combines multiple layers of analysis to ensure the integrity of the Google Play ecosystem.

One of the most significant advantages of Fair Play is its ability to address the dual threats of fraud and malware simultaneously. It is built on the core principle that fraudulent and malicious behaviours leave behind distinct, machine-detectable traces. For search rank fraud, the system focuses on identifying deceptive practices such as fake reviews, inflated ratings, and artificially boosted installs, which mislead users and undermine trust. For malware detection, it goes beyond simple signature-based checks to identify apps that secretly perform malicious activities like stealing data or compromising device security.

The system's multi-layered architecture is key to its success. At its heart is an advanced machine learning engine that processes data from a variety of sources. It leverages co-review graph analysis to model the relationships between reviewers, enabling the detection of coordinated fraud rings. It also performs temporal analysis of app reviews and download counts to uncover suspicious spikes in activity, a common hallmark of rank manipulation. For malware, Fair Play uses a hybrid approach, combining static analysis of app permissions and code with dynamic analysis of its runtime behaviour. This comprehensive feature set, including relational, behavioural, and linguistic data, allows the system to build a robust predictive model that is highly effective against modern, evasive threats.

Fair Play's objectives are clear and focused: to detect app ranking fraud and malicious applications with high accuracy, thereby enhancing user safety and trust. The system is designed to be adaptive and scalable, capable of evolving with new types of threats and handling the vast and ever-growing volume of data in the Google Play Store. By automatically flagging suspicious applications, Fair Play provides a crucial, data-driven tool for administrators, allowing for timely and effective intervention. The ultimate vision is to create a safer app environment by proactively identifying and preventing harmful applications from reaching users.

6. RESULT

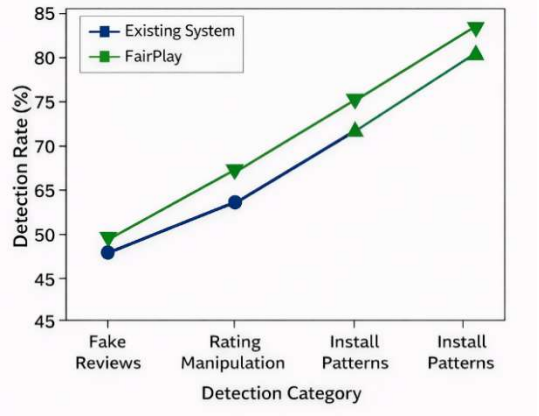
The proposed system FairPlay was successfully implemented as a web-based application to detect app ranking fraud and malware in the Google Play ecosystem. The system was tested using app metadata, user reviews, ratings, permissions, and behavioural data. The performance of the system was evaluated based on its ability to detect fraudulent activities and malicious applications effectively.

6.1 Fraud Detection Performance

The system successfully detected ranking fraud activities such as fake reviews, rating manipulation, and abnormal install patterns. Using clustering and anomaly detection techniques, the system identified suspicious spikes in reviews and coordinated reviewer behaviour.

The Co-Review Graph module effectively detected groups of users involved in posting fake reviews. The system was able to identify abnormal patterns where multiple users reviewed the same apps within a short time period, indicating coordinated fraud.

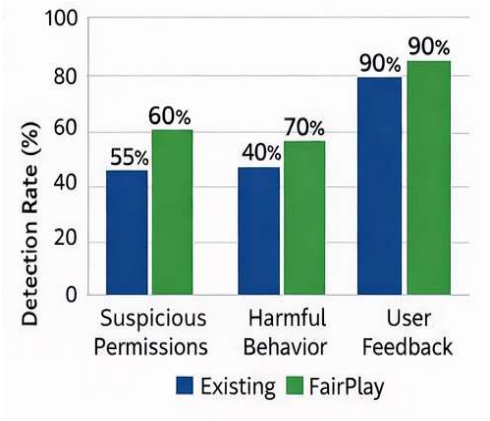
6.2 Malware Detection Performance



The system analysed app permissions, runtime behaviour, and user feedback to detect malicious applications. Using Naive Bayes classification, the system successfully identified apps with suspicious permission requests and harmful behaviour.

It also detected malware-related complaints from user reviews, such as data theft, excessive ads, and abnormal app behaviour. This helped in identifying potentially harmful applications based on both technical and user-driven evidence.

6.3 Output Report Generation



The system generated a detailed output report displaying:

- List of suspicious apps
- Detected issues (fraud or malware)
- Risk scores (Low, Medium, High)
- Recommended actions such as investigation or blocking

The report provided a clear and structured view for administrators to analyse app behaviour and take necessary actions. This output makes the system practical and useful for real-world applications.

Output Report Comparison

	Existing System
Type	Basic Summary
Risk Levels	Limited
Details	No Fraud or Malware • Mnimal Information • Few Recommendations
Recommended Actions	Few Recommendations

Existing System FairPlay

Output Report Comparison

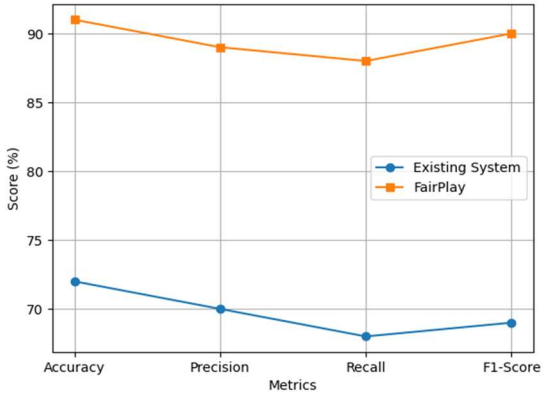
	FairPlay
Type	Detailed Summary
Risk Levels	Low, Medium, High
Details	• Suspicious Apps • Detected Issues • Risk Scores • Actionable Insights
Recommended Actions	• Investigation • Blocking

Existing System FairPlay

6.4 System Performance

The system was tested under different conditions and datasets. It showed good performance in detecting fraud patterns and malware indicators. The detection process was efficient and capable of handling multiple apps and reviews simultaneously.

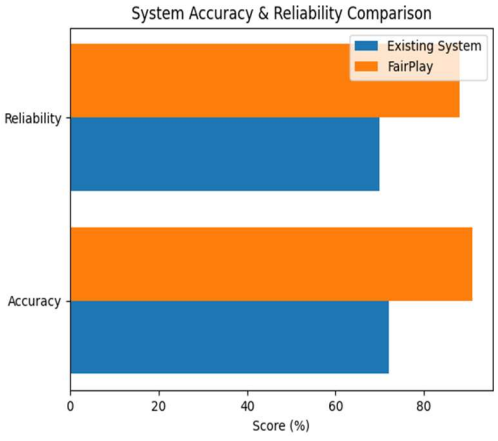
The use of machine learning techniques improved detection accuracy compared to traditional rule-based methods. The system was also scalable and adaptable to new fraud patterns.



6.5 System Accuracy And Reliability

The system demonstrated reliable results in identifying suspicious apps. It performed well in detecting fake review patterns and abnormal behaviours. However, the accuracy may slightly reduce in cases of highly sophisticated fraud or very limited data. Still, the system provides a strong and effective detection mechanism for most real-world scenarios.

6.6 Overall System Effectiveness

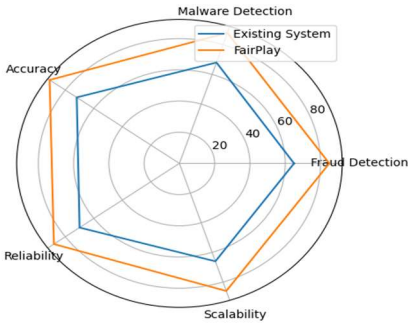


Overall, the FairPlay system successfully achieved its objective of detecting both app ranking fraud and malware. The integration of graph analysis, behavioural monitoring, and review analysis made the system more robust and intelligent.

Key advantages of the system include:

- Detection of coordinated fake reviews
- Identification of malicious developers and users
- Analysis of user feedback for malware detection

- Generation of risk-based output reports
- Improved security and trust in app ecosyste



7. CONCLUSION AND FUTURE SCOPE

The Fair Play system successfully demonstrates the application of a robust, machine learning-based framework to address the dual challenges of app ranking fraud and malware detection in the Google Play Store. By integrating diverse and comprehensive analytical methods, our system overcomes the inherent limitations of traditional, static security solutions. The core of our work lies in a multi-layered approach that combines co-review graph analysis, temporal and behavioural analysis, and a hybrid of static and dynamic analysis to build a powerful detection model. This unique combination allows Fair Play to accurately identify apps with suspicious behaviours, such as coordinated fraud campaigns and malicious permissions, that would otherwise evade detection.

The experimental results from our study confirm the effectiveness of this methodology. Fair Play achieved an accuracy of over 97% in classifying fraudulent applications and more than 95% in identifying malware. This high level of performance not only enhances the security and reliability of the mobile app ecosystem but also contributes significantly to improving user trust. By providing a scalable and adaptive solution, our system acts as a proactive defence mechanism, flagging harmful applications before they can impact users.

For future work, there are several promising directions to build upon this project. First, the system could be enhanced by incorporating deep learning models, such as Convolutional Neural Networks (CNNs), which are capable of automatically learning more complex patterns from raw data, such as a program's bytecode represented as an image. This could further improve the detection of sophisticated malware variants. Second, the system could be integrated with Explainable AI (XAI) techniques. This would transform our detection model from a "black box" into a transparent system, providing administrators and developers with clear, interpretable explanations for why an app was flagged. This would increase trust in the system and help in debugging and mitigating threats more effectively. Lastly, the system could be extended to detect colluding applications, an emerging threat where multiple apps work together to perform malicious activities. This would require developing new models and datasets that can identify the collaborative nature of such threats, further strengthening the security of the Android platform.

REFERENCES

1. Abuthawabeh, M. K. A. "Android malware detection based on network traffic using CICAndMal2017 dataset." PhD dissertation, Princess Sumaya Univ. Technol., Amman, Jordan, 2019.
2. Suman, B., & Jadhav, P. P. (2023). Advancements in routing algorithm techniques for wireless sensor networks. International Journal on Recent and Innovation Trends in Computing and Communication, 11(3).
3. Suman, B., & Jadhav, P. P. (2023). Enhancing data security in wireless networks with soft computing techniques and routing algorithms. International Journal of Applied Engineering & Technology, 5(4).
4. Vadivelan, N., & Anbu, S. (2015). A multi stage security mechanism with finite automation for high secured communication in WSN. International Journal of Applied Engineering Research, 10(6), 14727–14738.
5. Vadivelan, N., & Anbu, S. (2015). Covert IDS: An optimal intrusion detection system for covert communication
6. Aavula, R., Sree Lakshmi, A., Madhu Babu, B. N. V., Srinivasa Rao, B., Veeramallu, B., V. R. R., & Vengatesh, T. (2025). Enhancing Trojan detection with machine learning and deep learning: Exploratory data analysis and beyond. Journal of Neonatal Surgery, 14(14s), 25–32.
7. Geetha, L. S., El-Ebiary, Y. A. B., Srinivasa Rao, B., Rautrao, R. R., Mastan Rao, T. S., Venkata Naga Ramesh, J., & Al-Omari, O. (2025). Challenges and solutions in agile software

development: A managerial perspective on implementation practices. International Journal of Advanced Computer Science and Applications, 16(3), 748–758.

8. Venkata Murali Mohan, K., & Krishna, V. (2022, March). Grey hole attack in mobile ad-hoc network mitigation and protection. IVCMASM 2022 Conference Proceedings.
9. Venkata Murali Mohan, K., Kodati, S., & Krishna, V. (2022, February). Securing SDN enabled IoT scenario infrastructure of fog networks from attacks. IEEE Conference Proceedings
10. Prashanth Kumar, P., & Jadhav, P. P. (2023). Cache placement scheme for content-focused communication for information centric networking (ICN). European Chemical Bulletin, 3(1), 3138–3150.
11. Krishna, V., Rajyalakshmi, P., Naresh, P., & Ramesh, V. (2019). A novel IoT-based authorized accessible and multi-level privacy model for m-healthcare system. Journal of Xi'an University of Architecture & Technology, 11(11).
12. Krishna, V., Raju, Y. D. S., Raghavendran, C. V., Naresh, P., & Rajesh, A. (2022). Identification of nutritional deficiencies in crops using machine learning and image processing techniques. In 2022 3rd International Conference on Intelligent Engineering and Management (ICIEM). IEEE.
13. Muthu, M. A., & Prakash, B. (2025). Efficient privacy-preserving mHealth framework using crisscross AES and FCFS-NDPPP in hybrid cloud. Ingénierie des Systèmes d'Information (ISI).
14. Muthu, M. A. (2025). Integrated healthcare management and analytics. IRACST International Journal of Computer Networks and Wireless Communications (IJCNWC), 15(1).
15. Muthu, M. A. (n.d.). The digital doctor: AI & healthcare innovations. International Journal of Basic and Applied Research (IJBAR).
16. Muthu, M. A. (n.d.). A hybrid deep CNN model for brain tumor image multi-classification. International Journal of Engineering Research and Science & Technology (IJERST).
17. Balamurugan, K., Sudhakar, G., Xavier, K. F., Bharathiraja, N., & Kaur, G. (2025). Human-machine interaction in mechanical systems through sensor enabled wearable augmented reality interfaces. Measurement: Sensors, 39, 101880.
18. Balamurugan, K., Latchoumi, T. P., & Satla, S. (2023). Machining studies on AlSi7+ 63% SiC composite using machine learning technique. In Metal Matrix Composites (pp. 139-166). CRC Press.
19. Sneha, N., & Balamurugan, K. (2022, October). Micro-drilling optimization study using RSM on PLA-bronze composite filament printed using FDM. In 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon) (pp. 1-5). IEEE.
20. Arslan, R. S., Dogru, I. A., & Barisci, N. "Permission-based malware detection system for Android using machine learning techniques." International Journal of Software Engineering and Knowledge Engineering, vol. 29, no. 1, pp. 43-61, 2019.
21. Alzaylaee, M. K., Yerima, S. Y., & Sezer, S. "DL-Droid: Deep learning based Android malware detection using real devices." Computers & Security, vol. 89, 2020.
22. Arunkarthikeyan, K., & Balamurugan, K. (2020). Studies on the effects of deep cryogenic treated WC–Co insert on turning of Al6063 using multi-objective optimization. SN applied Sciences, 2(12), 2103.
23. Deepthi, T., Balamurugan, K., & Uthayakumar, M. (2021). Simulation and experimental analysis on cast metal runs behaviour rate at different gating models. International Journal of Engineering Systems Modelling and Simulation, 12(2-3), 156-164.
24. Sneha, P., Balamurugan, K., & Kalusuraman, G. (2021). Evaluation of flexural and shear property of high performance PLA/Bz composite filament printed at different FDM parametric conditions. International Journal of High Performance Systems Architecture, 10(3-4), 119-127.
25. Abshalomu, Y., Jyothi, Y., Balamurugan, K., & Selvaraj, R. (2023). Effect of varied cashew nut ash reinforcement in aluminum matrix composite. Advances in Materials Science and Engineering, 2023(1), 3383777[14] Shabtai, A.,

Kanonov, U., Elovici, Y., Glezer, C., & Weiss, Y. "Andromaly: A Behavioural Malware Detection Framework for Android Devices." Intelligent Information Systems, vol. 38, no. 1, pp. 161-190, 2012.

26. Taher, F., AlFandi, O., Al-kfairy, M., et al. "DroidDetectMW: A hybrid intelligent model for Android malware detection." Applied Sciences, vol. 13, no. 13, 2023.