**Conestoga College  
School of Applied Computer Science & Information Technology**

**SENG8080 - Case Studies Big Data**

**Fake App Detection**

**Ajay Luhar – 8897306**

**Sweet Patel - 8906674**

**Riyen Mehwala – 8826190**

**Nilesh Pandey – 8842221**

**Big Data Solution Architecture, Conestoga College (Downtown)**

**SENG8080-23F-Sec1-Case Studies**

**David Marsh**

**Oct 21, 2023**

**Abstract**

In this context, we're talking about detecting fake apps using app data, particular user reviews, total downloads, and user ratings. The abstract will expand as the project proceeds to include a full review of the study, with the final report providing a more extensive description lasting roughly two paragraphs.

The difficulty of identifying between genuine and fake mobile applications is central to our work—the preliminary results and strategies for addressing this critical issue. Detecting phone apps in the constantly growing app marketplace is getting increasingly difficult. Our study uses user reviews, the total number of downloads, and user ratings to establish a robust technique for identifying counterfeit apps. This study delves into these data sources' subtleties and interactions, giving insight into their potential in-app authentication. While the present abstract is brief, the full study will dig into approaches and techniques to analyze user reviews for trends that may suggest inauthenticity. It will also explain how the overall number of downloads and user ratings may be utilized as useful indications in this process.

**Table of contents**

|  |  |
| --- | --- |
| Topic | Page number |
| Introduction | 4 |
| Data Research and Integration | 5 |
| Data Collection | 6 |
| Data Storage and Maintenance | 7 |
| Data Quality | 8 |
| Data Analysis and Visualization | 8 |
| Extrension | 8 |
| References | 9 |

**Introduction**

Cell phones have become a crucial part of our lives in the present computerized age, giving us comfort, diversion, and a variety of available utilities. Clients now and again depend on application information, for example, input from others, complete downloads, and client evaluations, to make instructed decisions about which applications to introduce, with a vast number of applications open across different application stores. Nonetheless, as versatile application prevalence has developed, so has the occurrence of telephone and hazardous applications. These false projects could jeopardize clients' security, well-being, and universal experience.

The increasing number of fake apps, sometimes known as "scam apps" or "malicious apps," poses a serious issue to both app store platforms and consumers. Fake applications sometimes simulate well-known and reliable companies or provide supposedly appealing functionality to get users to download them. Once installed, these applications can perform a variety of dangerous behaviors, including stealing sensitive data or bombarding users with invasive advertisements, harming user confidence and device security.

To deal with this growing problem, advanced fraudulent app detection algorithms must be developed. This project aims to address this issue by utilizing app data, including customer feedback, the overall number of downloads, and user ratings, as useful markers for identifying real apps from their fake ones. We can design powerful algorithms and models that help identify phone applications by utilizing the app user community's collective wisdom and popularity measures.

This Project aims to thoroughly examine fake app detection algorithms that will assist end users, app store platforms, and developers. This project aims to improve app store security, user data protection, and the general quality of the mobile app environment by analyzing the depth of available app data.

Furthermore, we will look at the limitations and constraints of false app identification and future approaches for enhancing the accuracy and efficiency of those detection systems. The ultimate objective is to provide users with the knowledge and tools to make educated decisions when selecting and interacting with mobile apps in a constantly linked and dynamic digital environment.

**Data Research and Integration**

In the context of our project, data research and integration play a crucial role in the development of fraudulent app detection algorithms. We brainstormed multiple sources, such as Kaggle, and then concluded that web scraping would be the best tool in our project. One powerful tool at our disposal for acquiring and updating the data we need is web scraping.  
Web scraping will be used to collect data from the following sources:

* **App Store Platforms**: We will do Web scraping to retrieve information about all the apps available on different app store platforms. This data includes app names, developer details, download counts, and user ratings.
* **User Reviews and Feedback**: Extracting user-generated reviews and feedback from app store pages is vital for understanding user sentiments and identifying potential issues with applications. This user feedback can serve as a valuable resource for training and refining detection models.

**Benefits of Web Scraping:**

* Real-Time Data – Web scraping allows us to capture real-time data.
* Efficiency – Web scraping can be set up to run at specified intervals, reducing the need for manual data gathering and integration.

**Data Collection**

In our project, data collection is a pivotal step in developing robust fraudulent app detection algorithms. Here's how we gather relevant data:

**Method Selection:**

After careful consideration, we decided to use a Python program employing Python's app store library. This approach allows us to extract essential app information, including user reviews, ratings, total number of downloads, and other pertinent data. This method ensures that we have access to accurate and comprehensive app details.

**CSV File Creation:**

To maintain structured and organized data, we generate CSV (Comma-Separated Values) files for each app's information. These CSV files serve as structured datasets, making it easy to manipulate and analyze the data.

**Benefits of CSV Files:**

* Structured Data: CSV files are a widely accepted format for storing structured data.
* Ease of Analysis: CSV files are compatible with various data analysis tools.

**Data Storage and Maintenance**

In our project, effective data storage and maintenance are essential components to ensure that we can efficiently manage and access the collected app-related data. Here's how we achieve this:

**SQLite Database:**For storing the app-related data, we have chosen to utilize an SQLite database. SQLite is a lightweight, embedded relational database management system that is well-suited for smaller-scale applications and is particularly useful for our needs.

**Python Data Loading Script:**To facilitate the process of loading data into our SQLite database, we have created a Python script. This script serves as an intermediary, enabling us to transfer data seamlessly between Excel files and our database.

**Data Loading Process:**The data loading script is designed to take data from Excel files, typically stored in CSV format, and load it into the SQLite database. This two-way process involves importing new data into the database and updating existing records when necessary. This script ensures that our database remains synchronized with the evolving app landscape.

**Benefits of SQLite for Data Storage:**

* **Lightweight** – SQLite is a self-contained, serverless database engine, making it highly efficient and easy to set up.
* **Portability**: SQLite databases are file-based, making them portable and compatible with various platforms and systems.

**Data Maintenance and Updates:**To keep our database up to date, we follow a systematic approach. Whenever we acquire new data, whether from web scraping or other sources, we use the same Python script to update the database. This ensures that our database always reflects the most current information.

**Data Quality –** Work in Progress

Data comes from all sources and can be accurate or reliable to varying degrees. Data quality tried to track and maintain this.This is the area in which you will talk about how you will maintain high quality data. **Data Analysis and Visualization –** Work in Progress

In this section, you will discuss the findings and answers to the questions you set out to answer in your introduction and abstract.

**Extension –** Work in Progress

We're adding an online view to our continuing fraudulent app detection initiative. This website is an easy-to-use interface, allowing everyone to access our data analysis and visualizations quickly. Users may interact with the data, study patterns, and customize their experience using filters. The website encourages openness, gives educational tools on data analysis, and delivers current updates to keep the information up-to-date and relevant.

**Proposed Allocation Project Team Roles**

This is the section for you to outline ‘who does what’ in your team and how they do it. Normally, projects work best when individuals take primary responsibility for certain areas (e.g. Write Up, Design, Data acquisition, etc). At this stage, you should have a more detailed

Additionally, this is a section to discuss how your team will communicate and meet to get the work done (e.g. schedule, timeline, etc.).

This section exists to make sure you have, as a team, discussed and agreed on how you will get the work completed.

**Project Timeline**

This is primarily for your benefit to make sure that you are scheduling your time well in preparation for a successful project completion. This does not need to be especially detailed but should be a useful tool to help you plan your time and work.

A partial example of a schedule is below. Please note, it does not (and probably should not) be identical to this: develop something that will help you!

|  |  |  |
| --- | --- | --- |
| **Date** | **Deliverable** | **Responsible** |
| Sep 23 | Finalised the Project topic | Everyone |
| Sep 28 | 1st Project Draft was shared with the Team | Nilesh, Riyen |
| Oct 5 | Collection of Data Sources through Web Scraping | Ajay, Sweet |
| Oct 12 | Python Script to Extract Data | Ajay, Nilesh |
| Oct 19 | CSV work | Riyen, Sweet |

**References**