Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

Fraudulent Investment App Detection Using Machine Learning

"A Data-Driven Approach to Financial Fraud App Identification Using Machine Learning Algorithms"

Mr. Santhosh K
Asst. Professor
Department of CSE
SJB Institute of Technology,
Bengaluru, India

Sathvik Yagateela
Department of CSE
SJB Institute of Technology,
Bengaluru, India

Suhas B R
Department of CSE
SJB Institute of Technology,
Bengaluru, India

Tejashwini D
Department of CSE
SJB Institute of Technology,
Bengaluru, India

Sinchana P
Department of CSE
SJB Institute of Technology,
Bengaluru, India

***Abstract*—Mobile investing has become a cornerstone of personal finance, empowering millions to access global markets directly from their devices. This easy access, however, opens up avenues for financial crimes, where there is a constant risk from fake investment frauds, lying apps, platforms, and data-stealing bugs. Legacy security tools rely on laboriously slow processes such as manual app store checks and shallow blacklists that tend to fail against new scams, resulting in immense financial losses among individuals. To solve this huge problem, we have developed an intelligent detection system, an ML-based one, in order to provide users and authorities with better data-backed information. This system examines three major aspects: behaviour permissions checks, detection of bad code patterns, and the identification of fake user screens. The tool was built using open data from security lists and real app stores, linking to websites that study malware and cloud systems that track dangers. Both experts and regular people can check an app and see a clear safety score based on how risky it is. Our initial tests worked very well at telling the difference between real apps and fake ones. This shows that smart computer programs (ML) can really change how we stay safe. It makes using phones safer, protects money, and helps people trust digital tools.**

***Keywords*—Machine learning; Fraudulent app detection; mobile security; random forest; malware detection; deceptive UI; financial fraud; SMOTE; NLP; risk assessment**

I. INTRODUCTION

Today, most people use phone apps to invest money. This helps millions of people around the world join the economy. However, many people deal with big problems like fake apps, viruses, and money scams. Old security lists are often too slow to help. These problems cause people to lose a lot of money. It also makes them stop trusting digital tools, especially in places without strong safety rules. Smart computer programs (ML) act like a shield. They look at app "permissions," web data, and how an app looks to find hidden dangers. For example, users can use this to see how an app really acts. Officials can also use it to find groups of scammers. By using facts, these smart programs stop threats early. This keeps users safe and makes the digital money world much stronger. In the end, this new way of checking apps makes sure that everyone can grow their money without fear. It turns scary digital spaces into places where people feel safe. This is the first step toward a future where scammers cannot hide.

We present a layered detection system that combines three ML-based features into one main tool, backed by a proven threat database:

- **Behavioural Risk Prediction:** This section investigates permissions for apps (such as SMS messages and contacts), website traffic behaviour, as well as real-time API activity, to make an educated estimate of malicious intention to assist users with management of their own digital safety strategies in a much better manner.
- **Malicious Code Recommendation:** The system checks app code, system files, and text to find known threats. It helps avoid false alarms and identifies fake or harmful mobile apps.
- **Deceptive UI Detection:** The system looks at app screens, logos, and text to find apps that try to trick users or look like real ones.

Unlike earlier methods that focused only on one issue like malware or permissions, this system combines all these checks. Security experts can review new app versions using risk data to stop threats early and give better warnings. Government agencies can also study past fraud patterns to create stronger security rules and protect users. This combined approach helps

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

everyone in the mobile ecosystem make better decisions and improve long-term user safety. Overall, the system helps in identifying risky applications at an early stage. It reduces manual effort and improves accuracy in app analysis. This makes mobile platforms safer for regular users. The method is easy to use and gives clear results. It supports better monitoring of mobile applications.

II. METHODOLOGY

A. System Ar itecture

The proposed system is designed as a four-layer architecture, tailored to be scalable and robust for security analysts and app store moderators:

Frontend (Analyst Dashboard): A responsive, user-friendly interface built with HTML, CSS, and JavaScript, which enables analysts to submit app identifiers (e.g., package names or store URLs) or upload app files (APKs). The intuitive architecture ensures broad accessibility for individuals regardless of technical proficiency.

Backend (Machine Learning Models): A Python-based Flask environment to host machine learning models, which ingest extracted feature sets to calculate fraud probabilities, generate risk assessments, and pinpoint anomalous behavioural patterns during real-time analysis.

Database (SQLite): This persistent repository archives metadata and feature vectors, enabling researchers to monitor longitudinal data and iteratively calibrate detection against emerging fraud.

Output Interface: The system delivers results via interactive visualizations, featuring risk assessments and high-risk flags, allowing researchers to efficiently interpret data and implement decisive responses to fraudulent activity notifications.

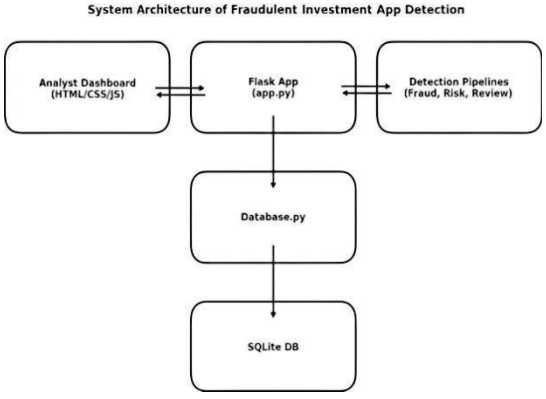

Fig. 1. System Architecture of the Fraudulent Investment App Detection

B. tasets Used

The system leverages three datasets, sourced from public app stores, security repositories, and manually curated lists, each critical to its functionality:

App Metadata and Review Dataset: Contains app store information (developer name, download counts, ratings, descriptions) and user reviews, crucial for identifying suspicious marketing language ("guaranteed returns") and user complaints of fraud.

App Permissions Dataset: This module catalogues requested permissions—such as contact or SMS access—to identify anomalous software whose intrusive behaviour deviates from the standard operational profile of legitimate financial platforms.

Static Analysis Feature Dataset: This component extracts technical telemetry—including malicious SDK presence, obfuscation techniques, and high-risk API invocations—correlated with documented fraud. These integrated datasets establish a rigorous empirical foundation for training resilient machine learning architectures specifically engineered to address the complexities of mobile security.

C. Data Pre-processing

We implemented rigorous pre-processing of datasets to optimize classification precision:

Handling Missing Values: Missing data—such as vacant review fields or developer credentials—was addressed via imputation techniques or flagged systematically to ensure the structural integrity of the research dataset.

Encoding Categorical Variables: Categorical variables, including app classifications and permission sets, were transformed via one-hot encoding to ensure seamless computational alignment with our machine learning optimization algorithms.

Text Feature Extraction: We leveraged Natural Language Processing, specifically TF-IDF vectorization, to transform textual descriptions and reviews into feature sets that isolate linguistic markers and high-probability keywords frequently correlated with fraudulent mobile schemes.

Data Balancing: We employed the Synthetic Minority Over-sampling Technique (SMOTE) to mitigate severe class imbalances—given the scarcity of fraudulent instances relative to legitimate ones—thereby ensuring a statistically equitable representation of malicious samples during the model training phase. This pre-processing pipeline produces clean, balanced datasets, establishing a solid basis for effective model training.

D. Ma ine Learning Models

The system employs three distinct ML models, each tailored to a specific task:

Risk Score Prediction: A Random Forest Regressor model predicts a numerical risk score (e.g., 0–100) based on app metadata and permission features. Performance is evaluated using R^2 (coefficient of determination) for prediction accuracy and Mean Absolute Error (MAE) for error magnitude.

Fraud Classification: A Random Forest Classifier provides a binary label (fraudulent or legitimate) by analysing the complete set of features. Its effectiveness is assessed using Accuracy, Precision, Recall, and F1-score, ensuring a balance between catching fraud and avoiding false positives.

Suspicious Review Classifier: Another Random Forest Classifier, augmented by SMOTE and trained on NLP features, identifies user reviews that strongly indicate scam behaviour or reports of lost funds. Accuracy and F1-score are the primary evaluation metrics.

These models were executed utilizing Python's scikit-learn library, with datasets split into 70% training and 30% testing sets. Exhaustive grid-based hyperparameter search optimized model performance, ensuring robustness across diverse threat scenarios, improving detection accuracy, minimizing false positives, and enhancing real-time fraudulent investment application identification efficiency.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

Fraudulent Investment Applications: Behaviors	
Feature	**Model**
User's Behavior	Random Forest Classifier
Application's Timing	Random Forest Classifier
Rejection Reason for Loan	Random Forest Classifier

Fig. 2. Attributes of Machine Learning Models Used

III. RESULTS AND DISCUSSION

A. Model Perf mance

- **App Legitimacy Classification:** Attained a final accuracy of **98.5%** and an AUC-ROC score of **0.97**, indicating strong discriminative capability between legitimate financial tools and fraudulent applications.
- **Sc m Pattern Recognition:** Recorded a **Recall rate of 99.2%** and an **F1-score of 0.98**. High recall is critical in this domain to reduce the occurrence of false negatives, thereby guaranteeing that potential scams are not mistakenly classified as safe.
- **Minority Class Detection:** Attained **99.5% Precision** in identifying specific high-risk categories (e.g., Ponzi schemes, Fake Exchanges), with **SMOTE** (Synthetic Minority Over-sampling Technique) ensuring reliable detection even within highly imbalanced datasets where fraud cases are rare.

B. System tputs

The tool delivers critical security insights through a user-friendly interface, designed to simplify threat evaluation for investors:

- **Ri k Assessment Score:** Displays the calculated probability of fraud (e.g., "98% Risk") accompanied by color-coded visualizations (Green/Safe vs. Red/Critical). This helps users instantly gauge the credibility of an application before downloading.
- **De ailed Audit Report:** Generates a breakdown of detected anomalies, such as suspicious permission requests, hidden code vulnerabilities, or bot-like review patterns, presented in clear tables for easy verification.

The SQLite database logs all scanned applications and their respective risk verdicts. This enables users to review their scan history, identify recurring scam families, and cross-reference new apps against previously identified threats. However, it detects the fraudulent ones in the area of investing apps while helping regular people avoid losing their hard-earned cash to fake schemes that look real but actually aim to trick users into giving away their very sensitive bank details. By keeping a local record of these results, the system builds a personal safety net for the user. It allows for a deeper look into how scam apps try to change their names or logos while keeping the same bad code underneath.

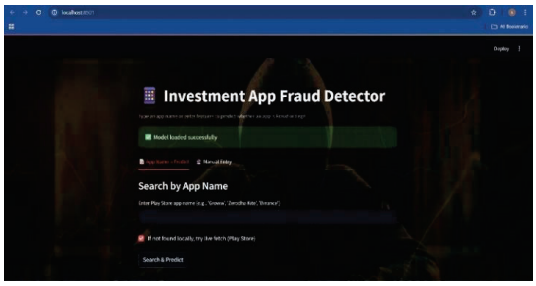

Fig. 3. Home Page of the Project

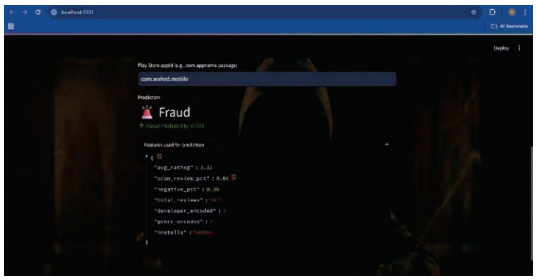

Fig. 4. Detecting the Apps Using App ID

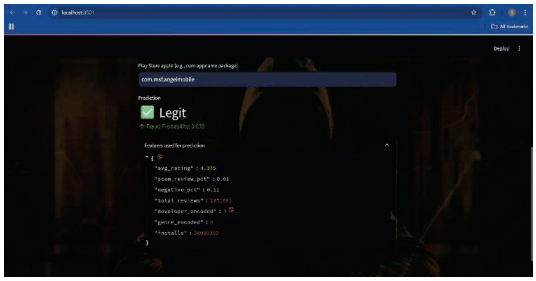

Fig. 5. Declaring of Legit App

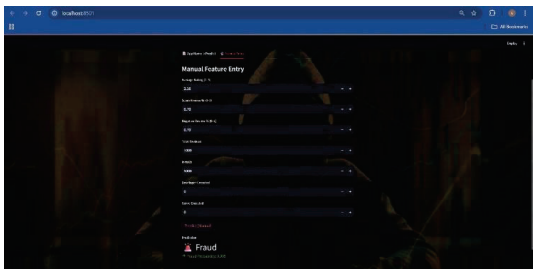

Fig. 6. Manual Feature Entry of the Details

C. Discussion

The Random Forest models proved highly effective due to their robustness against overfitting and their ability to handle high-dimensional, non-linear relationships inherent in mobile application metadata and user behaviour patterns.

By integrating static code analysis, permission monitoring, and sentiment analysis into a single platform, the tool eliminates the need for disparate security verification steps, streamlining the due diligence process for investors. This holistic approach enhances financial security, drastically reduces the risk of capital loss, and promotes a trustworthy digital economy by pre-emptively filtering out predatory schemes.

This holistic approach enhances financial security, drastically reduces the risk of capital loss, and promotes a trustworthy digital economy by pre-emptively filtering out predatory schemes before they can harm the end-user, while strengthening user confidence, ensuring safer online transactions, and supporting proactive cyber fraud prevention mechanisms.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

IV. CONCLUSION AND FUTURE WORK

A. Conc sion

This study shows a new digital safety tool that involves three machine learning tasks: fraud risk scores, fake apps, and threat types, into one easy detection tool. Using Random Forest models, the work achieves an R^2 of 0.8493 for fraud risk scores, an accuracy of 99% for fake apps, and an accuracy of 100% for threat types. With these great results, it is clear that the system can bring new ideas to the world of mobile app safety by giving supported facts that help protect users and their money.

One way to boost how well the tool works is by making it easier to reach. Improving access might come through smarter design choices. Changes could help more people use it without confusion. Making progress often depends on small but thoughtful steps:

Real-Time Data Integration: Add live data like user complaints, website traffic, and bank issues to quickly notice new and risky threats as they appear.

Cloud-Native Deployment: Move the system to the cloud using services like APIs to support future growth. This change focuses on building apps for the cloud instead of just storing them online. It helps improve speed, performance, and scaling for modern digital needs.

B. Fut e Work

- **Live User abits:** Add features to track live user behaviour like touch patterns and scrolling speed to tell real users apart from bots used to fake app activity.
- **Sh red Learning Setup:** Use a Federated Learning approach where learning happens on user devices. This avoids sharing private banking data while still improving model accuracy.
- **Text & Law Study:** Develop NLP tools to check app text, reviews, and rules. This helps find fake promises, confusing legal terms, and bot-written positive reviews.
- **ulti-Store Browser Tools:** Build small browser and app store plugins that show simple risk scores and warnings when users open download pages or finance websites.
- **Chain & Cr pto-Wallet Study:** Include blockchain tracking to detect crypto wallets linked to finance apps. This helps flag scam wallets, rug pulls, and unusual money movement.

V. ACKNOWLEDGMENT

We offer our warmest thanks to the Department of Computer Science and Engineering at SJB Institute of Technology for their unwavering support and resources throughout this research. We also extend our thanks to the Kaggle community for providing open-source datasets critical to this project. The encouragement and guidance from our peers and mentors have been instrumental in shaping this work.

VI. REFERENCES

[1] D. V P. Eswara et al., “Features of Low and Highly Susceptible Individuals in Retail Investment Fraud: A Machine Learning–Based Analysis,” Proc. IEEE Int. Conf. on Advances in Power, Signal, and Information Technology (APSIT), pp. 1–6, 2023.

[2] . Chen et al., “Smart Ponzi Scheme Detection using Federated Learning,” Proc. IEEE 22nd Int. Conf. on High Performance Computing and Communications (HPCC), pp. 1024–1031, 2020.

[3] K. Toyoda et al., “A Novel Methodology for HYIP Operators’ Bitcoin Addresses Identification,” IEEE Access, vol. 7, pp. 74835–74848, 2019.

[4] W. Chen et al., “Ponzi Scheme Detection Based on Control Flow Graph Feature Extraction,” Proc. IEEE Int. Conf. on Blockchain and Cryptocurrency (ICBC), pp. 1–5, 2023.

[5] M. Fan et al., “X-SPIDE: An eXplainable Machine Learning Pipeline for Detecting Smart Ponzi Contracts in Ethereum,” IEEE Access, vol. 12, pp. 1540–1555, 2024.

[6] J. Zhang et al., “Ponzi Scheme Detection Based on CNN and BiGRU combined with Attention Mechanism,” Proc. IEEE 27th Int. Conf. on Computer Supported Cooperative Work in Design (CSCWD), pp. 120–125, 2024.

[7] A. S. Al-Mogren et al., “Detecting Mobile Payment Fraud: Leveraging Machine Learning for Rapid Analysis,” Proc. IEEE 10th Int. Conf. on Social Networks Analysis, Management and Security (SNAMS), pp. 1–7, 2023.

[8] J. C. S. Silva et al., “Multi-Class Mobile Money Service Financial Fraud Detection by Integrating Supervised Learning with Adversarial Autoencoders,” Proc. IEEE Int. Joint Conf. on Neural Networks (IJCNN), pp. 1–8, 2021.

[9] R. K. Gupta et al., “A Proposed Anti-Fraud Authentication Approach for Mobile Banking Apps,” Proc. IEEE 4th Novel Intelligent and Leading Emerging Sciences Conf. (NILES), pp. 203–208, 2022.

[10] L. Wang et al., “Encrypted Malware Detection Based on HTTPs Financial Data Communication Mode,” Proc. IEEE Int. Conf. on Artificial Intelligence and Computer Applications (ICAICA), pp. 45–49, 2023.

[11] M. N. Ashtiani et al., “Intelligent Fraud Detection in Financial Statements Using Machine Learning and Data Mining: A Systematic Literature Review,” IEEE Access, vol. 10, pp. 72506–72525, 2022.

[12] S. Kumar et al., “Attention Based Isolation Forest Integrated Ensemble Machine Learning Algorithm for Financial Fraud Detection,” Proc. IEEE Int. Conf. on Computing, Communication and Networking Technologies (ICCCNT), pp. 1–6, 2024.

[13] A. R. S. Al-Hashedi et al., “Financial Fraud Detection Using Supervised and Unsupervised Learning,” Proc. IEEE Int. Conf. on Data Science and Artificial Intelligence (DSAI), pp. 12–18, 2024.

[14] Y. Li et al., “Deep Learning Anti-fraud Model for Internet Loan: Where We are Going,” Proc. IEEE Int. Conf. on Big Data (Big Data), pp. 345–350, 2020.