DETECTION OF FRAUD APPLICATION FOR MOBILES USING THE CONCEPT OF DATA MINING

Logeswari.G(421212205016), Jayasri.G(421212205011)

Mr. P. V. Gopirajan. Asst. prof. Department of IT, gopirajan.kvcet@gmail.com

Department of Information Technology, Karpaga Vinayaga College of Engineering and Technology

Maduranthakam Taluk, PIN: 603308

infojayasrig@gmail.com

ABSTRACT:
Ranking fraud in the mobile App market refers to fraudulent or deceptive activities which have a purpose of bumping up the Apps in the popularity list. Indeed, it becomes more and more frequent for App developers to use shady means, such as inflating their Apps’ sales or posting phony App ratings, to commit ranking fraud. While the importance of preventing ranking fraud has been widely recognized, there is limited understanding and research in this area. To this end, in this paper, we provide a holistic view of ranking fraud and propose ranking fraud detection system for mobile Apps. Specifically, we first propose to accurately locate the ranking fraud by mining the active periods, namely leading sessions, of mobile Apps. Such leading sessions can be leveraged for detecting the local anomaly instead of global anomaly of App rankings. Furthermore, we investigate three types of evidences, i.e., ranking based evidences, rating based evidences and review based evidences, by modeling Apps’ ranking, rating and review behaviors through statistical hypotheses tests. In addition, we propose an optimization based aggregation method to integrate all the evidences for fraud detection.

Finally, we evaluate the proposed system with real-world App data collected from the iOS App Store for a long time period. In the experiments, we validate the effectiveness of the proposed system, and show the scalability of the detection algorithm as well as some regularity of ranking fraud activities.

Index Terms—Mobile Apps, ranking fraud detection, evidence aggregation, historical ranking records, rating and review.

INTRODUCTION:
Ranking fraud for mobile app market refers to fraudulent or deceptive activities which have a purpose of bumping up the apps in the popularity list. It becomes more frequent for app developers to use shady means, such as increasing their apps sales to commit ranking fraud. We provide holistic view of ranking fraud and propose a ranking fraud detection system for mobile apps. We investigate three type of evidences: Ranking based evidences, Rating based evidences, Review based evidences.

Existing System:
In the literature, while there are some related work, such as web ranking spam detection, online review spam detection and mobile App recommendation, the
problem of detecting ranking fraud for mobile Apps is still under-explored. Generally speaking, the related works of this study can be grouped into three categories. The first category is about web ranking spam detection. The second category is focused on detecting online review spam. Finally, the third category includes the studies on mobile App recommendation.

DISADVANTAGES OF EXISTING SYSTEM:
Although some of the existing approaches can be used for anomaly detection from historical rating and review records, they are not able to extract fraud evidences for a given time period (i.e., leading session). Cannot able to detect ranking fraud happened in Apps’ historical leading sessions. There is no existing benchmark to decide which leading sessions or Apps really contain ranking fraud.

PROPOSED SYSTEM:
We first propose a simple yet effective algorithm to identify the leading sessions of each App based on its historical ranking records. Then, with the analysis of Apps’ ranking behaviors, we find that the fraudulent Apps often have different ranking patterns in each leading session compared with normal Apps. Thus, we characterize some fraud evidences from Apps’ historical ranking records, and develop three functions to extract such ranking based fraud evidences. We further propose two types of fraud evidences based on Apps’ rating and review history, which reflect some anomaly patterns from Apps’ historical rating and review records.

In Ranking Based Evidences, by analyzing the Apps’ historical ranking records, we observe that Apps’ ranking behaviors in a leading event always satisfy a specific ranking pattern, which consists of three different ranking phases, namely, rising phase, maintaining phase and recession phase. In Rating Based Evidences, specifically, after an App has been published, it can be rated by any user who downloaded it. Indeed, user rating is one of the most important features of App advertisement. An App which has higher rating may attract more users to download and can also be ranked higher in the leaderboard. Thus, rating manipulation is also an important perspective of ranking fraud. In Review Based Evidences, besides ratings, most of the App stores also allow users to write some textual comments as App reviews. Such review scan reflect the personal perceptions and usage experiences of existing users for particular mobile Apps. Indeed, review manipulation is one of the most important perspective of app ranking fraud.

ADVANTAGES OF PROPOSED SYSTEM:

The proposed framework is scalable and can be extended with other domain generated evidences for ranking fraud detection. Experimental results show the effectiveness of the proposed system, the scalability of the detection algorithm as well as some regularity of ranking fraud activities. To the best of our knowledge, there is no existing benchmark to decide which leading sessions or Apps really contain ranking fraud. Thus, we develop four intuitive baselines and invite five human evaluators to validate the effectiveness of our approach Evidence Aggregation based Ranking Fraud Detection (EA-RFD).

SYSTEM ARCHITECTURE:
In the first module, we develop our system environment with the details of App like an app store. Intuitively, the leading sessions of a mobile App represent its periods of popularity, so the ranking manipulation will only take place in these leading sessions. Therefore, the problem of detecting ranking fraud is to detect fraudulent leading sessions. Along this line, the first task is how to mine the leading sessions of a mobile App from its historical ranking records. There are two main steps for mining leading sessions. First, we need to discover leading events from the App’s historical ranking records. Second, we need to merge adjacent leading events for constructing leading sessions.

B. Ranking Based Evidences:
In this module, we develop Ranking based Evidences system. By analyzing the Apps’ historical ranking records, web serve that Apps’ ranking behaviors in a leading event always satisfy a specific ranking pattern, which consists of three different ranking phases, namely, rising phase, maintaining phase and recession phase. Specifically, in each leading event, an App’s ranking first increases to a peak position in the leaderboard (i.e., rising phase), then keeps such peak position for a period (i.e., maintaining phase), and finally decreases till the end of the event (i.e., recession phase).

C. Rating Based Evidences:
In the third module, we enhance the system with Rating based evidences module. The ranking based evidences are useful for ranking fraud detection. However, sometimes, it is not sufficient to only use ranking based evidences. For example, some Apps created by the famous developers, such as Gameloft, may have some leading events with large values of u due to the developers’ credibility and the “word-of-mouth” advertising effect. Moreover, some of the legal marketing services, such as “limited-time discount”, may also result in significant ranking based evidences. To solve this issue, we also study how to extract fraud evidences from Apps’ historical rating records.

D. Review Based Evidence:
In this module we add the Review based Evidences module in our system. Besides ratings, most of the App stores also allow users to write some textual comments as App reviews. Such reviews can reflect the personal perceptions and usage experiences of existing users for particular mobile Apps. Indeed, review manipulation is one of the most important perspective of App ranking fraud. Specifically, before downloading or purchasing a new mobile App, users often first read its historical reviews to ease their decision making, and a mobile App contains more positive reviews may attract more users to download. Therefore, imposters often post fake reviews in the leading sessions of a specific App in order to inflate the App downloads, and thus propel the App’s ranking position in the leader board.

E. Evidence Aggregation:
In this module we develop the Evidence Aggregation module to our system. After extracting three types of fraud evidences, the next challenge is how to combine them for ranking fraud detection. Indeed, there are many ranking and evidence aggregation methods in the literature, such as permutation based models score based models and Dempster-Shafer rules. However, some of these methods focus on learning a global ranking for all labeled training data.

DATA FLOW DIAGRAM:
The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by the system. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.

GLOBAL ANAMOLY:
UML DIAGRAM:
UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML. The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:
Primary goals in the design of the UML are as follows:

Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models. Provide extendibility and specialization mechanisms to extend the core concepts.
Be independent of particular programming languages and development process.
Provide a formal basis for understanding the modeling language. Encourage the growth of OO tools market. Support higher level development concepts such as collaborations, frameworks, patterns and components. Integrate best practices.

USE CASE DIAGRAM:
A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

SEQUENCE DIAGRAM:
A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in
what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

ACTIVITY DIAGRAM:
Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

REFERENCE: