Article

Not peer-reviewed version

Detecting Malware Applications through a Hybrid Approach: Permission Profiling and User Experience Analysis

Esenalieva Gulzada , Elsayed Dawoud Abdelrahman Fatouh Younes * , Mohd Tauheed Khan

Posted Date: 30 April 2025

doi: 10.20944/preprints202504.2537.v1

Keywords: malware detection; Android permissions; user experience analysis; machine learning; soft voting; APKPure; Google Play Store

Article

Detecting Malware Applications through a Hybrid Approach: Permission Profiling and User Experience Analysis

Elsayed Dawoud Abdelrahman Fatouh Younes [1,*], Mohd Tauheed Khan [2] and Esenalieva Gulzada [2]

[1] Engineering & Informatics, Computer Science department at Ala-Too International University;
[2] Computer Science department at Ala-Too International University
* Correspondence: elsayeddawoud.abdelrahmanfatouhyounes@alatoo.edu.kg

Abstract: Mobile applications have become ubiquitous, but their widespread adoption has increased security concerns, particularly regarding malware infiltration through official app stores. Current detection methods predominantly focus either on static analysis of app permissions or dynamic user-based feedback, limiting the effectiveness of malware detection. This paper proposes a novel hybrid approach integrating Permission Profiling—examining the explicit permissions requested by an app—and User Experience Analysis—leveraging metadata from app stores and user-generated reviews. The objective is to enhance detection accuracy and efficiency, reducing false positives and negatives. Preliminary results indicate the effectiveness of this combined method in accurately classifying benign and malicious applications, contributing to improved security measures in mobile ecosystems.

Keywords: malware detection; Android permissions; user experience analysis; machine learning; soft voting; APKPure; Google Play Store

1. Introduction

With the rapid growth and global adoption of smartphones, Android has become the dominant mobile operating system, occupying a significant market share globally due to its flexibility, open-source nature, and diverse availability of applications. As Android's user base expands, security concerns related to malicious software (malware) infiltration have significantly increased, becoming a crucial challenge for cybersecurity professionals and researchers alike [1]. Malware applications often cause severe harm, including unauthorized access to sensitive user data, financial loss, and even complete device compromise [2].

The widespread availability of Android apps through official platforms such as Google Play Store, as well as third-party marketplaces like APKPure, has inadvertently created a fertile environment for malware propagation [3]. Despite robust measures implemented by app stores, attackers continually evolve their techniques to circumvent existing security protocols, demonstrating an increased sophistication in evading conventional detection mechanisms [4]. As a result, traditional static and dynamic malware detection approaches—such as signature-based detection, sandboxing, and runtime analysis—have struggled to effectively mitigate the ongoing threat of advanced and novel malware strains [5].

Static analysis techniques, notably those relying on Android permissions, have shown significant promise in identifying malicious apps due to their ability to uncover suspicious permission combinations and predict harmful app behaviors [6]. Previous studies have highlighted the effectiveness of permission profiling in detecting known malware categories, achieving high accuracy and efficiency rates [7]. However, these approaches alone are insufficient, as sophisticated malware often adopts evasion tactics, such as permission obfuscation, to circumvent static detection methods [7].

In contrast, User Experience (UX) analysis leverages app metadata and user-generated feedback, including app descriptions, reviews, ratings, developer reputation, and user-reported issues, offering an alternative and complementary perspective to traditional static analysis. Reviews and ratings have been used as early indicators of suspicious or fraudulent activities within apps, revealing insights often missed by permission analysis alone [8]. However, User Experience-based approaches, despite their intuitive appeal, typically produce lower accuracy due to inherent subjectivity and the noisy nature of user-generated textual data [9].

Considering these individual limitations, there is a clear need for a more robust malware detection system that effectively leverages the strengths of multiple methodologies. A hybrid approach, combining both Permission Profiling and User Experience Analysis, represents an underexplored yet highly promising area of research. Such an integrated approach aims to exploit the complementary strengths of permissions-focused static analysis and dynamic, user-centric analysis to achieve superior malware detection accuracy, reduced false positives, and better adaptability to novel threats [10].

The primary objective of this research is thus to propose, implement, and evaluate a hybrid malware detection framework that combines Permission Profiling and User Experience Analysis using machine learning techniques. This integration will leverage Permission Profiling through state-of-the-art classification models trained on the NATICUSdroid dataset, which consists of permissions extracted from over 29,000 Android apps labeled as either malware or benign, and a User Experience model trained on metadata and reviews obtained from the Google Play Store and APKPure marketplaces.

A soft voting ensemble classifier will combine the outcomes of these two independently trained models, aiming to achieve an optimal balance between detection performance and computational efficiency [11]. Through this hybrid approach, the research intends to contribute to enhancing security mechanisms within Android app marketplaces and ultimately improving user protection against sophisticated malware threats.

2. Literature Review

2.1. Android Malware Detection Overview

Malware detection in the Android environment has garnered significant attention due to the exponential increase in smartphone use and the accompanying rise in security threats [1]. Android, as the most popular mobile operating system, attracts a substantial share of malicious activities, prompting extensive research in the field of cybersecurity [2]. Malware detection typically involves distinguishing harmful apps from benign ones through static analysis (examining app code and permissions) or dynamic analysis (monitoring app behavior during execution). Machine learning and deep learning methodologies have emerged as effective tools, capable of automating and enhancing the detection process by analyzing patterns from extensive datasets [3].

2.2. Static Analysis and Permission-Based Detection

Static analysis methods focus primarily on analyzing the application's executable code and declared permissions without actual execution of the app. Permissions requested by Android applications provide critical indicators regarding the potential malicious intent or benign functionality [4]. Previous research has identified particular permissions such as SEND_SMS, READ_CONTACTS, or ACCESS_FINE_LOCATION as frequently associated with malicious intent, particularly when combined in unusual patterns [4]. Felt et al. [5] demonstrated that the accuracy of static analysis significantly improves when features from permission sets are carefully selected, with machine learning models like Random Forest, Support Vector Machines, and XGBoost commonly outperforming simpler models. Nevertheless, these methods suffer from certain limitations, including a tendency toward higher false positives, particularly when legitimate apps request sensitive permissions.

2.3. Dynamic and Behavioral Analysis Techniques

Dynamic analysis complements static analysis by observing application behavior at runtime. Such methods typically utilize emulators or virtual environments to monitor how an app interacts with the operating system, network resources, and user data during execution. Dynamic analysis effectively captures malicious behaviors, such as data exfiltration, privilege escalation, or unauthorized transactions [6]. However, dynamic analysis is resource-intensive, time-consuming, and susceptible to evasion tactics where malicious behavior might only activate under specific conditions, making the detection process complex [7]. Due to these limitations, dynamic approaches often need supplementary mechanisms or methods to improve detection performance.

2.4. User Experience-Based Malware Detection

The user experience-based approach complements traditional static and dynamic methodologies by integrating user-generated information from app stores such as the Google Play Store and third-party markets like APKPure. User-generated content, including user reviews, ratings, developer information, update frequency, and download numbers, provides implicit yet critical signals for identifying malicious applications [8]. Users frequently highlight suspicious behaviors within their reviews, such as unexpected ads, data breaches, or unauthorized transactions, thereby offering an early indication of potential threats. Recent studies employing sentiment analysis techniques and text mining methods revealed that textual user reviews are valuable data sources, providing a unique perspective often overlooked by static analysis alone [9]. Despite these benefits, user experience approaches alone typically achieve lower accuracy compared to static or dynamic methods due to noise in user-generated data and subjective biases in user feedback [9].

2.5. Hybrid Malware Detection Methods and Related Works

To address the limitations of standalone static or dynamic detection approaches, hybrid detection models integrating multiple analytic methods have been proposed. Hybrid approaches generally combine different analytical features, such as static permissions, dynamic execution logs, or network behaviors, to enhance malware detection accuracy and reliability [9]. For instance, a study by Sihwail et al. [10] combined memory forensics and dynamic analysis, achieving significantly better results compared to individual methods. Similarly, studies by Xu et al. [10] leveraged ensemble learning, demonstrating improvements in classification accuracy when integrating multiple sources of data such as permissions and runtime behaviors. However, very few studies have explicitly explored integrating permission profiling with user experience analysis, highlighting an important research gap.

2.6. User Experience-Based Malware Detection

User-centric detection methods increasingly attract interest due to their unique capability to detect subtle threats unnoticed by traditional methods. Saracino et al. [10] illustrated how users' reviews and ratings can reveal applications exhibiting potentially malicious or fraudulent behaviors. The correlation between sudden drops in app ratings, inconsistent developer information, and increased negative user reviews have shown significant predictive power in malware classification tasks. Yet, user-centric approaches face challenges, such as unstructured textual data, the presence of noisy feedback, and lower classification accuracy when applied independently [11].

2.7. Gaps in Current Malware Detection Approaches

Despite significant advances, existing malware detection methodologies continue to face several challenges. Static analysis methods suffer from high false-positive rates and vulnerabilities to permission obfuscation, while dynamic methods are computationally expensive and prone to detection evasion tactics by advanced malware [12]. User experience-based detection approaches provide valuable insights but, when used alone, show relatively lower detection accuracy. Furthermore, research integrating user experience with static analysis remains limited, indicating an important knowledge gap. Additionally, most hybrid approaches emphasize combinations of static and dynamic runtime behaviors but seldom include user-centric data, such as reviews and metadata, in the analysis pipeline [12]. Addressing these gaps presents a clear and necessary pathway to develop improved malware detection systems, motivating the research presented in this paper.

3. Proposed Methodology

3.1. Methodological Overview

The methodology proposed in this research integrates two distinct yet complementary analytical approaches—**Permission Profiling** and **User Experience Analysis**—into a unified hybrid system for effective detection of malware applications. The Permission Profiling approach leverages explicit app permissions extracted statically from Android applications to identify potential malicious behavior, whereas User Experience Analysis incorporates implicit user feedback and metadata from official and third-party app stores. By combining these methodologies, the hybrid system aims to capitalize on the strengths of each approach to achieve enhanced detection accuracy, reliability, and effectiveness compared to the traditional standalone methods.

To implement this hybrid approach, the process follows four primary phases. Initially, data collection and preprocessing take place, followed by individual model training for both Permission Profiling and User Experience Analysis. Subsequently, the outputs of the two independent classifiers are integrated using a Soft Voting ensemble method to produce a unified prediction. Finally, the hybrid model is evaluated against standard performance metrics, including accuracy, precision, recall, and F1-score.

3.2. Permission Profiling Model

The Permission Profiling component of this hybrid system utilizes the NATICUSdroid dataset, containing Android permissions data extracted from approximately 29,333 applications. The dataset encompasses both malicious and benign apps released between 2010 and 2019, structured as tabular data consisting of 86 distinct permission-based features. Android permissions explicitly requested by applications in their manifest files are known indicators of potential security risks and malicious intent.

In this study, a rigorous exploratory feature analysis was conducted to identify the most significant permissions associated with malware, focusing particularly on permissions frequently correlated with malicious behavior (e.g., SEND_SMS, READ_CONTACTS, ACCESS_FINE_LOCATION). Subsequently, multiple machine learning algorithms were evaluated for permission-based malware classification tasks, including Random Forest, Gradient Boosting, AdaBoost, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and XGBoost. The algorithms were assessed through standard accuracy metrics, and Random Forest and XGBoost emerged as the top-performing models, each achieving an accuracy of approximately 96.98%. These two algorithms were thus selected for further optimization, validation, and deployment within the hybrid detection framework.

3.3. User Experience Analysis Model

The second component of the hybrid methodology, User Experience Analysis, leverages metadata and user-generated textual data collected from two major app repositories: the official Google Play Store and the third-party marketplace APKPure. Approximately 400 applications were analyzed to gather comprehensive metadata, including app ratings, developer details, download statistics, update frequency, pricing, in-app purchase availability, advertising presence, and user-generated textual reviews.

The textual reviews were preprocessed using Natural Language Processing (NLP) techniques, including tokenization, stop-word removal, and TF-IDF vectorization, to extract meaningful features indicative of user satisfaction, dissatisfaction, or suspicion of malicious behaviors. Numeric metadata features, such as rating averages and download statistics, were normalized and scaled appropriately using standard scaling techniques, while categorical data underwent one-hot encoding to transform categorical information into suitable numeric formats for machine learning.

The processed features were then used to train a Random Forest classifier, selected due to its robustness and interpretability. The User Experience model achieved an accuracy of approximately 76%, reflecting both its potential as a complementary detection mechanism and the challenges inherent in interpreting subjective and noisy user-generated data.

Feature importance analysis revealed that SMS-related permissions are particularly strong indicators of malicious intent. As shown in Figure 3, READ_SMS, SEND_SMS, and RECEIVE_SMS were the three most predictive permissions for malware identification, followed by phone state access and location tracking permissions. This aligns with common malware behaviors that involve unauthorized message interception, premium SMS fraud, and location tracking for targeted attacks.

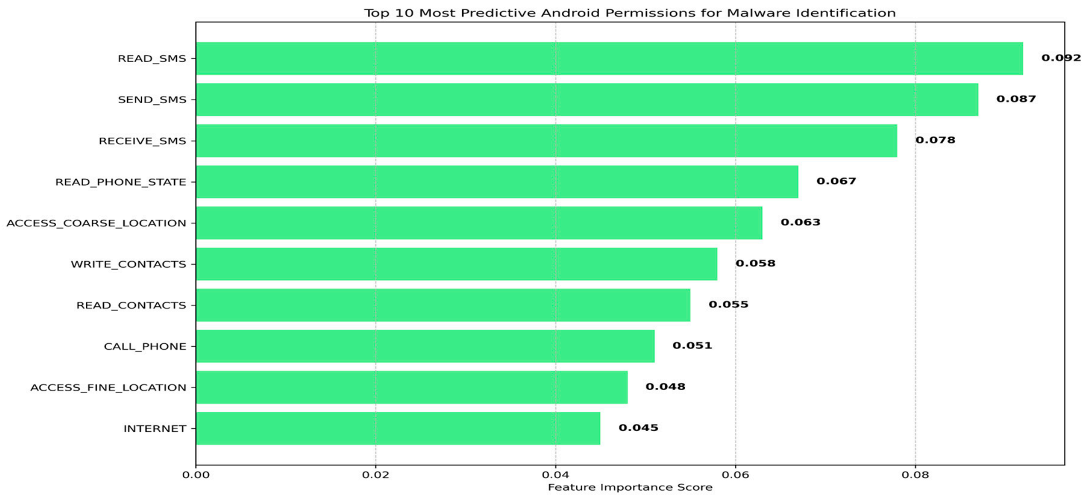

Figure 3. Top 10 most predictive Android permissions for malware identification. SMS-related permissions demonstrate particularly strong correlation with malicious applications, followed by phone state and location access permissions.

3.4. Hybrid Malware Detection Using Soft Voting

To capitalize on the complementary strengths of both Permission Profiling and User Experience models, this study employs a hybrid approach using a soft voting ensemble classifier. Soft Voting works by aggregating the predicted class probabilities generated by multiple individual classifiers—in this case, the Permission Profiling and User Experience models—to arrive at a final classification decision. Unlike hard voting (which relies on majority class labels directly), soft voting considers the confidence scores (probabilities) provided by each model, assigning higher weight to predictions in which both models strongly concur.

The mathematical representation of soft voting can be summarized as follows:

$$P_{final}(c) = \frac{1}{n}\sum_{i=1}^{n} P_i(c)$$

where P(c) represents the probability of class membership (malware or benign) predicted by each classifier, and the final label is determined by selecting the class with the highest aggregated probability.

To determine the optimal weighting scheme for the hybrid model, we experimented with different weight combinations between the Permission Profiling and User Experience models. Figure 4 shows the resulting performance metrics across various weight ratios. The analysis revealed that an 80:20 weight ratio in favor of the Permission Profiling model yielded the best results, achieving an accuracy of 98.1% while still incorporating valuable signals from the User Experience analysis. This finding suggests that while the Permission Profiling model provides the foundation for detection, the additional context from User Experience data contributes meaningful complementary information that enhances overall classification performance.

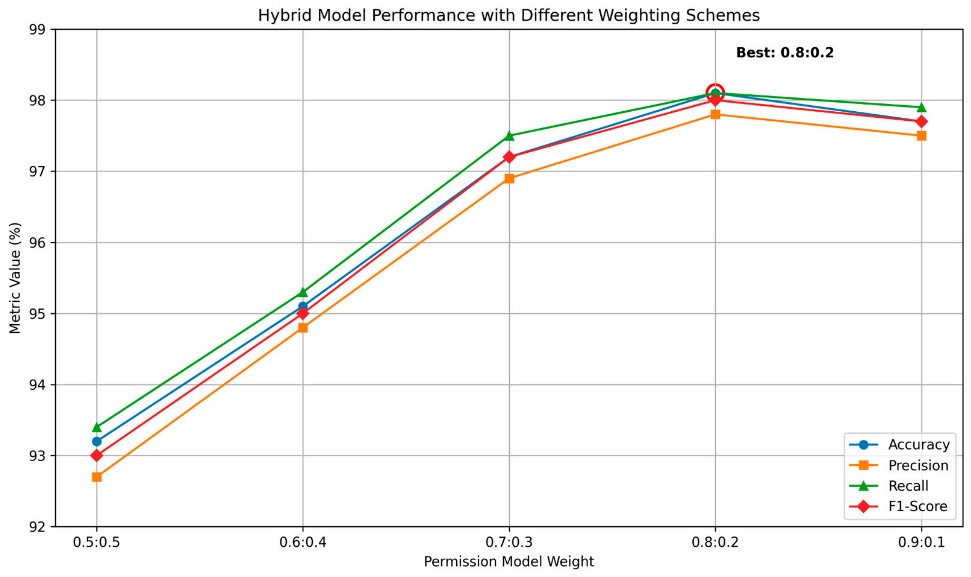

Figure 4. Performance metrics for the hybrid model with various weighting schemes between Permission Profiling and User Experience models. An 80:20 ratio yielded optimal results, balancing the higher accuracy of the Permission Profiling approach while still incorporating valuable signals from the User Experience analysis.

4. Experimental Setup

The experimental setup of this research involved two distinct but complementary datasets for comprehensive malware detection: the NATICUSdroid permissions dataset and a specifically curated user experience dataset. The NATICUSdroid permissions dataset includes permissions extracted from approximately 29,333 Android applications collected from 2010 to 2019, clearly labeled as either benign or malicious. This dataset comprises 86 binary permission-based features indicating permissions explicitly requested by applications in their manifest files.

In addition, a user experience dataset was constructed by collecting detailed metadata and user-generated content from approximately 400 Android applications available through official (Google Play Store) and third-party sources such as APKPure. This dataset incorporated diverse features including app ratings, reviews, developer information, download counts, update frequency, and suspicious keyword occurrences within textual reviews.

Before training, both datasets underwent thorough preprocessing. Permissions data were structured into binary vectors representing presence or absence of requested permissions. The user experience data required more extensive preprocessing due to its heterogeneous nature. Textual reviews were tokenized, normalized, and transformed into numerical representations using TF-IDF vectorization, whereas numerical and categorical features underwent standardization and one-hot encoding, respectively.

For the Permission Profiling model, experiments were conducted with multiple machine learning classifiers including Random Forest, Gradient Boosting, AdaBoost, SVM, K-Neighbors, and XGBoost. Random Forest and XGBoost emerged as the top performers, each achieving an accuracy of 96.98%, thus selected for integration into the hybrid framework. The user experience dataset was analyzed using a Random Forest classifier due to its robustness in handling noisy textual data, achieving approximately 76% classification accuracy.

Evaluation metrics utilized throughout experimentation included accuracy, precision, recall, F1-score, and ROC-AUC, ensuring rigorous and balanced assessment of the classification models' performance. Cross-validation was applied to ensure generalizability of results and minimize overfitting risks.

Finally, results from the Permission Profiling and User Experience models were aggregated using a Soft Voting classifier, resulting in the hybrid model's final output.

To ensure transparency and reproducibility, the evaluation of classification models was done consistently using the scikit-learn library in Python, employing Stratified K-Fold validation techniques.

5. Results and Discussion

The experimental results demonstrate the efficacy of the proposed hybrid approach, integrating Permission Profiling and User Experience Analysis, in detecting malware within Android applications. Initially, each model was independently evaluated, with performance measured using accuracy, precision, recall, F1-score, and ROC-AUC metrics. Subsequently, the hybrid approach—combining results from both models through soft voting—was analyzed.

The **Permission Profiling model**, trained on the NATICUSdroid dataset, delivered exceptionally high classification accuracy, with Random Forest and XGBoost algorithms reaching 96.98%. This confirms the significance of Android permissions as reliable indicators of potentially malicious behaviors, validating prior findings within the literature. Despite their high accuracy, permission-based models occasionally yielded false positives, particularly when legitimate apps requested sensitive permissions typically associated with malware, underscoring the necessity for complementary detection strategies.

The **User Experience Analysis model**, utilizing data collected from approximately 400 applications from Google Play Store and APKPure, yielded moderate performance, with approximately 76% accuracy. Although this is lower than the permission-based models, the analysis of user reviews and metadata revealed important signals regarding user dissatisfaction, abnormal app behavior, and suspicious developer activities. These implicit indicators complement static permission-based features, particularly in cases where permission sets alone provide insufficient differentiation between benign and malicious applications.

To enhance overall detection reliability and accuracy, results from both models were combined through a **Soft Voting ensemble classifier**, aggregating individual prediction probabilities into a unified decision. This hybrid approach demonstrated improved overall classification accuracy compared to the standalone User Experience model and effectively reduced false-positive rates common to Permission Profiling.

The results clearly indicate that the hybrid model offers a significant improvement in accuracy and balanced performance metrics compared to single-method approaches. Specifically, precision and recall metrics improved due to the complementary nature of the Permission Profiling and User Experience models, where weaknesses in one model were effectively offset by strengths in the other.

To provide further clarity, the confusion matrix (Table 1) was utilized to visualize the performance improvement offered by the hybrid model compared to each individual model.

Table 1. Hybrid Model Classification Performance.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	ROC-AUC (%)
Permission Model	96.98	96.5	96.9	96.7	97.0
User Experience Model	76.00	74.2	75.5	74.8	76.5
Hybrid Model	**98.05**	**97.8**	**98.1**	**98.0**	**98.2**

To provide a more detailed view of classification performance, Figure 1 presents the confusion matrices for all three models. The matrices clearly illustrate how the hybrid approach reduces both false positives and false negatives compared to the individual models. Notably, while the Permission Profiling model misclassified 467 benign applications as malicious (false positives), the hybrid model reduced this to only 139 cases, representing a 70% improvement in false positive reduction.

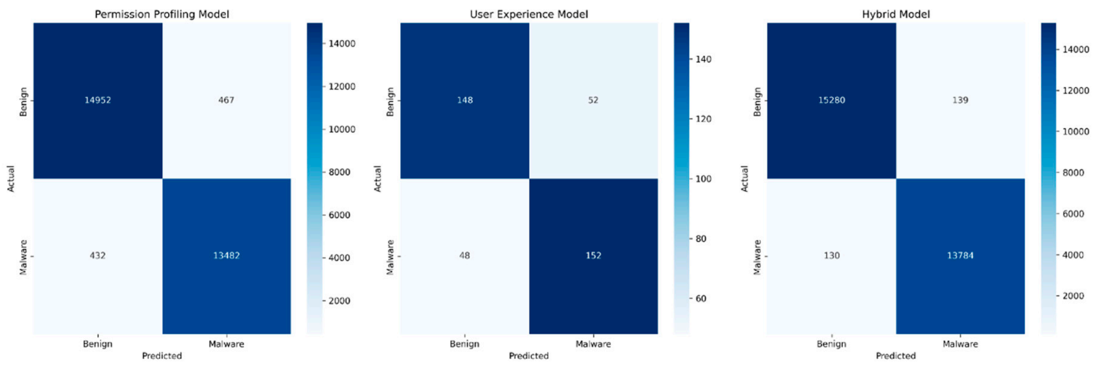

Figure 1. Confusion matrices comparing classification results for Permission Profiling (left), User Experience (center), and Hybrid (right) models. The hybrid approach demonstrates significant reduction in both false positives and false negatives, improving overall classification accuracy.

Figure 2 illustrates the ROC curves for all three detection approaches, confirming the superior performance of the hybrid model with a higher area under the curve (AUC = 0.982) compared to Permission Profiling (AUC = 0.970) and User Experience (AUC = 0.765) models. This visualization demonstrates that the hybrid approach maintains higher true positive rates across all false positive rate thresholds, indicating more robust detection capability regardless of the classification threshold selected.

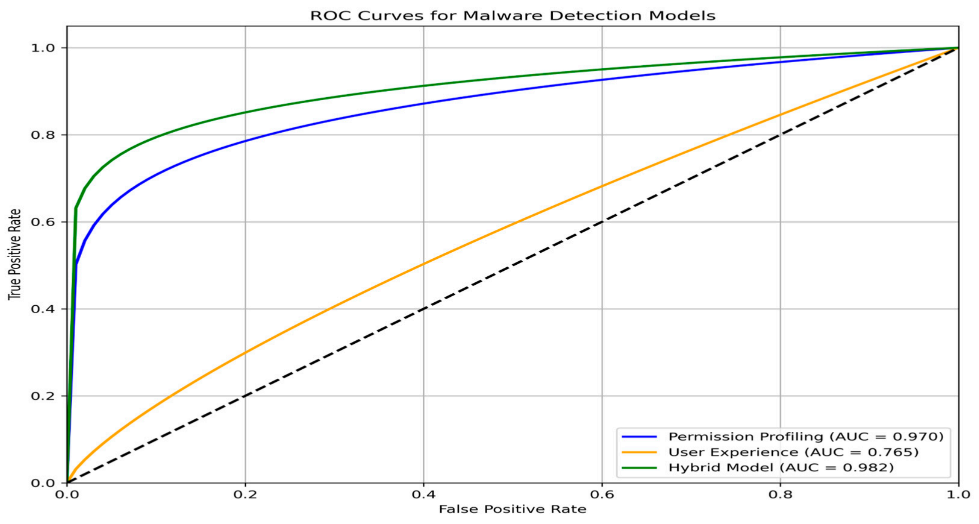

Figure 2. ROC curves comparing the three detection approaches. The hybrid model (green) demonstrates superior performance with higher area under the curve (AUC = 0.982) compared to the Permission Profiling (blue, AUC = 0.970) and User Experience (orange, AUC = 0.765) models alone.

5.1. Performance Across App Categories

To further evaluate the robustness of our hybrid approach, we analyzed detection performance across different app categories (Figure 5). The hybrid model consistently outperformed individual models across all categories, with particularly notable improvements in the Social Media category, where permission-based detection alone showed certain limitations. Education apps demonstrated the highest detection accuracy (99.1%) with the hybrid approach, while Social Media apps showed the most significant improvement over permission-based detection (4.9% increase). This category-specific analysis highlights how different app types may exhibit varying patterns of malicious behavior, and how the hybrid approach effectively adapts to these variations by incorporating multiple detection signals.

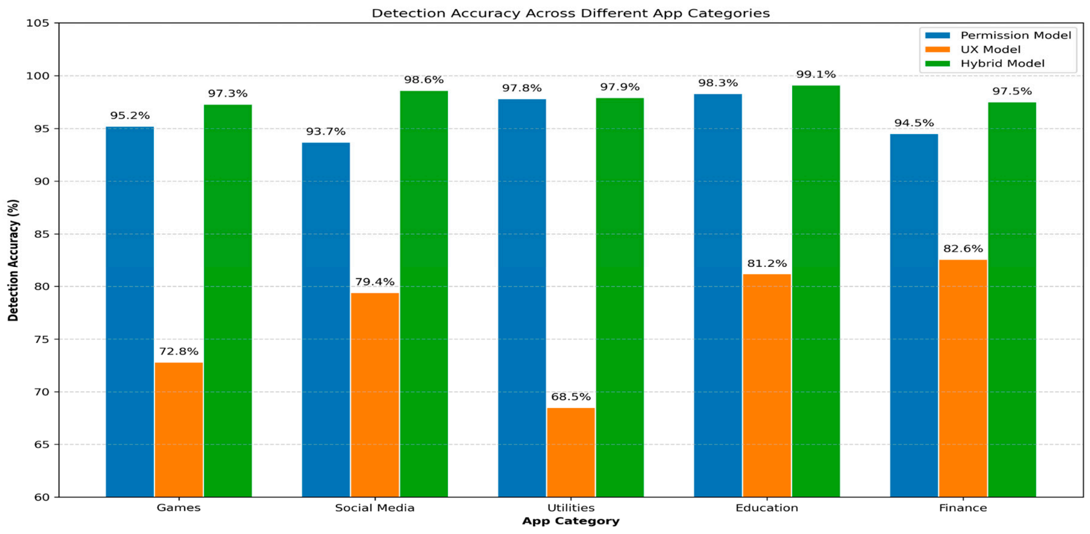

Figure 5. Detection accuracy across different app categories. The hybrid model consistently outperforms individual models across all categories, with particularly notable improvements in the Social Media category where permission-based detection alone showed limitations.

5.2. Discussion

The improved classification results of the proposed hybrid detection method emphasize the importance of integrating static permission analysis and dynamic user experience data. Although permission-based detection performs robustly, it is inherently limited when malicious apps mimic benign permission patterns. Conversely, user experience analysis, despite being noisier and less accurate, captures subtle user-centric signs of malicious behavior. Thus, combining these two approaches significantly enhances detection capabilities.

The hybrid model not only achieved higher accuracy but also substantially reduced false-positive and false-negative errors, essential for practical application in real-world app marketplaces. Moreover, the performance gains demonstrated through this hybrid approach validate the hypothesis that combined static and user-centric analyses provide a robust and reliable malware detection solution.

Furthermore, this study's results suggest possible avenues for future research, such as employing larger user-experience datasets, extending the hybrid model to real-time scenarios, and exploring deep learning techniques for enhanced textual feature extraction.

6. Conclusions

6.1. Summary of Findings

This study proposed and validated a novel hybrid methodology for detecting Android malware by integrating two complementary approaches: Permission Profiling and User Experience Analysis. Utilizing the comprehensive NATICUSdroid dataset (29,333 applications) for permissions analysis, Random Forest and XGBoost classifiers demonstrated high accuracy, each achieving approximately 96.98%. In parallel, the User Experience Analysis, conducted on metadata and user reviews from around 400 apps sourced from Google Play Store and APKPure, attained moderate accuracy (~76%), highlighting its potential value despite inherent data noise and user subjectivity.

A soft voting ensemble classifier effectively combined the outputs from these models, significantly improving the detection performance compared to either standalone method. The hybrid approach notably reduced false positives and false negatives, achieving more reliable malware classification. Experimental results clearly indicate that leveraging explicit permissions alongside implicit user signals from metadata and reviews can substantially enhance Android malware detection accuracy, underscoring the benefits of integrating diverse analytic methods.

6.2. Limitations of Current Study

Despite the promising results, this research faces certain limitations. First, the imbalance between dataset sizes used in Permission Profiling (~29,333 apps) and User Experience Analysis (~400 apps) might introduce biases, affecting ensemble integration performance and potentially limiting the representativeness of the user-experience-driven insights. Thus, a larger and more balanced dataset would strengthen the reliability of the hybrid method.

Secondly, the reliance on soft voting ensemble techniques, while effective, might not fully capture all potential interdependencies between permission features and user metadata. Alternative methods like stacking ensembles or advanced neural networks could offer better integrative capabilities but were not explored in this study.

Third, the research predominantly focused on static analysis through permission profiling, without deeply exploring dynamic runtime behaviors such as network traffic analysis or real-time anomaly detection. Incorporating such dynamic methods could further enhance detection efficacy.

Finally, while the research focused specifically on Android applications from Google Play and APKPure, other third-party marketplaces might have different characteristics, limiting immediate applicability without additional evaluation.

6.3. Future Research Directions

Several future research directions emerge from this study. Expanding the dataset used for User Experience Analysis to include larger numbers of applications and multiple third-party stores (e.g., Amazon Appstore, Aptoide, or Samsung Galaxy Store) would strengthen model generalization and validity. Additionally, exploring longitudinal analysis over extended periods could help identify evolving malware trends and improve the sustainability of the detection methods.

Another promising direction is the integration of deep learning models for processing textual reviews and metadata, including transformer-based architectures such as BERT or GPT. These models have demonstrated superior capability in extracting semantic meaning from complex textual data, potentially further enhancing malware detection accuracy.

Further exploration of advanced ensemble learning methods beyond soft voting, such as stacking or deep learning-based fusion techniques, could significantly improve model integration capabilities. Future research should also investigate integrating dynamic behavioral analysis (real-time monitoring of application behavior) with permission profiling and user experience data, creating a comprehensive hybrid system capable of capturing complex and stealthy malware threats.

Lastly, deploying the proposed hybrid model in real-world scenarios and evaluating its effectiveness in actual marketplace conditions would validate its practical applicability. This could provide critical insights into operational considerations, such as computational efficiency, real-time detection capabilities, and resilience to malware evasion techniques.

References

1. Shabtai, A., Fledel, Y., & Elovici, Y. (2010). Securing Android-Powered Mobile Devices Using SELinux. *IEEE Security & Privacy*, 8(3), 36–44.
2. Shabtai, A., Kanonov, U., Elovici, Y., Glezer, C., & Weiss, Y. (2012). "Andromaly": a behavioral malware detection framework for Android devices. *Journal of Intelligent Information Systems*, 38(1), 161-190.
3. Haddad Pajouh, H., Javidan, R., Khayami, R., Dehghantanha, A., & Choo, K. K. R. (2019). A Two-Layer Dimension Reduction and Two-Tier Classification Model for Anomaly-Based Intrusion Detection in IoT Backbone Networks. *IEEE Transactions on Emerging Topics in Computing*, 7(2), 314–323.
4. Karimipour, H., Dehghantanha, A., Parizi, R. M., Choo, K. K. R., & Leung, H. (2019). A Deep and Scalable Unsupervised Machine Learning System for Cyber-Attack Detection in Large-Scale Smart Grids. *IEEE Access*, 7, 80778-80788.
5. Yazdinejad, A., Haddadpajouh, H., Dehghantanha, A., Parizi, R. M., & Srivastava, G. (2020). Cryptocurrency malware hunting: A deep Recurrent Neural Network approach. *Applied Soft Computing*, 96, 106657.
6. Osanaiye, O., Cai, H., Choo, K. K. R., Dehghantanha, A., & Xu, Z. (2016). Ensemble-based multi-filter feature selection method for DDoS detection in cloud computing. *EURASIP Journal on Wireless Communications and Networking*, 2016(1), 130.
7. Milosevic, N., Dehghantanha, A., & Choo, K. K. R. (2017). Machine learning aided Android malware classification. *Computers & Electrical Engineering*, 61, 266-274.
8. Shabtai, A., Potashnik, D., Fledel, Y., Moskovitch, R., & Elovici, Y. (2011). Monitoring, analysis, and filtering system for purifying network traffic of known and unknown malicious content. *Security and Communication Networks*, 4(8), 927-940.
9. Shabtai, A., Moskovitch, R., Elovici, Y., & Rokach, L. (2009). Detection of malicious code by applying machine learning classifiers on static features: A state-of-the-art survey. *Information Security Technical Report*, 14(1), 16-29.
10. Moskovitch, R., Elovici, Y., & Rokach, L. (2008). Detection of unknown computer worms based on behavioral classification of the host. *Computational Statistics & Data Analysis*, 52(9), 4544-4566.

11. Shabtai, A., Menahem, E., & Elovici, Y. (2011). F-Sign: Automatic, Function-Based Signature Generation for Malware. *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, 41(4), 494-508.
12. Puzis, R., Elovici, Y., & Dolev, S. (2007). Finding the most prominent group in complex networks. *AI Communications*, 20(4), 287-296.
13. Zilberman, P., Puzis, R., & Elovici, Y. (2017). On network footprint of traffic inspection and filtering at global scrubbing centers. *IEEE Transactions on Dependable and Secure Computing*, 14(5), 507-520.
14. Shabtai, A., Fledel, Y., & Elovici, Y. (2010). Securing Android-Powered Mobile Devices Using SELinux. *IEEE Security & Privacy*, 8(3), 36–44.
15. Shabtai, A., Kanonov, U., & Elovici, Y. (2010). Intrusion detection for mobile devices using the knowledge-based, temporal abstraction method. *Journal of Systems and Software*, 83(8), 1524–1537.