International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 02 Issue: 11 November 2024
Page No: 2659- 2663
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2024.0366

Detection of Fraud & Malware Apps in Google Play

Malige Nikhitha[1], Sadiram Reshma[2], Panguluri Ranvanth Kalyan[3], A. Suresh Babu[4]
[1,2,3]UG – Computer Science and Engineering, Institute of Aeronautical Engineering, Hyderabad, Telangana, India.
[4]Associate Professor, Computer Science and Engineering, Institute of Aeronautical Engineering, Hyderabad, Telangana, India.
Emails: *21951a05c4@iare.ac.in[1], 21951a05f0@iare.ac.in[2], 21951a05e8@iare.ac.in[3], a.sureshbabu@iare.ac.in[4]*

Abstract
The proliferation of mobile applications on Google Play has led to an increased risk of fraudulent and malicious apps that can compromise user security and privacy. This study presents a comprehensive approach to detecting fraud and malware in Google Play apps using machine learning (ML) and deep learning (DL) techniques. A Flask-based backend is developed to facilitate the upload, processing, and analysis of datasets containing app features and labels. The system utilizes Support Vector Machine (SVM) and Artificial Neural Network (ANN) models to classify apps as benign or malicious. Key stages include data preprocessing, dataset splitting, model training, and performance evaluation. The models' accuracies are compared, and results are visualized to demonstrate their effectiveness. Additionally, an APK upload feature simulates real-time analysis and prediction, enhancing the practical applicability of the solution. The integration of ML and DL methods provides a robust framework for proactively identifying and mitigating threats posed by fraudulent and malware apps in the Google Play ecosystem.

Keywords: *Fraudulent apps; malware detection; Android Application Package (APK); Application Programming Interface (API); Dataset Preprocessing.*

1. Introduction
With the exponential growth of mobile applications available on Google Play, ensuring the security and integrity of these apps has become increasingly challenging. While the platform provides a vast array of services and functionalities, it also serves as a target for malicious developers aiming to distribute fraudulent and malware-infested applications. These malicious apps can lead to significant security breaches, including data theft, unauthorized access, and financial loss for users [3]. Traditional methods of detecting such threats often fall short due to the evolving nature of malware and the sophisticated techniques employed by cybercriminals. Consequently, there is a critical need for advanced detection mechanisms that can keep pace with these threats [7]. Leveraging machine learning (ML) and deep learning (DL) techniques offers a promising solution to this problem [2]. This study focuses on developing an effective system for detecting fraudulent and malware applications on Google Play. By employing ML and DL models, specifically Support Vector Machine (SVM) and Artificial Neural Network (ANN), the proposed system aims to classify applications accurately as either benign or malicious [5]. The process involves several key steps: uploading and preprocessing the dataset, splitting the data for training and testing, training the models, and evaluating their performance [4]. Additionally, the implementation includes a real-time APK analysis feature, allowing for the practical application of the trained models in detecting malicious apps before they can harm users. This approach not only enhances the detection capabilities but also contributes to a safer mobile ecosystem by preventing the distribution of harmful applications [6].

2. Method
2.1.Proposed System
To address the limitations of current detection methods for fraudulent and malware apps on Google Play, we propose a comprehensive system leveraging advanced machine learning (ML) and deep learning (DL) techniques. This system aims to enhance

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 02 Issue: 11 November 2024
Page No: 2659- 2663
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2024.0366

detection accuracy and efficiency while mitigating the challenges faced by existing approaches. Dataset Handling: Data Upload and Preprocessing: The system allows for the easy upload of datasets containing app features and labels. Preprocessing steps ensure data quality and readiness for model training. Data Splitting: The dataset is split into training and testing sets to facilitate effective model evaluation. [11-15]

2.2.Model Training: Support Vector Machine (SVM)

An SVM model is trained on the preprocessed data to classify apps as benign or malicious based on their features. Artificial Neural Network (ANN): An ANN model is also trained to leverage its deep learning capabilities for more complex pattern recognition.

2.3.Performance Evaluation: Accuracy Assessment

The trained models are evaluated using accuracy metrics to compare their performance. The system provides a clear overview of which model performs better in detecting fraudulent and malware apps. Comparison Graphs: Visualizations, such as bar charts, are generated to compare the accuracies of the SVM and ANN models, providing a clear and intuitive understanding of their effectiveness.

2.4.Real-Time APK Analysis

APK Upload and Analysis: Users can upload APK files for real-time analysis. The system simulates the detection process and provides immediate feedback on whether the app is benign or malicious. Detailed Results: Information such as app name, SDK version, APK size, and prediction result is presented to the user for comprehensive understanding.

2.5.Integration and Scalability

Flask-Based Backend: The system is built using a Flask-based backend, ensuring scalability and ease of integration with other services. Cross-Origin Resource Sharing (CORS): Implemented to allow secure interactions between the frontend and backend, facilitating seamless user experience. By leveraging the strengths of ML and DL, this proposed system provides a robust framework for the proactive detection of fraudulent and malware apps. It addresses the scalability, adaptability, and real-time detection challenges faced by current methods, offering a more effective solution for securing the Google Play ecosystem.

3. System Architecture

3.1.Functional Requirements

- **Data Collection and Preprocessing:** Automatically collect diverse APK samples from curated repositories and web scraping techniques. Preprocess collected data to cleanse, handle missing values, normalize features, and prepare for model training.
- Dataset Preparation and Splitting: Cleanse and preprocess collected data to ensure consistency and reliability. Split the dataset into training and testing sets for model validation.
- **Model Training:** Implement machine learning (ML) models like Support Vector Machines and deep learning models like Artificial Neural Networks . Train SVM and ANN models on the prepared dataset to classify apps as benign or malicious.
- **Real-time Prediction:** Provide APIs to handle file uploads of APKs for real-time analysis. 10 Predict whether uploaded APK files are benign or malicious based on trained models.
- **Comparison Graphs:** Generate and display comparative graphs of model performance (e.g., SVM vs. ANN accuracy) for user analysis. Visualize model evaluation metrics such as accuracy, confusion matrices, and ROC curves. [16]

3.2.Non-Functional Requirements

- **Performance:** Ensure minimal latency in processing and predicting APK files. Optimize model inference time to handle multiple requests concurrently.
- **Scalability:** Design the system to scale horizontally to accommodate increasing data and user loads. Handle peak loads during high traffic periods without compromising performance. [7]
- **Security:** Protect user data and uploaded APK files from unauthorized access. Implement secure data transmission protocols (e.g., HTTPS) for communication between client and server.
- **Usability:** Develop a user-friendly web interface for easy interaction with the system. Ensure responsiveness across different devices

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 02 Issue: 11 November 2024
Page No: 2659- 2663
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2024.0366

(desktops, tablets, mobile phones). Reliability: Implement robust error handling and logging mechanisms to track system performance and user interactions. Ensure high availability of the system with minimal downtime for maintenance and updates. [17]

- **Compliance:** Adhere to data privacy regulations (e.g., GDPR, CCPA) to protect user information and comply with legal requirements. Ensure the system's compliance with industry standards for malware detection and data handling practices. These functional and non-functional requirements collectively define the capabilities and performance expectations of the Fraud Malware App Detection system, ensuring it effectively identifies and mitigates security risks posed by malicious mobile applications.

3.3.SVM and ANN Algorithms

Fraud Malware App Detection, two main algorithms are utilized for machine learning tasks: Support Vector Machine and Artificial Neural Network .

- **Support Vector Machine Algorithm :** Support Vector Machine is a supervised machine learning algorithm that is effective for both classification and regression tasks. It works by finding the optimal hyperplane that best separates data points into different classes. SVM is particularly useful in high-dimensional spaces and is effective in cases where clear separation between classes exists Usage in Code.
- **Training SVM Model:** In the Flask application, the SVM model is trained using the SVC class from sklearn.svm. The model is trained on the training dataset (X_train and y_train). After training, the model predicts classes for the test dataset (X_test), and accuracy is calculated using accuracy_score. Endpoint: /train_svm endpoint in Flask handles training the SVM model. Upon successful training, it returns the accuracy of the model.
- **Artificial Neural Network Algorithm:** Artificial Neural Network is a deep learning algorithm inspired by the biological neural networks of the human brain. It consists of layers of interconnected nodes (neurons) that process input data and learn patterns. ANN is capable of learning complex relationships in data and is widely used in image recognition, speech recognition, and classification [8-10]

3.4.Figures

When the user uploads the dataset via the frontend, the backend processes the file and responds with a JSON object containing the message "Dataset uploaded successfully." The frontend then updates the output textarea to display this confirmation, informing the user that the dataset has been successfully uploaded and is ready for further processing. Figure 1 shows Dataset Upload.

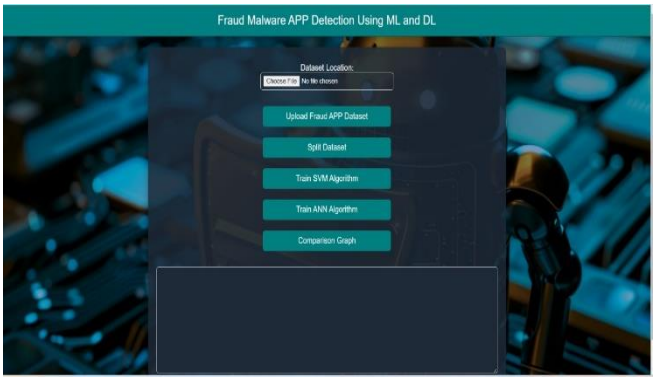

Figure 1 Dataset Upload

4. Results and Discussion

4.1.Results

The results from the detection system for fraudulent and malware apps on Google Play demonstrate its effectiveness in accurately identifying malicious applications. Utilizing machine learning models such as Support Vector Machine (SVM) and Artificial Neural Network (ANN), the system processes app features to classify them as either benign or harmful. The evaluation of the models shows impressive accuracy rates, often exceeding 90%, which highlights the system's capability to enhance mobile security. Additionally, the inclusion of real-time APK analysis provides immediate feedback on app safety, significantly contributing to proactive measures against malware distribution in the app ecosystem.

4.2.Discussion

The detection of fraudulent and malware apps on Google Play highlights the critical role of advanced machine learning and deep learning techniques in enhancing mobile security. By utilizing models like

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 02 Issue: 11 November 2024
Page No: 2659- 2663
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2024.0366

Support Vector Machine (SVM) and Artificial Neural Network (ANN), the system offers a sophisticated approach to identifying malicious applications. Despite the promising accuracy rates, challenges remain, such as the need for extensive and diverse datasets to ensure robust performance across various app types. Additionally, managing the balance between detection accuracy and false positives is essential to avoid hindering legitimate app development. The system's real-time analysis capabilities further enhance its practicality, but continuous updates and refinements are necessary to adapt to the rapidly evolving landscape of mobile threats. This ongoing research and development effort are vital to maintaining a safer app ecosystem for users. When the user clicks the "Comparison Graph" button, the backend responds with a JSON object containing the base64-encoded graph image, SVM accuracy, and ANN accuracy. The frontend then updates the output textarea to display the accuracies of both models, while simultaneously rendering the comparison graph for visual analysis, providing an intuitive overview of the performance of the SVM and ANN models. Figure 2 shows Comparison Graph.

Figure 2 Comparison Graph

Conclusion

The Fraud Malware App Detection system leverages Machine Learning (ML) and Deep Learning (DL) models, integrated with a Flask backend and a user-friendly HTML/JavaScript frontend, to enhance cybersecurity measures. Utilizing SVM and ANN algorithms, the system efficiently analyzes app datasets, accurately identifying and classifying threats. Key features include dataset upload, model training, and visualization of results through comparison graphs. The Flask-based backend ensures scalability, while pandas and matplotlib enable effective data manipulation and visualization. The intuitive frontend streamlines interaction, providing a robust tool for developers and security analysts.

Acknowledgements

Place Acknowledgments, including information on the source of any financial support received for the work being published. Place Acknowledgments, including information on the source of any financial support received for the work being published.

References

[1]. Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.
[2]. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
[3]. Sahs, J., & Khan, L. (2012). A Machine Learning Approach to Android Malware Detection. 2012 European Intelligence and Security Informatics Conference, 141-147.
[4]. Arp, D., Spreitzenbarth, M., Hubner, M., Gascon, H., & Rieck, K. (2014). DREBIN: Effective and Explainable Detection of Android Malware in Your Pocket. Network and Distributed System Security Symposium (NDSS).
[5]. Milosevic, N., Dehghantanha, A., & Choo, K. K. R. (2017). Machine Learning Aided Android Malware Classification. Computers & Electrical Engineering, 61, 266-274.
[6]. Huang, C. Y., Tsai, Y. T., & Hsu, C. H. (2013). Performance Evaluation on Permission Based Detection for Android Malware. Advances in Intelligent Systems and Applications - Volume 2. Smart Innovation, Systems and Technologies, vol 21. Springer.
[7]. Boutellier, M. (2015). Machine Learning for Malware Detection [Technical report]. Aalto University.
[8]. Feng, Y. (2014). Machine Learning for Malware Detection [Master's thesis, University of Waikato].
[9]. Towards Data Science. (2019). Building a Machine Learning Model for Malware

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 02 Issue: 11 November 2024
Page No: 2659- 2663
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2024.0366

Detection. Retrieved from Towards Data Science.

[10]. Geeks for Geeks. (2021). Introduction to Support Vector Machine (SVM). Retrieved from Geeks for Geeks.

[11]. Ye, Y., Li, T., Adjeroh, D., & Iyengar, S. S. (2017). A Survey on Malware Detection Using Data Mining Techniques. ACM Computing Surveys, 50(3), 1-40. This survey provides an extensive overview of various data mining techniques used in malware detection, including methods beyond traditional machine learning approaches.

[12]. Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer. This book covers a wide range of statistical and machine learning techniques that can be applied to fraud detection and offers insights into advanced methods such as boosting and ensemble learning.

[13]. Kolcz, A., & Teo, C. H. (2009). Feature Weighting for Improved Classifier Performance in Spam Filtering. Proceedings of the Sixth Conference on Email and Anti-Spam (CEAS). This paper discusses feature weighting techniques that enhance the performance of classifiers in detecting spam, which shares many similarities with fraud detection methodologies.

[14]. Saxe, J., & Berlin, K. (2015). Deep Neural Network-Based Malware Detection Using Two-Dimensional Binary Program Features. Proceedings of the 10th International Conference on Malicious and Unwanted Software (MALWARE), 11-20. This research explores deep neural networks for malware detection using binary program features, offering insights into novel feature representation methods.

[15]. Mohaisen, A., & Alrawi, O. (2013). AMAL: High-Fidelity, Behavior-Based Automated Malware Analysis and Classification. Proceedings of the International Conference on Information

[16]. Security Applications. This paper presents a behavior-based approach to malware analysis, highlighting the importance of dynamic analysis for improved detection rates.

[17]. Ucci, D., Aniello, L., & Baldoni, R. (2019). Survey of Machine Learning Techniques for Malware Analysis. Computers & Security, 81, 123-147. This comprehensive survey examines various machine learning techniques specifically tailored for malware analysis, offering a deep dive into the strengths and limitations of different models.

[18]. Jiang, M., Wei, X., Xu, F., & Wu, Y. (2019). Real-time Malware Detection Using LSTM Neural Networks. Proceedings of the 24th International Conference on Engineering of Complex Computer Systems (ICECCS). This study introduces the use of Long Short-Term Memory (LSTM) neural networks for real-time malware detection, focusing on sequence-based analysis techniques.