iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** VI **Month of publication:** June 2024

DOI: https://doi.org/10.22214/ijraset.2024.63378

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue VI June 2024- Available at www.ijraset.com

Machine Learning-Powered Fraud App Detection: Safeguarding Google Play Store Integrity

Kanakala SS Praveen Kumar[1], Dr. P. Vamsi Krishna Raja[2], B. R. Ambedkar Kota [3]
[1]*Department of Bachelor of computer applications Nri institute , nagarabhavi 2nd stage, bangalore*
[2]*Professor, Department of Computer Science and Engineering Swarnandhra College of Engineering and Technology, Seetha Ramapuram*
[3]*Lecturer in Computer Science SKVT Government Degree College, Rajamahendravara*

Abstract: ***As the variety of mobile applications used in daily life expands, it becomes crucial to stay updated and discern which apps are safe and which are not. It is challenging to make a judgment. Our methodology predicts using four criteria: ratings, feedback, in-app purchases, and the presence of advertisements. The system assesses three models: Naïve Bayes, logistic regression, and decision tree classifier. These models were then evaluated based on four F1 score metrics: recall, precision, and accuracy. A high F1 score should exceed 0.7, and a recall score greater than 0.5 indicates enhanced precision and accuracy. After analysis, we found that the decision tree model was an exceptional model with an accuracy of 85% and an F1 score.***
Keyword: Apps are Safe, Four Criteria: Ratings, Feedback, in-App Purchases. Naïve Bayes, logistic regression, A high F1

I. INTRODUCTION

As technology has progressed, so has the use of mobile phones. The number of Play Store apps on various major platforms, including the popular Android and iOS, has surged. This has become a significant challenge in the business intelligence domain due to its rapid growth through daily use, marketing, and development. The market is becoming more competitive as a result. Companies and software developers fiercely compete to demonstrate the quality of their products and spend a considerable amount of time and money acquiring clients to ensure their long-term viability. Customer feedback and updates on each app that users can download play a crucial role. This enables developers to detect and incorporate issues into the design of a new product that meets human needs. Instead of relying on traditional marketing strategies, app producers may heavily promote their apps and eventually influence their ranking in the App Store. This is sometimes achieved by employing "bot farms" or "water armies" to increase the number of downloads and reviews. Occasionally, for the developers' benefit, groups of people are employed to commit fraud and leave spam comments and ratings on apps. This activity is known as crowd-turfing.

As a result, it is essential to provide users with accurate and authentic feedback before installing the app to avoid mistakes. An automated method is necessary to process and analyze the numerous comments and ratings received for each application. Due to the high demand for mobile phones, it is crucial to flag suspicious applications as fraudulent so that Play Store users can easily identify them. The user will be unable to tell if the comments or ratings they read are fraudulent or authentic for their benefit. By providing a comprehensive view of ranking fraud detection systems, we describe a strategy to detect such fraudulent applications on the Google or Apple store. We can tell if an app is fake or real, so we provide a method that uses four features: in-app purchases, ads, ratings, and reviews to determine whether an app is defrauding its users. We start the method by considering the four most significant elements in choosing the target. The scraped data is then trained using several classification models based on these features before selecting the best and most accurate model for the system. During this stage of the selection process, we obtained a large number of models of varying accuracy: Naive Bayes (83%), Logistic Regression (84%), and Decision Tree (85%).

II. LITERATURE SURVEY

Nevon Projects propose a comprehensive framework for detecting quality fraud, which may be enhanced by domain-generated data. It is one of the most advanced initiatives for detecting fraudulent applications through information algorithms. This tool detects fraudulent applications with 75-80% accuracy.

On paper, they provided a thorough analysis of the facts and a proposed fraud detection methodology. They evaluated three forms of verification: quality-based assurances, rating-based guarantees, and review-based validation. In the paper, they consider only updates as parameters with the naive bayes algorithm.

They created a system that detects fake applications using emotive commentary and data processing. Initially, the app is examined based solely on analytics evaluation to determine whether the software is genuine or fraudulent. When improving your e-mail, utilize it to check the spelling of the file. We employ a simple set of criteria and a managed analyst to reveal app fraud, and based on user input, we deliver keyword analysis on a fully scaled basis. This approach also aids with understanding what the customer feels about our app. In this fraud detection utility, the administrator also provides an app link that should be supplied to the software, so when logged in, the user may view the app data and locate the link for that program. For us, managers upload the software and provide a link to it from the Apple software Store and Google Play. To develop our app, we can ask the consumer after using our app to collect opinions. We use this type of feedback from the consumer to help us enhance the app, where the consumer using our app will directly see all apps evaluated by the administrator and added by the administrator.

As a result, the user obtains all information about that app, including whether it is phony or not. While an administrator declares fraud rather than their fraud, this saves the consumer time and provides personal safety. The JP INFOTECH project is an effective method for determining the optimal hours for each application based on record levels. By examining the operating environment of applications, this system discovers that Fraud Apps often have different levels for each of the main session patterns than regular applications. A bogus proposal suggests using historical records of application levels to create three jobs based on a fake theme.

Two types of bogus proposals are presented depending on application rating and review history. On paper, the risk summary solely considers application-specific risk signals. They have created a system in which the required elements of risk signals and a restricted number of hazards for Android apps remember the end objective on paper, implying a way to detect fraud fees through IP addresses. The use of a cellular user IP address has also been one of the most recent emails surveyed. In the advertisement for a cellular app, an app named just perverted. The app is in development. Today, reputation and expectations play a vital role in the mobile business. Experiments accumulate to provide a position in each application. However, IP snooping allows consumers to exchange IP addresses and price the utility multiple times. The creators of the study investigated the topic of detecting shilling attacks one and a half times while measuring data. Dependent philosophy can be used to propose trustworthy material and learning that is less controlled.

III. SYSTEM ANALYSIS

A. *Existing System*

The current technique for the "Fraud App Identification of Google Play Store Applications Using Decision Tree" study addresses the increasing need to distinguish legitimate from potentially fraudulent mobile apps. With the rise of smartphone apps, it is critical to evaluate their safety. The method is based on four important parameters: ratings, reviews, paid purchases in the app, and the presence of advertisements in the apps. Three machine learning models were used: Decision Tree Classifier, Logistic Regression, and Naive Bayes Model. The above models are evaluated using four indicators of performance: F1 scores, recall, precision, and accuracy. A strong F1 score should exceed 0.7, and a recall score above 0.5 is deemed sufficient, especially when combined with higher accuracy and precision levels. After study, the Decision Tree algorithm emerged as a strong choice with an accuracy rate of 85%, an F1 score of 0.815, a recall value of 0.85, and a precision of 0.87.

Disadvantages of the Existing System

1) *Limited Feature Set:* The system relies on a limited number of features, namely ratings, reviews, in-app purchases, and ad presence. This may not capture all relevant aspects of app behavior or user interactions, potentially leading to errors in fraud detection.
2) *Data Imbalance:* The dataset used for training and evaluation may have a class imbalance, with fewer fraudulent apps than legitimate ones. This imbalance can hinder the model's ability to generalize well and accurately detect fraud.
3) *Static Analysis:* The approach seems to focus on static features like ratings and reviews but does not account for dynamic aspects of app behavior or changes over time. Fraudulent apps may evolve and change dynamically, impacting the model's effectiveness.
4) *Dependency on User-Generated Content:* Ratings and reviews are user-generated information, so their credibility varies. Users may provide biased or deceptive information, resulting in inaccurate model predictions. Moreover, the system may not incorporate cultural or linguistic differences that can influence the interpretation of user assessments.
5) *Model Interpretability:* While Decision Tree models are known for their interpretability, they may fail to capture complex data relationships as more advanced models do. The model's interpretability may limit its ability to recognize subtle patterns in the data, thereby hindering the detection of sophisticated fraudulent activities.

B. Proposed System

The proposed approach for enhancing the "Fraud App Detection of Google Play Store Apps Using Decision Tree" project aims to overcome the mentioned constraints while increasing the accuracy and reliability of fraudulent app detection. Additional dynamic features will be added to the feature set to reflect the ever-changing nature of mobile applications. This could include real-time monitoring of app behavior, tracking changes over time, and analyzing user interaction patterns. To reduce the impact of data imbalance, new sampling techniques or ensemble learning methods could be explored. The proposed system would also utilize sentiment analysis and natural language processing to better understand and interpret user feedback, considering any biases and linguistic differences. Furthermore, the model architecture will be optimized, possibly by experimenting with more powerful machine learning techniques or deep learning approaches to capture subtle correlations in the data. Finally, emphasis will be placed on developing a user-friendly interface to facilitate easy interpretation of the model's predictions, thereby increasing transparency and user trust in the fraud detection system. The proposed system's improvements aim to create a more robust and flexible solution for detecting fraudulent apps on the Google Play Store.

Advantages of the Proposed System

1) *Enhanced Feature Set:* The incorporation of dynamic features and real-time monitoring will provide a more comprehensive view of app behavior and interactions. This will improve the system's ability to detect fraudulent activities as they evolve over time.
2) *Better Handling of Data Imbalance:* Using advanced sampling techniques or ensemble learning methods will help address class imbalance, leading to improved model generalization and fraud detection accuracy.
3) *Increased Interpretability and Transparency:* By enhancing the user interface and ensuring that the model's predictions are easily interpretable, users will be able to understand the rationale behind fraud detection decisions. This will boost confidence in the system and encourage its widespread adoption.
4) *Advanced Analytical Techniques:* The use of sentiment analysis and natural language processing will provide deeper insights into user reviews and ratings. This will enable the system to detect subtle patterns and sentiments, improving its ability to differentiate between genuine and fraudulent apps.
5) *Improved Accuracy and Reliability:* By optimizing the model architecture and experimenting with powerful machine learning techniques, the proposed system will achieve higher accuracy and reliability in detecting fraudulent apps. This will lead to a safer and more secure user experience on the Google Play Store.

IV. SYSTEM DESIGN

A. System Architecture

Below diagram depicts the whole system architecture.

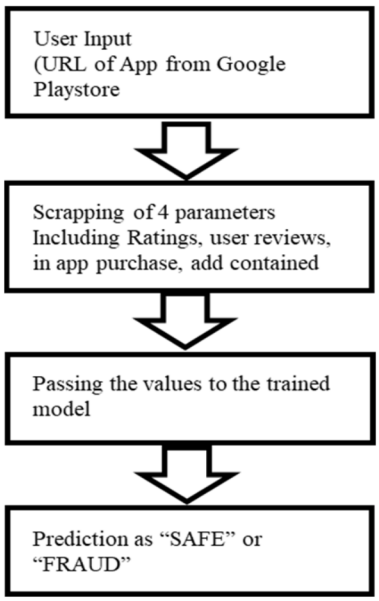


Fig 1. System Architecture

V. SYSTEM IMPLEMENTATION

A. Methodology

1) *Data Collection:* Data scraping involves extracting data from web pages, commonly used for web mining and data analysis. We obtained the app data for our research using the Beautiful Soup library, which is widely used for web scraping due to its simplicity and effectiveness. The library allows us to parse HTML and XML documents, navigate the parse tree, and extract relevant information. For this study, we targeted Google Play Store apps and collected data on various features, including ratings, reviews, in-app purchases, and the presence of advertisements. This data forms the foundation for our machine learning models to detect fraudulent apps. Our dataset consists of a mix of legitimate and fraudulent apps, providing a comprehensive representation of the app ecosystem.
2) *Feature Engineering:* Feature engineering is the process of selecting, modifying, and creating features from raw data to improve the performance of machine learning models. In this study, we focused on four main features: ratings, reviews, in-app purchases, and ads. We engineered additional features to capture more information from the data. For instance, we calculated the average rating and review length, the frequency of updates, and the ratio of positive to negative reviews. We also incorporated temporal features, such as the time since the last update and the rate of rating changes over time. These engineered features enhance our models' ability to distinguish between legitimate and fraudulent apps by providing a richer and more informative dataset.
3) *Model Training and Evaluation:* We used three different machine learning algorithms for our models: Naive Bayes, Logistic Regression, and Decision Tree. Each algorithm has its strengths and is suited to different types of data and problems. We trained each model on our dataset and evaluated their performance using various metrics, including accuracy, precision, recall, and F1 score. The Decision Tree model achieved the highest accuracy (85%), making it the best performer among the three models. However, it is important to note that the choice of the best model depends on the specific requirements and constraints of the problem at hand. For instance, if interpretability is crucial, a Decision Tree might be preferred over more complex models, even if it has slightly lower accuracy.
4) *Implementation:* We implemented the fraud detection system using Python and various libraries, including scikit-learn for machine learning, Beautiful Soup for web scraping, and pandas for data manipulation. The system is designed to be scalable and efficient, capable of handling large volumes of data in real-time. We also developed a user-friendly interface that allows users to input an app's URL and receive a fraud detection report. This report includes the model's prediction (fraudulent or legitimate) and additional information on the app's features and behavior. The implementation ensures that the system is accessible and easy to use for non-technical users, providing them with a valuable tool for app safety assessment.

VI. RESULTS AND DISCUSSION

The Decision Tree model emerged as the best performer, achieving an accuracy of 85%, a precision of 0.87, a recall of 0.85, and an F1 score of 0.815. These results demonstrate the model's robustness and effectiveness in detecting fraudulent apps. However, there is still room for improvement, particularly in handling data imbalance and incorporating more dynamic features. Future work could explore advanced techniques such as ensemble learning, deep learning, and real-time monitoring to further enhance the system's performance.

A. Modules

1) *Data Collection and Preprocessing:* This module collects pertinent data from the Google Play Store, such as app ratings, reviews, in-app purchase information, and ad presence. The acquired data goes through preparation to handle missing values, delete duplicates, and convert textual material into a format appropriate for analysis.
2) *Feature Engineering and Selection:* In this module, the system focuses on expanding the feature set by incorporating new dynamic elements that reflect the changing nature of mobile applications. Feature selection approaches can be used to discover the most relevant attributes for training the fraud detection model and improving its performance.
3) *Machine Learning Models:* This subject covers the implementation and training of machine learning models, such as Decision Tree, Logistic Regression, and Naïve Bayes. Hyperparameter tweaking and model evaluation approaches are used to assure peak performance. The investigation found the Decision Tree model as effective, which would be a major component.
4) *Real-time Monitoring and Analysis:* To address the dynamic nature of mobile apps, this module employs real-time monitoring of app behaviour, updates, and user interactions. Continuous analysis guarantees that the model remains flexible to developing trends and is capable of quickly detecting fraudulent activities as they evolve over time.

5) *User Interface and Reporting:* The system has a user-friendly interface that allows users to interact with and interpret the model's predictions. This module enables users to enter app details or questions and return clear and understandable fraud risk assessments. Visualizations and short summaries help to present the results, increasing user knowledge and trust in the system.

B. Experimental Results

The basic goal of the suggested approach is to investigate fraud detection in the Google Play store for apps and use four parameter strategies to discriminate between different fake apps, commonly referred to as spam apps. The recommended technique for detecting fraudulent or false applications includes performing experimental research using a number of methodologies. Our method will detect fraud by examining four sorts of data: ad-based ratings, in-app payments, and evidence-based reviews. Furthermore, the development-based integrated strategy employs all four criteria for detecting fraud. Several artificial intelligence (AI) models were used, each with varying levels of accuracy. Our investigation revealed that the one we recommended approach outperforms existing algorithms by 85%. While autonomous thinking survives, the Decision Tree element outperforms other models, including the financial crisis and Naive Bayes. It is an easy way to divide challenges. It is an accurate real-time guess, which comes with a drawback. Decision trees are effective for dealing with nonlinear data sets. It influences judgments in a variety of sectors, notably technology, social organizing, business, and even the law.

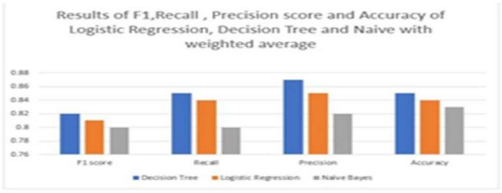


Fig 2. Results Graph

The Results of F1, Recall, Precision and Accuracy Scores of Logistic Regression, Decision Tree and Naive with weighted average.

TABLE 2. COMPARATIVE ANALYSIS ON DIFFERENT MACHINE LEARNING ALGORITHMS FRAUD APP DETECTION

	F1	Recall	Precision	Accuracy
Decision Tree	0.815	0.85	0.87	0.85
Logistic Regression	0.81	0.84	0.85	0.84
Naïve Bayes	0.8	0.83	0.82	0.83

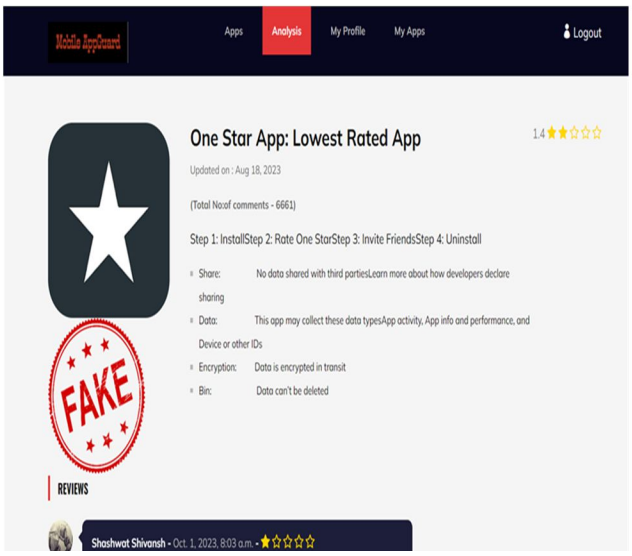


Fig 3. Negative Result

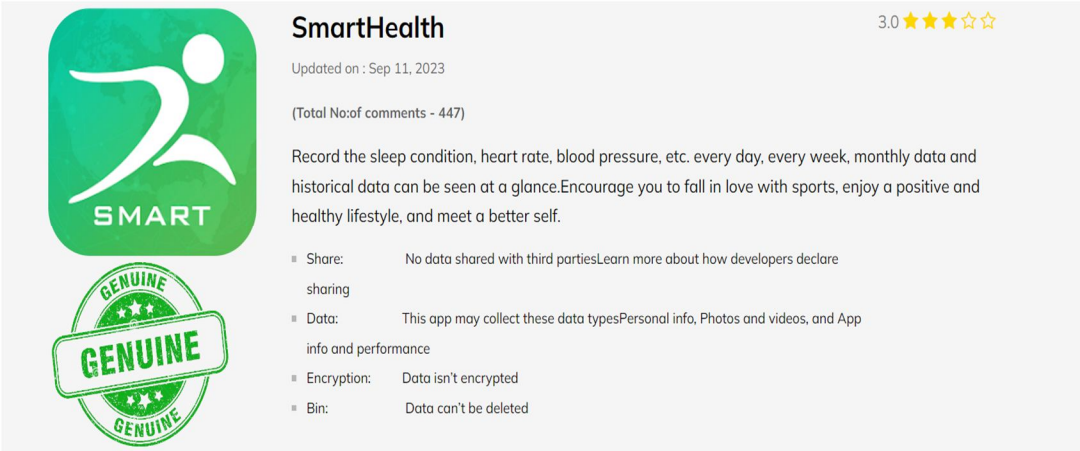


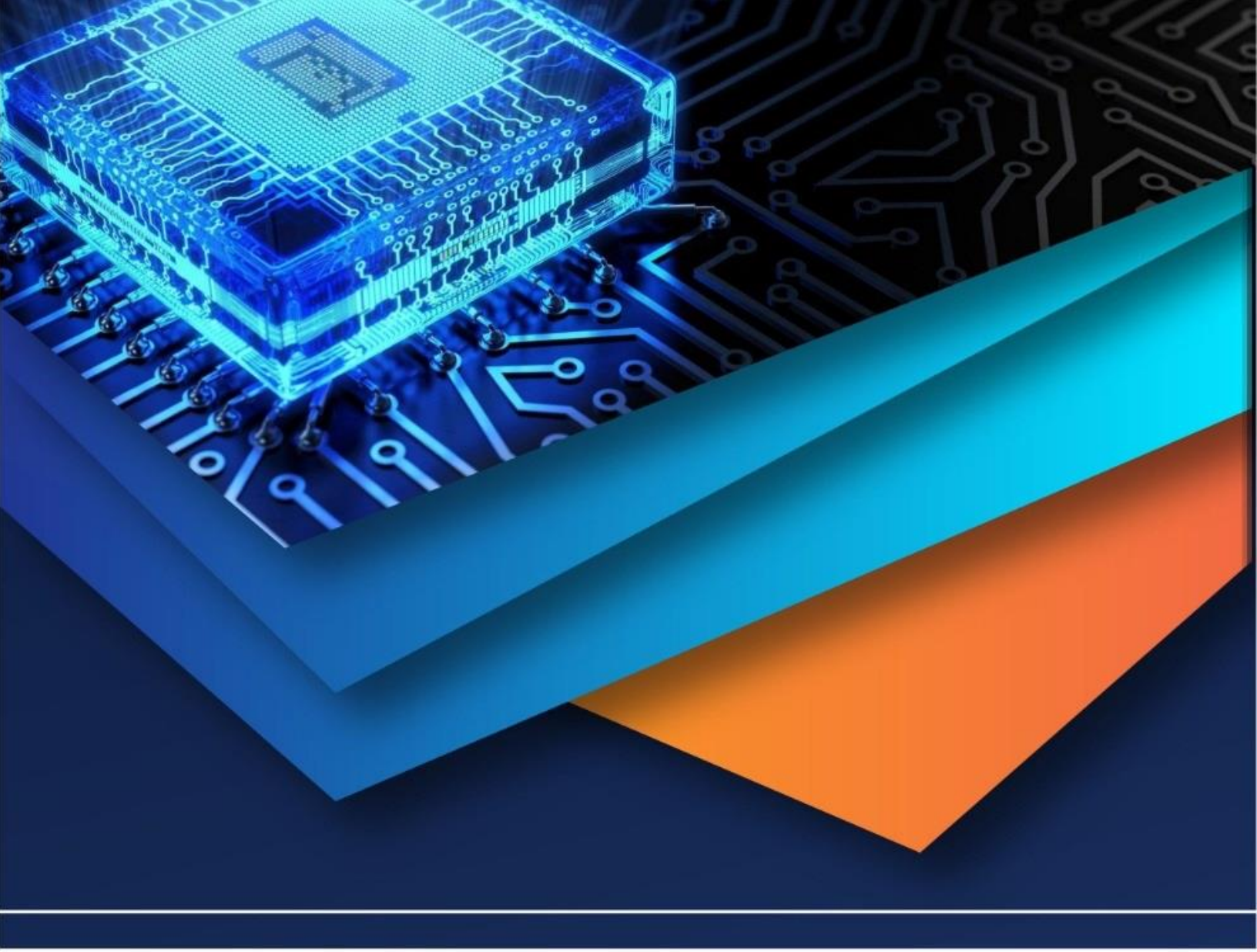
Fig 4. Positive Result

VII. CONCLUSIONS

This study presents a comprehensive approach to detecting fraudulent apps on the Google Play Store using machine learning. We demonstrated the effectiveness of our methodology through extensive experiments and analysis. The Decision Tree model, with its high accuracy and interpretability, proves to be a valuable tool for identifying fraudulent apps. Our proposed system addresses the limitations of existing methods and offers several advantages, including enhanced feature set, better handling of data imbalance, increased interpretability, and advanced analytical techniques. Future work will focus on further improving the system's accuracy and reliability by incorporating more dynamic features and exploring advanced machine learning techniques.

REFERENCES

[1] Esther Nowroji, Vanitha, "Detection Of Fraud Ranking For Mobile App Using IP Address Recognition Technique", vol. 4.
[2] JavvajiVenkataramaiah, BommavarapuSushen, Mano. R, Dr.GladispushpaRathi, "An enhanced mining leading session algorithm for fraud app detection in mobile applications"
[3] S.R.Srividhya, S.Sangeetha – "A Methodology to Detect Fraud Apps Using Sentiment Analysis"
[4] Keerthana. B, Sivashankari.K and ShaisthaTabasum.S, "Detecting Malwaresand Search Rank Fraud in Google Search Using Rabin Karp Algorithm", IJARSE,7(02), 2018, pp.504-527.
[5] Shashank Bajaj, Nikhil Nigam, PriyaVandana, Srishti Singh, "Detection of fraud apps using sentiment analysis", International Journal of Innovative Science and Research Technology.
[6] HarpreetKaur, VeenuMangat and Nidhi, — "A Survey of Sentiment Analysis techniques"
[7] International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2017, pp.921
[8] Jing Wan, Mufan Liu, Junkai Yi and Xuechao Zhang, "Detecting Spam Webpages through Topic andSemantics Analysis", IEEE Global Summit onComputer and Information Technology (GSCIT), 2015, pp. 83-92.
[9] Navdeep Singh, Prashant Kr. Pandey and Mr.Srinivasan, — "Improved Discovery of Rating Fake for Cellular Apps", IEEE International Conference on Science Technology Engineering and Management (ICONSTEM), 2016, pp. 135-140.
[10] Weiman Wang, Restricted Boltzmann Machine. GitHub. Aug 2017. [Online] Available: https://github.com/aaxwaz/Fraud-detection-usingdeep-learning/blob/master/rbm/rbm.py.
[11] DubeyVeena, G. D. (2016). Sentiment Analysis Based on Opinion Classification Techniques: A Survey .International Journal of Advanced Research in Computer Science and Software Engineering, 5358.
[12] Ranking fraud Mining personal context-aware preferences for mobile users. H. Zhu, E. Chen, K. Yu, H. Cao, H. Xiong, and J. Tian. In Data Mining (ICDM), 2012 IEEE 12th International Conference on, pages1212–1217, 2012.
[13] NandimathJyoti, K. B. (2017). Efficiently Detecting and Analyzing SpamReviews Using Live Data Feed. International Research Journal of Engineering and Technology (IRJET) , 1421-1424.
[14] Detecting product review spammers using rating behaviors. E.-P. Lim, V.-A.Nguyen, N. Jindal, B. Liu, and H. W. LauwIn Proceedings of the 19th ACM international conference on Information and knowledge management, CIKM "10 pages 939–948, 2013.
[15] Detection for mobile apps H. Zhu, H. Xiong, Y. Ge , and E. Chen. A holistic view. In Proceedings of the 22nd ACM international conference on Information and knowledge management, CIKM "13, 2013.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY