**DETECTING FRAUD APPS USING SENTIMENT ANALYSIS**

*Submitted in the partial fulfilment for the award of the degree of*

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE WITH SPECIALIZATION IN**

**BIG DATA ANALYTICS**

**Submitted by:**

21BCS3854 SHUBHANSHU PANDEY

21BCS11228 RIDHIMA HANDA

21BCS3807 SHAMEEM AHMAD

21BCS3703 ADIT KULSHRESHTHA

**Under the Supervision of**

**AMANPREET KAUR**



**CHANDIGARH UNIVERSITY, GHARUAN, MOHALI - 140413,**

**PUNJAB**

**September, 2022**

# Abstract

With the proliferation of mobile applications in everyday life, it is critical to keep track of which ones are safe and which are not. How it

secure and reliable Each application is solely based on the reviews that have been mentioned for that application. As a result, it is necessary to keep track of and develop a system to ensure that the apps available are genuine or not. The goal is to create a system that detects fraudulent apps before the user downloads them using sentimental analysis and data mining. Sentimental analysis is used to help determine the emotional tones behind words expressed online.

This method is useful for monitoring social media and getting a quick sense of public opinion on specific issues. On the internet, the user may not always find accurate or true product reviews. We can search for sentimental comments from users across multiple applications. The evaluations could be false or real. We can determine whether the app is genuine or not by analysing the rating and reviews, which include both user and admin comments. Using sentimental analysis and data mining, the machine can learn and analyse the sentiments, emotions about reviews and other texts.

**Keywords:** Sentimental Analysis, Review based evidence, positive negative ratings, Rate evidence, Users review, Leading session.

# Table of Contents

|  |  |
| --- | --- |
| Title Page | i |
| Abstract | ii |
| 1.    Introduction |  |
| 1.1           Problem Definition |  |
| 1.2 Project Overview |  |
| 1.3 Hardware Specification |  |
| 1.4 Software Specification |  |
| 2.    Literature Survey |  |
| 2.1 Existing System |  |
| 2.2 Proposed System |  |
| 2.3 Literature Review Summary |  |
| 3.    Problem Formulation |  |
| 4.    Research Objective |  |
| 5.    Methodologies |  |
| 6.    Experimental Setup |  |
| 7.    Conclusion |  |
| 8.    Tentative Chapter Plan for the proposed work |  |
| 9.    Reference |  |

# 1. INTRODUCTION

## 1.1 Problem Definition

There are many apps on the internet that look and function similarly to the original app, but they are fake. So, the users who use fake apps have faced issues after installing the app. That app harms their devices and extracts some personal information about the user. So, the reviews of the original app are decreasing because so many apps are now banned because of data leaks. So, to overcome this problem, we are using sentiment analysis to detect the fraud apps that are present in the internet world. The use of mobile phones is increasing as technology advances. The creation of various mobile apps on various systems, such as the famous Android and iOS, has increased dramatically. Because of its rapid increase in daily utilisation, sales, and developments, it has become a major task in the world of business intelligence. This increases industry rivalry. Companies and application developers are competing fiercely with one another in order to demonstrate the quality of their products and devote significant resources to drawing consumers in order to maintain their future progress.

## 1.2 Problem Overview

The most essential part that customers perform is their ranking, ratings, and reviews on the specific application that they obtain. This could be a method for writers to identify their weaknesses and incorporate them into the creation of a new one while keeping the needs of the people in mind. Positioning misrepresentation for versatile application showcase refers to phoney or deceptive exercises with the goal of boosting the applications' prevalence in the list. It is becoming more common for application creators to use dubious means, for example, extending their application agreements, to submit positioning misrepresentation. We show a comprehensive view of positioning misrepresentation and suggest a methodology for detecting positioning extortion in a variety of applications. We look at three types of confirmations: Confirmations based on ranking; Confirmations based on ranking Confirmations based on a review. Some engineers may use promoting methods, such as an ad crusade, to progress their application. Occasionally, for the benefit of the developers, teams of employees are hired who perpetrate fraud jointly and provide false remarks and evaluations on an application. This is referred to as mob turfing. As a result, it is always critical to ensure that users are given with proper and genuine remarks prior to downloading an app in order to prevent certain mishaps. This necessitates the use of an automatic solution to surmount and methodically analyse the various remarks and scores given for each application.

## 1.3 Hardware Specification

* The application must provide accurate results.
* Perform the desired function: sorting fraud applications.
* Provide better flexibility and is user friendly.
* User should have to access system to the previous analysed reports.
* User of the system should have operating systems like Windows 7, Windows 8 and Windows10 (32/64 bit).
* The system is implemented using Android Studio (JAVA, XML).
* We require minimum 3 GB RAM, 8 GB RAM recommended, plus 1 GB for the Android Emulator.
* The system should have 1280 x 800 minimum screen resolution.

## 1.4 Software Specification

Windows 10 is the latest version of the Microsoft Windows operating system for personal computers and other devices. It was released on July 29, 2015, as a successor to Windows 8.1. Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis. Python is a general-purpose language, meaning it can be used to create a variety of different programs and isn’t specialized for any specific problems. This versatility, along with its beginner-friendliness, has made it one of the most-used programming languages today. A survey conducted by industry analyst firm RedMon found that it was the second-most popular programming language among developers in 2021. Python is commonly used for developing websites and software, task automation, data analysis, and data visualization. Since it’s relatively easy to learn, Python has been adopted by many non-programmers such as accountants and scientists, for a variety of everyday tasks, like organizing finances. Flask is a web application framework written in Python. Armin Ronacher, who leads an international group of Python enthusiasts named Pocco, develops it. Flask is based on Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects. This tutorial has been prepared for anyone who has a basic knowledge of Python and has an urge to develop websites. After completing this tutorial, you will find yourself at a moderate level of expertise in developing websites using Flask.

**2. LITERATURE SURVEY**

## 2.1 Existing System

The authors of this work have highlighted several difficulties with NLP at the document, sentence, feature, and lexicon levels. In order to address the issues of natural language processing, they have also compared several methods and approaches, including lexicon-based, statistical, k-nearest neighbour, centroid, and naive bayes. Reviews of applications are extracted and transformed into tokens. Tokenization is the process of turning a stream of text into tokens, which are collections of words, phrases, and symbols.

Identify the user's feelings. A good review increases the score by 1, and a bad review decreases the score by 1. With this, it will calculate each review's score and evaluate whether the application is legitimate or fraudulent.

The purpose of this document is to encourage users to audit spammers or create spam diagrams. They observe a few distinctive behaviours used by survey spammers and imitate these behaviours to understand the spammers. Developers try to show running in a precise manner. Nonetheless, spammers may target specific items or collections of items with the intention of amplifying their impact. Second, in their assessments of things, they frequently depart from advice from industry experts. In paper, authors looked at the problem of discovering mixed-shilling attacks on rating data. For trustworthy item proposals and semi-managed learning, philosophy can be used.

## 2.2 Proposed System

This study presents a Hybrid Shilling Attack Detector, or Hy SAD, to address this problem. Hy SAD adapts Relief with a few effective acknowledgment metrics and Semi Oversight Naive Bays (SNB) to precisely distinguish Random-Filler display attackers and Average-Filler display attackers from regular consumers. The sentiment analysis and data mining used to extract the dataset collected are the key focuses of this research. The genuine value of the programmes given by Play and App can be ascertained using this method. Such a system's projected data set will be enormous, and data mining in conjunction with visual data will aid in system execution.

Information is obtained from a variety of internet-based, mobile, and exchanges that include surveys, comments, and other data related to the particular business. Also, in this case, sensation analysis is used to separate the data for upcoming updates based on the measurements obtained by estimation analysis. A crucial but challenging subject is the analysis of large informational resources.

## 2.3 Literature Review Summary

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Year and**  **Citation** | **Article/ Author** | **Tools/ Software** | **Technique** | **Source** | **Evaluation Parameter** |
| 2020 | Dr. Shailendra  Aswale | MATLAB Toolbox | Sentiment  Analysis | Dataset | Five Segments |
| 2018 | Gauri Rao,  Shashank Bajaj | Rapid Miner | Sentiment  Analysis | Dataset | Five Segments |
| 2019 | Mandava Rama Rao, Nandhini Kannan | TensorFlow | Sentiment  Analysis | Dataset | Three Segments |

# 3. PROBLEM FORMULATION

The security and privacy of users are maintained in large part by spotting fraudulent programmes in app stores. By the examination of user ratings and feedback, sentiment analysis may be used to spot fake programmes. The following is a description of the issue formulation for sentiment analysis-based fraud detection apps. The objective is to create a machine learning model that can precisely divide app reviews into two groups, authentic and fraudulent, using a dataset of app evaluations. A labelled dataset of reviews, where each review is classified as either valid or fraudulent, should be used to train the model. The content of the reviews should be analysed by the sentiment analysis model to identify characteristics that may be utilised to distinguish between honest and dishonest evaluations. These characteristics might include the tone of the review, the language used, its length, and the repetition of particular words or phrases. The label of fresh reviews that have not been seen previously should therefore be predicted by the machine learning model using these criteria. To gauge the model's performance and accuracy, a different test set of evaluations should be used. The ultimate objective of this issue is to create a model that can quickly and effectively identify fraudulent programmes in app stores, enabling people to download and utilise apps with more knowledge. The label of fresh reviews that have not been seen previously should therefore be predicted by the machine learning model using these criteria. To gauge the model's performance and accuracy, a different test set of evaluations should be used. The ultimate objective of this issue is to create a model that can quickly and effectively identify fraudulent programmes in app stores, enabling people to download and utilise apps with more knowledge.

**4. OBJECTIVE**

The goal of sentiment analysis fraud detection research is to create an automated system that can correctly identify counterfeit programmes in app stores. The following sub-objectives can be further divided into this main goal. Provide a thorough collection of app reviews, including both real and fraudulent ratings. This dataset should include many reviews for many apps. The dataset needs to be reflective of the kinds of app store evaluations that are commonly available. Create a sentiment analysis model that works effectively. The model should be able to recognise each review's sentiment properly and extract characteristics that can be used to distinguish between honest and dishonest reviews. The machine learning model should be trained on a labelled dataset of reviews and tested on a different test set in order to gauge its performance and accuracy.

Comparing the performance of the sentiment analysis model to other methods will help you decide whether it is successful at spotting fraudulent applications when compared to other methods like rule-based techniques or other machine learning models. Examine the following elements that affect app store fraud: The study goal should also involve looking at the elements that lead to fraud in app stores, such as phoney reviews or deceptive advertising, by examining the fraudulent reviews that the sentiment analysis model discovered.

The goal of the research project on sentiment analysis-based fraud app detection is to give app store consumers a trustworthy and effective method for spotting fake apps while also enhancing the safety and privacy of app users.

# 

# 5. METHODOLOGY

Our paper's major goal is to present a strategy for extracting useful data from mobile applications based on user reviews, ratings, and ranking, and aggregating this evidence to find applications that are fraudulent. Our algorithm will make use of each user review and rating in order to determine how each user feels about the subject.

DATA COLLECTION DETAILS:

The gathering of data is crucial to machine learning. The process of acquiring and analysing data from various sources is known as data collection. In order for machine learning to more precisely identify some input parameters, it needs a large amount of data with many features. The crucial component that enables algorithm training is data collecting. A higher number of qualities has been shown to produce better results. Training data is gathered from Training data for fraud app detection. The algorithm was trained using a training dataset so that it could learn and deliver results. The training set includes 13000 submissions (reviews, sentiment value). 50% of the reviews in the training dataset are favourable, while 50% are unfavourable. Testing dataset: The model and method were assessed using the training dataset. Real-time data that is taken from the Google Play Store makes up the testing dataset.

DATA PREPROCESSING:

Pre-processing is the process of transforming data into a form that a computer can comprehend. The dataset that was collected during data collection is not in a format that the classifier can use. The dataset must undergo a variety of data preparation and feature extraction processes in order to be appropriate for the creation of classification models. The pre-processing methods on the dataset are carried out using the Python module Pandas. stages in pre-processing are:

**Tokenization**

Tokenization is the process of breaking up a long passage of text into smaller lines, words, or even new words for languages other than English. The nltk module itself comes with several built-in tokenization features.

**Stop words Elimination**

Stop Word Elimination is a method of removing pointless information from data. stop words are words that are not useful in NLP.

**Converting to lowercase**

All of the upper-case letters in this are changed to lower case.

**Vectorizer TFID**

The Tiff Vectorizer can encode new documents, tokenize existing ones, learn the vocabulary, and compute inverse document frequency weightings.

# 6.EXPERIMENTAL SETUP

The following stages are commonly included in the experimental setup for identifying fraud apps using sentiment analysis. Data collection: Several app shops were used to gather a sizable dataset of app reviews. This dataset ought to contain both honest and false reviews. Data preparation: Stop words, punctuation, and other noise, such as special characters, are removed from the acquired data during pre-processing. To maintain consistency, the data is tokenized and all the words are changed to lower case. Feature extraction: Relevant characteristics, such as the sentiment of the review, the frequency of particular words or phrases, the length of the review, and the language employed, are extracted from the pre-processed data using sentiment analysis algorithms.

Training and evaluation: Using a labelled dataset of reviews, where each review is classified as either genuine or fraudulent, a machine learning model is trained on the extracted characteristics. The performance and accuracy of the trained model are then assessed on a different test set.

Comparison with alternative approaches: In order to assess the success of the sentiment analysis model in identifying fraudulent applications, it is contrasted with alternative strategies like rule-based techniques or other machine learning models.

Examining the usefulness of the sentiment analysis model in spotting

counterfeit applications, the trial findings are evaluated. By examining the false reviews, the model found, the causes of app store fraud, such as bogus reviews and deceptive advertising, are also examined.

Reporting: A scientific article or technical report that includes a thorough explanation of the experimental setup, the results attained, and the results' interpretation presents the experiment's findings. Furthermore, highlighted are the study's limitations and possible future research areas.

Overall, the experimental setup for using sentiment analysis to identify fraud apps entails gathering and pre-processing data, extracting features, training and evaluating a machine learning model, comparing with other approaches, interpreting the outcomes, and reporting the findings in a technical report or scientific paper.

# 7.CONCLUSION

Using the use of web-based social networking research, this study successfully developed an enhanced sensation characterisation technique for peculiarity location. Using twitter data as a contextual analysis, the feasibility of the suggested method is demonstrated. Using the suggested approach, the strangeness estimate designs were efficiently identified and translated. The contextual analysis demonstrated the usefulness and supremacy of the method. In terms of handling conclusion design characterizations, our method was accepted considering the extraordinary level of agreement created by similar grouping assignments carried out by human annotators. This investigation presents fresh ideas for formulating a thorough opinion assessment method using data from web-based social networking sites to identify instances or examples of discrepancy.

For businesses to strengthen their administrative infrastructure, for politicians and government leaders to understand the rationale behind their ongoing polling results, and for other for-profit organisations to improve their client incentives and brand promises, this should be extremely profitable.

## 8. TENTATIVE CHAPTER PLAN FOR THE PROPOSED WORK

**CHAPTER 1: INTRODUCTION**

With the rise of mobile applications, fraudsters have also found new ways to deceive users. Malicious apps disguised as legitimate ones have become a common problem in app stores, which leads to not only financial loss but also personal information exposure. It is crucial to detect such apps as early as possible to protect users' security and privacy. Next, we will describe the machine learning models used in our study, including logistic regression, decision trees, and support vector machines. We will explain the parameters and tuning methods used for each model and compare their performance on the dataset. We will also analyse the impact of different evaluation metrics on the results.

**CHAPTER 2: LITERATURE REVIEW**

Mobile applications have become an essential part of modern life, and their popularity continues to grow. However, this growth has also attracted the attention of fraudsters who seek to deceive users with malicious apps. Fraudulent apps can cause significant financial losses and can also expose users' personal information, such as credit card details and login credentials. Thus, it is crucial to detect such apps as early as possible to protect users' security and privacy.

**CHAPTER 3: OBJECTIVE**

The objective of this research paper is to explore the effectiveness of using sentiment analysis for detecting fraud apps in the mobile application market. The paper will aim to develop a robust model that can analyze user reviews and ratings to identify fraudulent apps that deceive users for financial gain or other malicious purposes. The study will evaluate the performance of the model against existing fraud detection techniques and assess its accuracy, precision, recall, and F1-score. The results of this research can potentially provide valuable insights into the development of more effective and efficient fraud detection methods for mobile app markets, thus contributing to the enhancement of user trust and safety.

**CHAPTER 4: METHODOLOGIES**

In this research paper, we will adopt a mixed-methods approach that combines qualitative and quantitative analysis techniques. We will collect a large dataset of user reviews and ratings from popular mobile app markets and use natural language processing (NLP) techniques to pre-process the data. We will then apply sentiment analysis algorithms to extract sentiment polarity and subjectivity features from the reviews. Next, we will use machine learning models, such as logistic regression, decision tree, and random forest, to classify the reviews into fraudulent or legitimate categories based on the sentiment analysis features. We will evaluate the model's performance using metrics such as accuracy, precision, recall, and F1-score. Finally, we will compare the results of our model with existing fraud detection methods to validate its effectiveness.

**CHAPTER 5: EXPERIMENTAL SETUP**

The experimental setup for detecting fraud apps using sentiment analysis involves the following steps. First, a dataset of user reviews and ratings for various apps is collected from app stores. Second, the dataset is pre-processed by removing irrelevant information and converting text to lowercase. Third, a sentiment analysis model is trained using a machine learning algorithm, such as logistic regression or support vector machines, to classify reviews as positive, negative or neutral. Fourth, the trained model is tested on a separate test dataset to evaluate its accuracy and performance. Finally, the model is applied to detect fraudulent reviews and apps based on the sentiment scores and patterns of reviews.

**CHAPTER 6: CONCLUSION AND FUTURE SCOPE**

Finally, this study showed the efficacy of using sentiment analysis to identify fraudulent applications. We can spot trends of fraudulent activity and differentiate them from genuine applications by analysing user reviews. This method of fraud detection can enhance the accuracy and effectiveness of the mobile app business. In the future, this study could be expanded by investigating other machine learning methods for scam detection, such as anomaly detection and neural networks. Furthermore, the efficacy of this method can be verified further by applying it to a bigger dataset and evaluating it against real-world instances of app deception. Overall, this study offers up new possibilities for enhancing the security and integrity of mobile software networks.

## REFERENCES

1. Safrin, Raheesa, K. R. Sharmila, TS Shri Subangi, and E. A. Vimal. "Sentiment analysis on online product review." Int. Res. J. Eng. Technol 4, no. 04

(2017).

2. Shahana, P. H., and Bini Omman. "Evaluation of features on sentimental analysis." Procedia Computer Science 46 (2015): 15851592.

3. “Muhammad Taimoor Khan, Mehr Durrani, Armughan Ali, Irum Inayat, Shehzad Khalid & Kamran Habib Khan “Sentiment analysis and complex natural

language” (2016) a springeropen journal

4. Narayanan, Ramanathan, Bing Liu, and Alok Choudhary. "Sentiment analysis of conditional sentences." In Proceedings of the 2009 Conference on

Empirical Methods in Natural Language Processing: Volume 1-Volume 1, pp. 180-189. Association for Computational Linguistics, 2009.

5. Li, Gang, and Fei Liu. "A clustering-based approach on sentiment analysis." In 2010 IEEE International Conference on Intelligent Systems and Knowledge

Engineering, pp. 331-337. IEEE, 2010.

6. Xu, Jianlin, Yifan Yu, Zhen Chen, Bin Cao, Wenyu Dong, Yu Guo, and Junwei Cao. "MobSafe: cloud computing based forensic analysis for massive

mobile applications using data mining." Tsinghua science and technology 18, no. 4 (2013): 418-427.

7. Tichkule, Ashwini, Nidhi Nikhar, Dewanand Kapgate, and Omkar Dudhbure. "Revelation of Fraud Apps using Sentiment Analysis App Reviews,

8. Rohini, V., and M. Thomas. "Comparison of Lexicon based and Naïve Bayes Classifier in Sentiment Analysis." International Journal for Scientific

Research & Development 3, no. 4 (2015).

9. Guzman, Emitza, and Walid Maalej. "How do users like this feature? a fine grained sentiment analysis of app reviews." In 2014 IEEE 22nd international

requirements engineering conference (RE), pp. 153-162. IEEE, 2014.

10. Zhou, Xujuan, Xiaohui Tao, Jianming Yong, and Zhenyu 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), pp. 557-562. IEEE, 2013.