Journal of Applied Information Science
11 (2) 2023, 29-34
http://www.publishingindia.com/jais

# Fraud App Detection using Sentimental Analysis

Ganesh Rahangdale[1], Joel Craig[2], Poossan Gopikrishnan[3*], Shivam Butle[4], Shoyeb Sheikh[5] and Priya Narnavere[6]

[1]U.G. Student, Department of Information Technology Engineering, J D College of Engineering and Management, Fetri Nagpur, Maharashtra, India. Email: ganeshrahangdale41@gmail.com
[2]U.G. Student, Department of Information Technology Engineering, J D College of Engineering and Management, Fetri Nagpur, Maharashtra, India. Email: joelcraig01@gmail.com
[3]U.G. Student, Department of Information Technology Engineering, J D College of Engineering and Management, Fetri Nagpur, Maharashtra, India. Email: poossangopikrishnan@gmail.com
[4]U.G. Student, Department of Information Technology Engineering, J D College of Engineering and Management, Fetri Nagpur, Maharashtra, India. Email: shivambutle7@gmail.com
[5]U.G. Student, Department of Information Technology Engineering, J D College of Engineering and Management, Fetri Nagpur, Maharashtra, India. Email: sheikhshoyeb28@gmail.com
[6]Assistant Professor, Department of Information Technology Engineering, J D College of Engineering and Management, Fetri Nagpur, Maharashtra, India, Email: pynarnavere@jdcoem.ac.in
*Corresponding Author

**Abstract: This project aims to develop a fraud app detection system using sentiment analysis. The system leverages Java 1.8 Spring Boot, React, HTML, CSS, JavaScript, and Bootstrap to create a robust web application. The methodology involves collecting user reviews and comments, preprocessing the data, and applying sentiment analysis models to determine sentiment scores. The system then uses predefined fraud criteria to flag potentially fraudulent reviews. Integrating the system into a Java Spring Boot backend and visualizing results using React provides real-time monitoring and investigation. Continuous improvement, user feedback handling, and effective model selection ensure enhanced accuracy and adaptability to evolving fraudulent patterns. This project presents an integrated fraud detection system for user reviews, utilizing sentiment analysis within a Java 1.8 Spring Boot backend and React frontend. It encompasses data collection, preprocessing, sentiment analysis, and predefined fraud criteria to flag suspicious reviews. Real-time monitoring and investigation capabilities are offered through an intuitive web interface. The project's commitment to continuous improvement, user feedback integration, and effective model selection ensures adaptability to evolving fraudulent patterns, enhancing accuracy and preserving the credibility of online platforms in an era where user-generated content profoundly influences consumer decisions.**

**Keywords: Fraud apps detection, Sentiment analysis, Technological development.**

## I. Introduction

Sentiment is an emotion or attitude that is brought on by the client's emotions. As consumer opinions are gathered and mined to determine an app's rating, sentiment analysis is also known as opinion mining. Information is gathered, analysed, and then classified as either positive or negative depending on how it is felt. People always research the app's reputation among users before making a purchase. Sentiment analysis is a procedure that gathers and analyses a sentence's opinion or sentiment using natural language processing (NLP) [1]. It is well-liked since many people choose to heed user recommendations. It is beyond the control of manual procedures to analyse enormous amounts of reviews and to aggregate them into an effective choice because the number of opinions in the form of reviews, blogs, etc. are expanding continuously. Sentiment analysis converts these actions into automated procedures with minimal human assistance. Because different phrase forms express thoughts and opinions in different ways, it is not always possible to have a single strategy that works for all situations. Sentence terms that are also referred to as opinion words, such as wonderful, beautiful, bad, etc., cannot tell an opinion sentence from a non-opinion sentence. Even if a conditional statement lacks an opinion, it may contain numerous sentimental phrases or sentences. It can be challenging to discern the orientation of attitudes on themes or qualities in conditional phrases because they have certain distinctive traits of their own. Positive, negative, or neutral sentiment orientations are the different types of opinions. Sentences that explain implications

or potential outcomes are known as conditional sentences. Many different types of conditional connectives can be used to form these sentences. A conditional sentence contains two clauses: the condition clause and the consequent clause, that are dependent on each other. Their relationship has significant implications on whether the sentence describes an opinion [2].

## II. Literature Survey

This paper hopes to see customers making spam diagrams or audit spammers. They see a couple trademark practices of survey spammers and model these practices with a particular ultimate objective to perceive the spammers. Creators endeavor to display the running with phones. Regardless, spammers may target particular things or things that accumulate keeping in mind the end goal to develop their effect. Second, they tend to leave trade specialists in their evaluations of things. In paper, creators have examined the issue of finding half and half shilling assaults on rating data. The philosophy relies upon can be used for dependable thing proposals and semi-managed learning. This paper shows a Hybrid Shilling Attack Detector or Hy SAD for short, to deal with this issue [3].

Nowadays, the majority of us use mobile devices with iOS or Android operating systems, and we frequently use the functionality of the play store or app store. A wide variety of software is available on both markets, however unfortunately some of those programmes are fake. Both data theft and device damage are possible with these apps. Thus, they must be labelled in order for store patrons to recognise such programmes. To manage the data, feedback, and application evaluation, we suggest a web application. As a result, it will be simpler to determine whether or not an application is fraudulent. The online application allows for the simultaneous processing of many applications. Most of us use mobile devices these days that run Android or iOS, and we routinely cannot always find reliable or honest product reviews online. As a result, the admin will assess the reviews and comments, making it easy for the admin to decide whether the application is honest or dishonest.

## III. Methodology

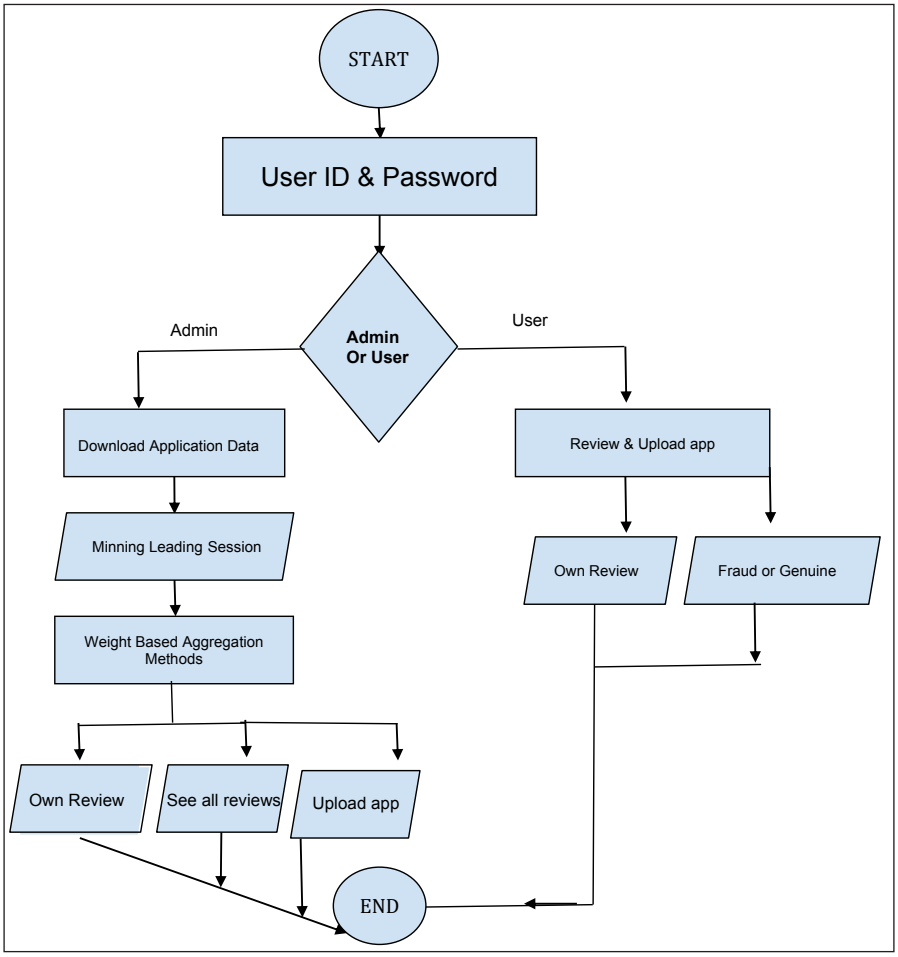


Fig. 1: Flowchart

*Data Collection:* Data collection serves as the foundational step in understanding user sentiments and experiences related to the app. By systematically gathering user reviews, comments, and feedback, you compile a rich and diverse dataset that reflects a wide spectrum of opinions and viewpoints [4]. App stores provide insights directly from users, social media platforms offer real-time reactions, and your own database could contain historical data for analysis. This multi-source approach ensures a holistic representation of user sentiment.

The collected data not only captures positive and negative feedback but also nuances and trends in user opinions. User reviews often contain valuable insights into specific app features, performance, customer support, and overall satisfaction. The amalgamation of these diverse perspectives creates a comprehensive dataset that forms the basis for subsequent analysis.

Incorporating data from various sources fosters a more accurate understanding of user sentiment, enabling you to develop a well-informed fraud detection system. By encompassing both structured app store reviews and more informal social media commentary, you gain a more complete picture of user perceptions. This, in turn, enhances the credibility and effectiveness of the subsequent steps in your project, such as sentiment analysis and fraud criteria definition.

*Data Preprocessing:* In the data preprocessing phase, the text data undergoes a series of transformations to optimize it for analysis. First, special characters, URLs, and unnecessary symbols are removed, ensuring that the text is devoid of extraneous elements. Subsequently, the process of tokenization divides the text into individual words or tokens, facilitating granular analysis. To maintain consistency, all text is converted to lowercase, mitigating the influence of case variations. Further refinement involves the removal of common stopwords like "and," "the," and "is," which have limited significance in sentiment analysis. Lastly, stemming or lemmatization reduces words to their base or root form, enhancing dimensionality reduction and improving data coherence [5]. These combined preprocessing steps establish a structured and clean textual foundation that is well-suited for accurate sentiment analysis and subsequent fraud detection.

*Sentiment Analysis:* Perform an appropriate sentiment analysis model for the task we are using NLP to achieve the goal. In this project, sentiment analysis serves as a pivotal component for detecting fraudulent app reviews. The sentiment analysis process involves evaluating the sentiment expressed in user reviews and comments to determine whether they are positive, negative, or neutral. This analysis aids in gauging user satisfaction and identifying potentially fraudulent content.

The collected user reviews and comments undergo preprocessing, including text cleaning, tokenization, lowercasing, and possibly stopword removal. After this preprocessing, sentiment analysis models are applied to generate sentiment scores for each review. These scores quantify the sentiment of the text, allowing for a more objective assessment of user opinions.

The sentiment scores obtained from the sentiment analysis models are then used to inform the fraud detection algorithm. This algorithm incorporates predefined fraud criteria that flag reviews with sentiment scores falling within specific ranges or exhibiting certain patterns. Reviews with extremely negative sentiment or containing keywords associated with fraudulent behavior might trigger flags.

By leveraging sentiment analysis, the project is able to automate the process of identifying potentially fraudulent reviews. Reviews that express unusually negative sentiment or deviate from the expected sentiment distribution can be flagged for further investigation. This systematic approach enhances the accuracy and efficiency of the fraud detection system.

Incorporating sentiment analysis into the project enables the system to quantitatively assess user sentiments, providing valuable insights for detecting suspicious content. This analysis, coupled with the system's integration into a Java Spring Boot backend and visualization using React, creates a comprehensive solution for real-time monitoring and fraud detection in app reviews.

*Evaluation*

*Fraud Criteria:* Established criteria to identify fraudulent or suspicious reviews based on the sentiment scores obtained from the sentiment analysis model. For example, reviews with very negative sentiment or specific keywords related to fraud may be flagged. Evaluation in this project involves the establishment of fraud criteria used to identify potentially fraudulent or suspicious reviews. These criteria are developed based on the sentiment scores derived from the sentiment analysis model. For instance, reviews displaying extremely negative sentiment or containing particular keywords associated with fraudulent activity are earmarked for further scrutiny. This systematic approach ensures that flagged reviews align with specific patterns indicative of potential fraud. The fraud criteria provide a quantifiable framework to assess the legitimacy of user reviews and contribute to the overall accuracy and effectiveness of the fraud detection system [6].

*Fraud Detection Algorithm*

*Sentiment Scoring:* Apply the sentiment analysis model on the preprocessed data to get sentiment scores for each review or comment.

*Flagging:* Implement an algorithm to flag reviews that meet the fraud criteria.

*User Feedback Handling:* Develop a mechanism for users to provide feedback on flagged reviews. This feedback will be used to improve the accuracy of the fraud detection system over

time.

*Integration:* Integrate the fraud detection system into your Java Spring Boot backend to automatically process user reviews and comments.

*Visualization and Reporting:* Create a user interface using React, HTML, CSS, and JavaScript to display the analyzed sentiment and flagged reviews for further investigation.

*Testing and Deployment:* Thoroughly test the system to ensure it functions correctly. Deploy the fraud app detection system to your production environment. This project aims to build a fraud detection system for app reviews using sentiment analysis. It involves applying sentiment analysis models to user reviews, generating sentiment scores that quantify the tone of each review. These scores are then utilized in an algorithm that automatically flags reviews based on predefined fraud criteria. To enhance accuracy, the system incorporates user feedback on flagged reviews, allowing continuous refinement. The system is integrated into a Java Spring Boot backend for streamlined processing, while a React-based interface visualizes sentiment scores and flagged reviews, facilitating investigation. Rigorous testing ensures reliability before deployment to the production environment, providing users with a reliable tool to identify potentially fraudulent reviews and improve overall review authenticity [7].

## IV. Experimental Results

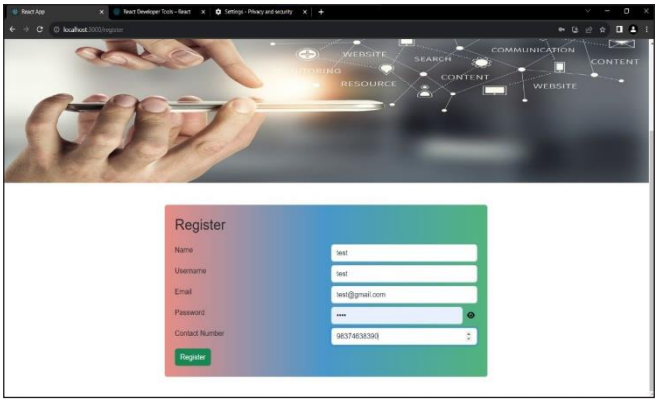


Fig. 2: Registration Page

This project focuses on the development of a fraud app detection system that utilizes sentiment analysis techniques. The system harnesses a technology stack including Java 1.8 Spring Boot, React, HTML, CSS, JavaScript, and Bootstrap to establish a robust web application [8]. The project methodology encompasses several stages: firstly, gathering user reviews and comments from various sources. Subsequently, the collected data undergoes preprocessing steps, including text cleaning and tokenization. Sentiment analysis models are then employed to generate sentiment scores that reflect user sentiments.

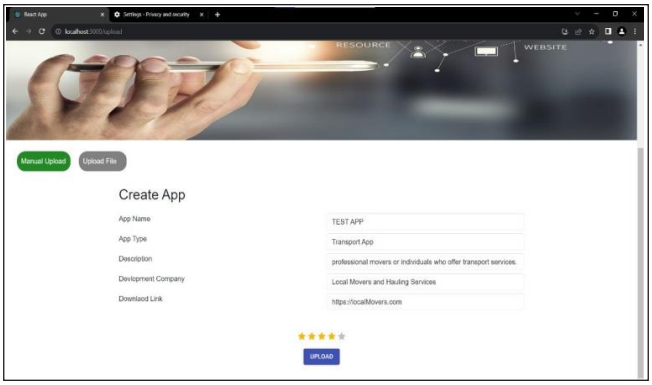


Fig. 3: Manual Upload

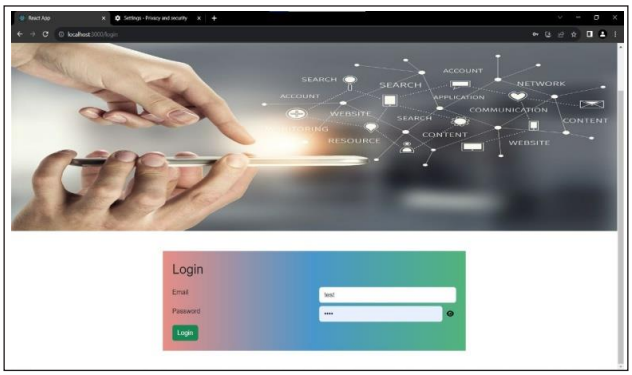


Fig. 4: Login Page

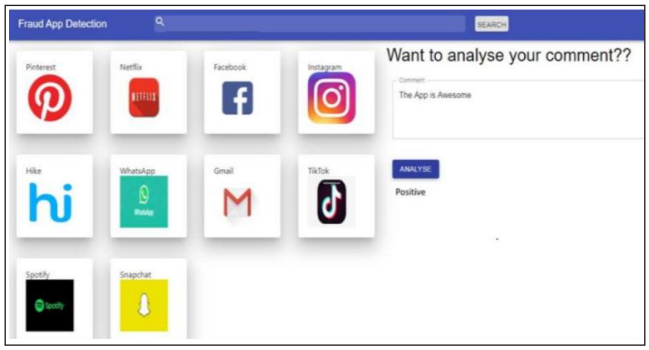


Fig. 5: Detection of Fraud App

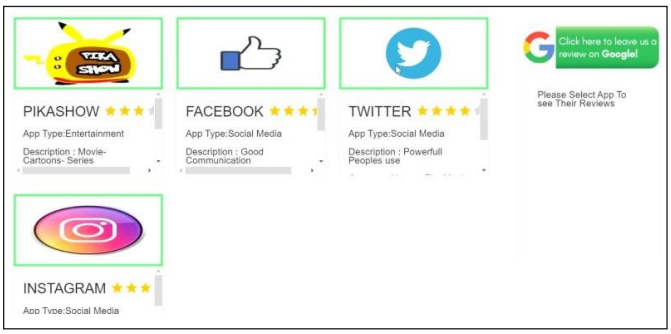


Fig. 6: Rating and Reviews

Implementing a robust fraud detection system for app reviews entails several key steps. Data collection involves gathering user feedback from diverse sources, such as app stores and social media platforms. Prior to analysis, data preprocessing tasks, including text cleaning, tokenization, lowercasing, stopword removal, and stemming or lemmatization, ensure data consistency and quality. Applying sentiment analysis models, which could include machine learning or deep learning techniques, quantifies user sentiment in each review. This stage provides insight into user satisfaction or dissatisfaction.

The evaluation phase involves defining criteria for identifying fraudulent reviews, often characterized by very negative sentiment or keywords associated with fraud. These criteria serve as guidelines for the fraud detection algorithm. The algorithm utilizes sentiment scores obtained from the sentiment analysis model to flag reviews meeting the established fraud criteria. This algorithmic approach aids in automating the identification of suspicious reviews for further review.

Integration of additional methodologies enhances the system's effectiveness. Aspect-based sentiment analysis dissects feedback by app components, offering more precise insights. Ensemble models combine sentiment analysis methods for improved accuracy. N-gram analysis captures nuanced sentiments expressed through phrases. User behavior analysis examines review history and posting patterns to identify potential fraudulent users. Incorporating time series analysis reveals trends and sudden shifts in sentiment that may suggest fraudulent activity [9].

The results of an effective fraud detection system encompass enhanced accuracy in identifying suspicious content, reduced fraudulent activity, improved user trust in reviews, and informed app development decisions. As false positives and negatives decrease over time, user experiences improve, contributing to a more reliable app rating ecosystem. The system's continuous learning and refinement adapt to evolving fraudulent tactics. Ultimately, an accurate fraud detection system safeguards business reputation, saves costs, and promotes transparent and trustworthy user experiences. Regular assessment and adjustment of methodologies ensure the system's adaptability to dynamic review landscapes and evolving user behaviors.

The core functionality of the system revolves around the application of predefined fraud criteria. These criteria are used to flag reviews that exhibit characteristics potentially indicative of fraudulent behavior. The integration of this system within a Java Spring Boot backend, along with visualization through React, offers a comprehensive web-based solution. This integration facilitates real-time monitoring and the ability to investigate flagged reviews promptly.

A commitment to continuous improvement is integral to the project's success. The system is designed to incorporate user feedback, enabling refinements based on real-world usage and insights. The selection of appropriate sentiment analysis models plays a crucial role in achieving accurate results [10]. Moreover, the system's adaptability is emphasized, ensuring its capacity to effectively identify and respond to evolving fraudulent patterns.

In summary, this endeavor aims to create a powerful fraud app detection system by employing sentiment analysis. Through the utilization of cutting-edge technologies like Java Spring Boot and React, the project strives to provide a seamless user experience for monitoring and mitigating potentially fraudulent reviews. By following a structured methodology, incorporating user feedback, and ensuring model accuracy, the resulting system endeavors to uphold the integrity of app reviews and user trust.

## V. Conclusion

Through the use of online social networking research, this study successfully developed an improved feeling characterisation technique for peculiarity location. Utilising tweet data as a contextual investigation, the feasibility of the suggested technique is demonstrated. Using the suggested technique, the strangeness estimate designs were efficiently identified and translated. The Contextual analysis demonstrated the usefulness and dominance of the method. When it comes to handling conclusion design characterizations, given the acceptance of our method in light of an unnatural state of anger that has become stronger with similar grouping assignments carried out by annotators. This investigation gives fresh ideas for describing a robust opinion examination method using information from web-based networking media to distinguish instances or examples of inconsistency. The tactic will apply in situations like design changes after a while. This should be really profitable. for businesses to secure their administrative hub, for government innovators and political aspirants to understand the rationale behind their ongoing research arises, and for other intimate associations to become more refined their clients' brand assurances and incentives.

Implementing fraud app detection using sentiment analysis involves collecting user feedback, preprocessing the data, applying sentiment analysis models, and developing a fraud detection algorithm based on predefined criteria. Integrating this system into a Java Spring Boot backend and visualizing the results using React, HTML, CSS, and JavaScript allows for real-time monitoring and investigation of potentially fraudulent activities. Continuous improvement, user feedback handling, and effective model selection are critical for enhancing the system's accuracy and adapting to evolving fraudulent patterns.

## References

[1] R. Safrin, K. R. Sharmila, T. S. ShriSubangi, and E. A. Vimal, "Sentiment analysis on online product review," *Int. Res. J. Eng. Technol,* vol. 4, no. 4, 2017.

[2] P. H. Shahana, and B. Omman, "Evaluation of features on sentimental analysis," *Procedia Computer Science*, vol. 46, pp. 1585-1592, 2015.

[3] Mohd. T. Khan, M. Durrani, A. Ali, I. Inayat, S. Khalid, and K. H. Khan, "'Sentiment analysis and complex natural language," A Springer Open Journal, 2016.

[4] R. Narayanan, B. Liu, and A. Choudhary, "Sentiment analysis of conditional sentences," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2009, vol. 1, pp. 180-189.

[5] L. Gang, and F. Liu, "A clustering-based approach on sentiment analysis," in *2010 IEEE International Conference on Intelligent Systems and Knowledge Engineering*, IEEE, 2010, pp. 331- 337.

[6] X. Jianlin, Y. Yu, Z. Chen, B. Cao, W. Song, Y. Guo, and J. Cao, "MobSafe: Cloud computing based forensic analysis for massive mobile applications using data mining," *Tsinghua Science and Technology*, vol. 18, no. 4, pp. 418-427, 2013.

[7] A. Tichkule, N. Nikhar, D. Kapgate, and O. Dudhbure, "Revelation of fraud Apps using sentiment analysis App reviews," *International Journal of Innovations in Engineering and Science*, vol. 4, no. 5, 2019.

[8] V. Rohini, and M. Thomas, "Comparison of Lexicon based and Naïve Bayes classifier in sentiment analysis," *International Journal for Scientific Research & Development*, vol. 3, no. 4, 2015.

[9] E. Guzman, and W. Maalej, "How do users like this feature? A fine grained sentiment analysis of app reviews," in *2014 IEEE 22nd International Requirements Engineering Conference (RE),* IEEE, 2014, pp. 153-162.

[10] X. Zhou, X. Tao, J. Yong, and Z. Yang, "Sentiment analysis on tweets for social events," in *Proceedings of the 2013 IEEE 17th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, IEEE, 2013, pp. 557-562.