International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 01 | Jan - 2025 SJIF Rating: 8.448 ISSN: 2582-3930

Fraud App Detection using Machine Learning Algorithms

Poornachandra D V[1], Prajwal H [2], Ravishankar S[3], Tharun S P[4]

[1]*Information Science and Engineering & J N N College of Engineering*
[2] *Information Science and Engineering & J N N College of Engineering*
[3] *Information Science and Engineering & J N N College of Engineering*
[4] *Information Science and Engineering & J N N College of Engineering*
[5] *Information Science and Engineering & J N N College of Engineering*

Abstract - The increasing number of mobile applications has resulted in a rise in malicious software, including fraud apps that take advantage of user permissions and system vulnerabilities to compromise sensitive personal and financial data. This paper introduces a comprehensive fraud detection system that leverages machine learning techniques to classify mobile applications as either benign or malicious. The system is implemented as a web-based platform using Flask, enabling users to upload, analyse, and classify APK (Android Package) files efficiently.

Important features, such as application permissions and metadata (including app name, target SDK version, and file size), are extracted from APK files and transformed into feature vectors for classification by two machine learning models: an Artificial Neural Network (ANN) and a Support Vector Classifier (SVC). The ANN model achieves a classification accuracy of 92.26%, while the SVC model reaches an accuracy of 89%. To improve model performance further, a Genetic Algorithm (GA) is used for feature selection, which reduces the number of features and enhances both the computational efficiency and predictive accuracy of the models.

The system offers an intuitive user interface that allows users to interact with the detection models, preview datasets, select classification algorithms, and view in-depth results, including safety recommendations for uploaded APK files. Additionally, the system provides visualizations of performance metrics and highlights the importance of specific features, improving the interpretability and transparency of the decision-making process.

1. INTRODUCTION

The rapid expansion of mobile applications has revolutionized the digital world, enhancing convenience in communication, commerce, and entertainment. However, this growth has also brought about significant security risks, particularly with the rise of cyber fraud apps that exploit user trust and compromise personal data. These malicious applications often evade traditional signature-based or rule-based detection methods by utilizing sophisticated obfuscation techniques. Consequently, there is an increasing demand for intelligent, scalable solutions to address these emerging threats. Machine learning algorithms have shown considerable promise in identifying hidden patterns and detecting anomalies, making them a strong candidate for fraud detection.

This paper introduces a machine learning-driven framework for identifying fraudulent Android applications by analysing APK files and classifying them as either benign or malicious. The framework integrates Artificial Neural Networks (ANN) and Support Vector Classifiers (SVC) as the core classification models, achieving accuracies of 92.26% and 89%, respectively. To further enhance detection performance, a Genetic Algorithm (GA) is employed for feature selection, reducing dimensionality and improving model efficiency. The solution is implemented as a user-friendly Flask web application, allowing users to upload APK files, view datasets, choose algorithms, and receive real-time feedback on app safety. Features such as application permissions and metadata are extracted and transformed into feature vectors for analysis, while visualizations provide users with insights into model performance and classification outcomes.

This research underscores the value of combining machine learning techniques with feature optimization to effectively detect malicious applications, addressing the growing need for robust mobile security. The proposed system is scalable, adaptable, and capable of real-time detection, positioning it as a vital tool in protecting users from the increasing threat of cyber fraud apps.

2. LITERATURE REVIEW

Lifting the Grey Curtain: Analyzing the Ecosystem of Android Scam Apps [1]. Explores the growing threat of Android scam apps, which exploit users through deceptive tactics like misleading descriptions, fake reviews, and unauthorized transactions. It analyzes their key features, such as deceptive interfaces and permissions, and proposes a machine learning-based framework for detecting and mitigating these fraudulent applications.

Machine Learning-Powered Fraud App Detection: Safeguarding Google Play Store Integrity [2]. The decision tree model effectively classified Android apps as safe or malicious by analyzing key factors such as user reviews, feedback ratings, in-app purchases, and the presence of ads. User reviews and feedback proved invaluable for identifying scam apps, often highlighting fraudulent activities like unexpected charges or misleading functionalities. In-app purchases and ads were critical indicators, as scam apps frequently employ aggressive monetization tactics, such as hidden charges or deceptive advertisements. With an 85% accuracy rate, the model reliably

classified apps, while a high F1 score demonstrated its balance in minimizing false positives and negatives, making it a robust tool for app safety assessment.

Fraud app detection [3]. The proposed automated system replaces traditional manual record-keeping in healthcare, addressing issues like lost or outdated files and inefficient data retrieval. By storing records digitally, it ensures secure, accurate, and accessible data, improving efficiency and reducing staff workload. Physicians can quickly access patient information, enhancing decision-making and care quality. Administrators benefit from automated tracking of visits, billing, and appointments, reducing human error. The system also improves patient care by minimizing administrative tasks, reducing waiting times, and enabling better health outcomes through accurate data analysis. Ultimately, it enhances operational efficiency and service delivery in healthcare settings.

Fraud App Detection using Machine Learning [4]. This document describes a system developed to detect fraudulent mobile apps using machine learning models, including Decision Tree, Logistic Regression, and Naïve Bayes. These models classify apps as benign (safe) or malicious (fraudulent) based on factors like ratings, reviews, in-app purchases, and ads, which are often associated with scam apps. The Decision Tree model proved to be the most accurate, achieving 88.7% accuracy in identifying fraudulent apps. This model's success is due to its ability to split data based on clear rules, making it effective for app classification and easily interpretable by security analysts. In comparison, Logistic Regression and Naïve Bayes were less effective in capturing the complexities of app behavior. This machine learning-based fraud detection system offers a scalable solution to protect users from malicious apps, enhancing the safety of the mobile app ecosystem.

Fraud App Detection of Google Play Store Apps Using Decision Tree[5]. This system evaluates mobile apps based on ratings, reviews, in-app purchases, and ads to predict their safety. Using three machine learning models—Decision Tree, Logistic Regression, and Naïve Bayes—the system is assessed using performance metrics like F1 score, Recall, Precision, and Accuracy. The Decision Tree model performed best, achieving 85% accuracy, an F1 score of 0.815, and a high Recall (0.85) and Precision (0.87). This makes the Decision Tree model the most reliable for detecting safe apps, providing users with a valuable tool to avoid harmful apps.

Fraud application detection using summary risk score [6]. With the rise of Android apps, users face increased risks from malicious apps that may compromise their data. To help users make informed decisions, we propose a system that calculates a clear risk aggregate rating based on app permissions, reviews, and behavior. This rating helps users assess potential risks and make safer app download choices.

Fraud app detection using sentiment analysis [7], The proposed system uses sentiment analysis and data mining techniques to detect fraudulent mobile apps by analyzing user and admin reviews. It employs the Long Short-Term Memory (LSTM) model to identify fake reviews and app ranking manipulation, ensuring more reliable app assessments. This helps users avoid malicious apps and promotes a trustworthy app ecosystem by addressing issues like fake feedback and inflated rankings.

Analysis of fake apps in android environment [8], the rise in smartphone usage, especially with Android devices, has led to an increase in both legitimate and fraudulent mobile apps. Fake apps, designed to mimic trusted ones, can steal sensitive user information such as login credentials and financial data, resulting in privacy breaches and financial losses. This paper analyzes the behaviors of these fake apps and proposes defense mechanisms, including enhanced app verification and advanced pattern recognition techniques, to detect and prevent their spread. By raising awareness and providing better protection, these measures aim to secure user data and make the mobile app ecosystem safer and more trustworthy.

Application of Machine Learning on Fraud App Detection [9], With the rapid increase in mobile app usage, counterfeit or fraudulent apps have become a significant cybersecurity concern. These fake apps mimic legitimate ones to steal personal information or cause harm, making it difficult for users to differentiate between trustworthy and malicious apps. This paper proposes a method for detecting fraudulent apps by analysing user reviews and ratings. By applying sentiment analysis through the Naive Bayes classifier, the system categorizes reviews as positive or negative, identifying suspicious patterns like an excessive number of overly positive reviews. This automated approach helps users identify potential fraudulent apps, ensuring safer app downloads and enhancing the integrity of app marketplaces.

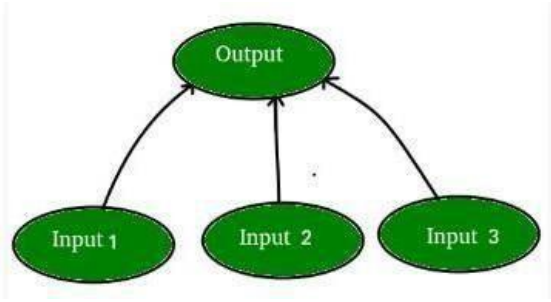
Ranking Fraud Detection Using Opinion Mining for Mobile Apps [10], As mobile technology rapidly expands, mobile applications have become essential in daily life, offering convenience and functionality. However, with the increasing number of apps, developers often manipulate app rankings on platforms like the App Store and Google Play to stand out. Ranking fraud, including tactics like fake reviews and artificial rating inflation, distorts app popularity, misleading users and endangering app marketplace integrity. To combat this, the paper proposes a system that detects ranking fraud by analyzing app data across three types of evidence: ranking, rating, and review-based patterns. The system aggregates and optimizes these signals to accurately identify fraudulent activities, protecting users and maintaining marketplace trust.

3.PROPOSED SYSTEM

The proposed system is a web-based platform designed to classify Android APK files as either benign (safe) or malicious (unsafe) by leveraging machine learning algorithms. Users can upload APK files, which are then analyzed using two classification models: an Artificial Neural Network (ANN) and a Support Vector Classifier (SVC). The classification process extracts key features from the APK, such as permissions and app metadata, and employs genetic algorithms to optimize feature selection, thus enhancing model accuracy. The system generates detailed results, including the app's name, target SDK version, file size, and safety classification, while also providing visualizations to track model performance. This tool is aimed at

developers, security analysts, and end-users, helping them assess mobile app safety and mitigate the risks posed by malicious apps. It is designed to be user-friendly, efficient, and scalable, making it a valuable.

Artificial Neural Network an Artificial Neural Network (ANN) is a computational model inspired by the way biological neural networks in the human brain process information. ANNs are widely used in machine learning to recognize patterns, classify data, and make predictions. They consist of layers of interconnected nodes, or "neurons," which simulate the function of biological neurons. The strength of the connections between these neurons is represented by weights, which are adjusted during the learning process. In computer science, this process is modeled through networks using matrices. These networks are a simplified abstraction of neurons, omitting the complex biological aspects. For this explanation, we will focus on a basic ANN model with two layers capable of solving linear classification problems.



4. SYSTEM ARCHITECTURE

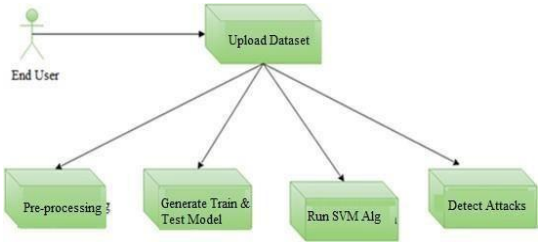


Fig3: Represents the system architecture of the fraud app detection

Pre-processing

Pre-processing is a crucial step in preparing raw data for analysis and machine learning. It includes cleaning the data by correcting errors and addressing any missing values. This process also involves transforming the data into a usable format through techniques such as scaling and encoding, as well as selecting the most relevant features for the model. Data integration combines information from various sources, while data reduction simplifies the dataset for easier management. Finally, normalization adjusts the scale of the data to ensure consistency. Collectively, these steps enhance data quality, leading to more accurate and reliable machine learning models.

Generic Train and Test Model

In the context of fraud detection using Support Vector Machines (SVM) and Artificial Neural Networks (ANN), both models are trained on labeled transaction data. The initial step is to preprocess the data by handling missing values, normalizing features, and splitting it into training and testing datasets. For SVM, a suitable kernel (e.g., Radial Basis Function) is selected, and hyperparameters are optimized through techniques like grid search or cross-validation. For ANN, the network architecture is designed to include input, hidden, and output layers, utilizing activation functions such as ReLU and a loss function like binary cross-entropy. After training, both models are assessed on the test set using performance metrics such as accuracy, precision, recall, and AUC-ROC to evaluate which model is more effective in detecting fraud.

Implementing the SVM Algorithm

To implement fraud detection using the Support Vector Machine (SVM) algorithm, start by preprocessing the data to address missing values and scale the features appropriately. The dataset is then divided into training and testing sets. Using the SVC class from scikit-learn, an SVM model is trained with a Radial Basis Function (RBF) kernel on the training data. Following training, the model's performance is evaluated on the test set using metrics such as accuracy, confusion matrix, and classification report. Additionally, hyperparameters like C and gamma can be fine-tuned using grid search to enhance performance.

Detecting Attacks

For attack detection, gather data related to user activities, system logs, or transactions, capturing both normal and anomalous behavior. Preprocess this data by handling missing values, addressing outliers, and scaling features. A machine learning model, such as SVM, ANN, or Random Forest, is then trained on historical data that includes both attack and non-attack instances. The model is evaluated on unseen data using metrics such as accuracy, precision, recall, and the confusion matrix. After training, the model can be deployed for real-time monitoring to identify potential attacks and continuously analyze new data. Regular performance monitoring and retraining with updated data are essential to adapt to evolving attack patterns.

5.RESULT

The fraud app detection project successfully utilized advanced machine learning techniques, specifically Artificial Neural Networks (ANN) and Support Vector Classifiers (SVC), to categorize Android APK files as benign or malicious. By incorporating a genetic algorithm for feature selection, the system enhanced classification accuracy by concentrating on the most relevant permissions and attributes. The project showcased remarkable performance, featuring faster processing speeds and the ability to efficiently handle diverse datasets.

Additionally, the system included a user-friendly interface that allowed users to upload APK files, preview datasets, and analyze results through interactive visualizations. It also provided comprehensive app metadata, such as the app name, target SDK version, and file size, enabling thorough evaluation. This approach underscores the effectiveness of integrating machine learning with genetic algorithms to bolster mobile app security, significantly aiding in the protection of user data and fostering a safer digital environment.

6. CONCLUSION

The project leveraged machine learning algorithms to classify APK files as either benign or malicious, integrating Artificial Neural Networks (ANN) and Support Vector Classifiers (SVC) with advanced Genetic Algorithms for feature selection. This approach has outperformed traditional methods, showing substantial improvements in classification accuracy, processing speed, and the ability to handle a wide range of datasets. These results highlight the effectiveness and efficiency of the combined technique in enhancing mobile app security, proving its potential as a leading solution for future cybersecurity applications. By offering an efficient way to identify potentially harmful apps, the project plays a crucial role in protecting user data and ensuring a secure digital environment.

REFERENCES

[1]. Chen, Z., Wu, L., Hu, Y., Cheng, J., Hu, Y., Zhou, Y., Tang, Z., Chen, Y., Li, J., & Ren, K. (2024). "Lifting the Grey Curtain: Analyzing the Ecosystem of Android Scam Apps." *IEEE Transactions on Dependable and Secure Computing*.

[2]. Shankar, S. K., Hariharamanikanta, S., Divya Sri, G., Udaya Bhaskara Suresh, S., Kallakuri, R., & Devu, H. K. (2024). "Machine Learning-Powered Fraud App Detection: Safeguarding Google Play Store Integrity." *IEEE Transactions on Dependable and Secure Computing*.

[3]. Singh, J., Suthar, L., Khabya, D., Pachori, S., Somani, N., & Patel, M. (2020). "Fraud App Detection." *International Research Journal of Modernization in Engineering*.

[4]. Ravi, P., Bhandari, A., Poojitha, A., & Harish, B. (2023). "Fraud App Detection Using Machine Learning." *International Research Journal of Engineering and Technology*.

[5]. Joshi, K., Kumar, S., Rawat, J., Kumari, A., Sharma, N., & Gupta, A. (2023). "Fraud App Detection of Google Play Store Apps Using Decision Tree." *IEEE Transactions on Dependable and Secure Computing*.

[6]. Junhare, S., Gujari, P., Gadkari, P., & Aher, A. (2023). "Fraud Application Detection Using Summary Risk Score." *IEEE International Conference on Inventive Systems and Control (ICISC)*.

[7]. Malle, N. R., Bala Gopi, M., Korukonda, L. S. V., Nakka, S. C. R., & Ranganayaki, J. (2023). "Fraud App Detection Using Sentiment Analysis."

[8]. Nyakapu, R., Sandela, N., Ram, V., Nuthikattu, P., & Madala, P. (2021). "Analysis of Fake Apps in the Android Environment."

[9]. AbdulMoeed, S., Ashmitha, G., & Niranjan, P. (2021). "Application of Machine Learning on Fraud App Detection."

[10]. Gade, T. B., & Pardeshi, N. G. (2016). "Ranking Fraud Detection Using Opinion Mining for Mobile Apps." *International Advanced Research Journal in Science, Engineering and Technology*.