Fake Ad Click Detection by Identifying Anomalies

Student Name :	Gautam Shanbhag
Student ID :	18210455
Email :	gautam.shanbhag2@mail.dcu.ie
Programme:	MCM (Data Analytics) Full-Time
Module Code :	[CA640] Professional & Research Practice
Date of Submission :	30/Nov/18
Title :	Fake Ad Click Detection by Identifying Anomalies
Supervisor :	Mark Roantree

DISCLAIMER

A report submitted to Dublin City University, School of Computing for module CA640 Professional and Research Practice, 2018/2019. I understand that the University regards breaches of academic integrity and plagiarism as grave and serious. I have read and understood the DCU Academic Integrity and Plagiarism Policy. I accept the penalties that may be imposed should I engage in practice or practices that breach this policy. I have identified and included the source of all facts, ideas, opinions, viewpoints of others in the assignment references. Direct quotations, paraphrasing, discussion of ideas from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged and the sources cited are identified in the assignment references. I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work. By signing this form or by submitting this material online I confirm that this assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study. By signing this form or by submitting material for assessment online I confirm that I have read and understood DCU Academic Integrity and Plagiarism Policy (available at: http://www.dcu.ie/registry/examinations/index.shtml)

Name: Gautam Shanbhag

Date: 30/Nov/2018

Fake Ad Click Detection by Identifying Anomalies

Gautam Shanbhag
MCM Data Analytics
Dublin City University
Dublin, Ireland
gautam.shanbhag2@mail.dcu.ie

Abstract—**Click Fraud is a technique of deceitfully increasing the number of clicks on a pay-per-click ad. There are two common driving factors for click fraud; one is an attempt by the advertisers to sabotage their competitors by increasing their costs and exhausting their budget caps early. Second is the practice in which Ad publishers try to generate more revenue for themselves by clicking on the ads displayed on their site. The aim of this practicum is to predict whether a user will download an app after clicking a mobile app advertisement. This is achievable by measuring the journey of a user's click throughout the session and flagging IP addresses that produce many clicks, but never end up installing apps. I will perform this analysis by clustering, classifying and using machine learning modeling.**

Keywords—Fraud Detection, Anomalies, Pattern Recognition, Machine Learning, Clustering, Classification.

I. Introduction

Pay-per-click model is suffering from tremendous fraud issues. In many cases, the clickers click on advertisements without having actual interest in the ad's link. This enables them earn extra money for depleting the advertiser's budget. Click fraud refers to all such fraudulent means used for generating more clicks without being recognized by the search engine [1].

Loss estimation of Mobile app advertisers in 2013 was nearly 1 billion dollars (12% of the mobile ad budget) [2]. The frauds fall under two main categories:

(1) Bot-driven frauds employ bot networks or paid users to initiate fake ad impressions and clicks [2]

(2) Placement frauds manipulate visual layouts of ads to trigger ad impressions and unintentional clicks from real users (47% of user clicks are reportedly accidental [3])

Traditional search advertising click fraud was mainly triggered by an automated program. This kind of click fraud has evident characteristics, such as high traffic within a short time, single access mode, repetitive, etc. We can filter out this kind of click fraud through online detection rules [4].

A click farm is a form of click fraud, where the click fraudster hires large groups of workers to click on paid advertising links. The workers click the links, surf the target website for a certain amount of time and in some cases they also sign up for newsletters before moving to another link. In such cases, the workers can earn huge amount of money by clicking on enough ads per day to hit their target. This gives them an alternative source of income. It is extremely troublesome for an automated channel to detect such simulated traffic as fake as the user behavior appears the same as that of a genuine user [5].

The phrase 'invalid clicks' is similar to click fraud, however, in addition to clicks that are intentionally fraudulent, we also consider some accidental clicks which are generated by genuine user clicks as invalid clicks. Invalid clicks can be troublesome for advertisers. Large advertising networks analyze and validate clicks on behalf of the advertiser. In the event of detection of such invalid clicks, the networks flag such records and remove them from payments and reports [6].

The aim of this project is to examine various machine-learning algorithms to identify which ones are apt in anomaly detection.

The rest of the paper is organized as follows. Section II provides a background of the relevant work in this domain. In Section III, the research plan for detecting Click Fraud and related work are present. This section is sub divided into steps taken or required to achieve the conclusion. Formulating data set to working on training and test data set to identify anomalies and classify into real or fraud clicks and also the turn around which resulted into leads would be analyzed and identified. Finally, the paper concludes with references.

II. Literature review

Recognizing Click-Fraud is a relatively new area of research. In minimalistic form, the advertisers detect frauds based on massive ad clicks crossing the set threshold. If a mobile app / web page is receiving a high number of clicks from the same IP address in a short interval, these clicks can be flagged as fraud. The detection becomes complicated if the clickers use proxy networks or global IP switching techniques. In such cases, the advertisers have to use global IP blacklists and periodically update the list as the number of bots increase, at the expense of losing income from genuine customer clicks [7].

Some research studies suggest using supervised learning methods for generic bot detection. Li et al presented a generalized botnet detection method using Particle Swarm Optimization (PSO) and K-means [8], where the authors used a dataset that contained a high proportion of bots, which is uncommon in real life. Zhao et al. [9] applied several machine-learning techniques to detect botnets of different types. However, this research used a merged dataset that was collected from different sources and contained 5.8% of malicious data. Merged dataset can be flawed, biased and may not represent the real world.

According to [10], prevention mechanisms are based on blocking suspicious traffic by IP, city, country, referrer, ISP, etc. An online database is maintained which contains these suspicious factors. This system is then tested using real world data from ad campaigns. It is concluded from the results that click fraud analysis quality can be improved using multi-level data fusion.

Antoniou D. and others in their IEEE Paper [11] on Exposing Click-fraud using a burst detection algorithm have used splay tree algorithm, which is a self-adjusting form of binary search tree by exploiting the time and space locality. They monitor the sequence of accesses to a webpage and the time span to design the splay tree.

III. RESEARCH PLAN

To initially start off the project, I took inspiration from research onion framework adopted by Saunders & Lewis (2012) [12]. This approach provided insights on step by step analysis that would be required to go about developing the research plan and finalizing the concept idea of this review.

A. Dataset Description

For any given idea to be tested and refurbished, data forms the core of the analysis and deduction to be followed. I will address three main data quality measures: accuracy, completeness, and consistency. The data set for my analysis was extracted form Kaggle provided by TalkingData, China's big data service platform who had made it open to community for help to build an algorithm that predict whether a user will download an app after clicking on the mobile app ad. The data set has more than 1 million records consisting of columns like timestamp, app id, app domain, app category, device id, device ip, device type, device model and few more fields that are categorical. The data is encrypted thoroughly.

The first step of my research plan is to check for noise in the data by using OpenRefine so to get more filtered and clean dataset. I would then split the data into Training Data & Test data in the ratio 70:30.

B. Exploratory data analysis and Clustering

Initially, I would analyze the data set to determine important characteristics of the data. Results from the analysis will help me define my clustering logic in a better way. I would then perform clustering to group the objects in my data set. I intend to use Density based Anomaly Detection, K-Means Clustering, Mean Shift Clustering, and Gaussian Mixture Models and later perform comparison on the output of these models. The best-suited model and its output will be used for further modelling.

C. Machine Learning Model Training & Testing

I propose to perform Random Forest with Recursive Feature Elimination algorithm to identify the traits and patterns to identify potential fraud clicks present in the dataset. I will be studying how I can use frameworks like Keras on my dataset. Then I will be testing various machine learning algorithms and train them with the relevant dataset.

D. Analyzing the output of the Classifier

Numenta Anomaly Benchmark sets the benchmark for Anomaly detection procedures [13]. I intend to test the different result set by comparing them using different techniques and NAB. In addition, I will test this algorithm using the test data set. Thus evaluating the efficiency. This will help me make a statement on the percentage of fraudulent data present in the dataset and the efficiency to further detect fraud clicks by the designed algorithm.

IV. CONCLUSION

There are various algorithms that can be used for fraud detection and classification; however, the efficiency of every algorithm differs while looking at different factors. I want to consider the capability of different machine learning algorithms where I will analyze characteristics of dataset to identify anomalies in detecting ad click fraud. After reading different published journals and papers, I have a clearer picture of the working of different machine learning algorithms. This will help me pick a beneficial machine-learning algorithm, which would have better accuracy in identifying click fraud. This output will be based on factors such as the efficiency of the algorithm, time required, click per second, bounce back rate, device IP etc.

REFERENCES

[1] K C Wilbur, Y Zhu, D S. Anderson, "Click Fraud[J]", Access & Download Statistics, vol. 28, no. 2, pp. 293-308, 2009.

[2] Urbanski Al., (01 May 2013). “Bots Mobilize”, DMN [Online]. Available: http://www.dmnews.com/bots-mobilize/article/291566/.

[3] Liu, B & Nath, S & Govindan, R & Liu, J. (2014). DECAF: Detecting and characterizing ad fraud in mobile apps. Proc. of the 11Th USENIX Symposium on Networked Systems Design and Implementation (NSDI’14). 57-70.

[4] X. Jiarui and L. Chen, "Detecting Crowdsourcing Click Fraud in Search Advertising Based on Clustering Analysis," 2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom), Beijing, 2015, pp. 894-900.

[5] Munson, Lee (October 15, 2007). "What is a Click Farm?". Security-FAQs. Retrieved January 10, 2017.

[6] Beal V. “Click Fraud”, Webopedia [Online]. Available: https://www.webopedia.com/TERM/C/click_fraud.html

[7] H. Haddadi, "Fighting Online Click-Fraud Using Bluff Ads", ACM SIGCOMM, vol. 40, n. 2, pp. 22-25, April 2010.

[8] S.H. Li, Y.C. Kao, Z.C. Zhang, Y.P. Chuang, and D.C. Yen, “A network behavior-based botnet detection mechanism using PSO and K-means,” ACM TMIS, vol. 6, p. 3, April 2015.

[9] D. Zhao et al., “Botnet detection based on traffic behavior analysis and flow intervals,” Comput. Secur., vol. 39, pp. 2-16, November 2013.

[10] C. Walgampaya, M. Kantardzic, R. Yampolskiy, "Real Time Click Fraud Prevention using multi-level Data Fusion", WCECS 2010, October 20-22, 2010, San Francisco, USA

[11] D. Antoniou et al., "Exposing click-fraud using a burst detection algorithm," 2011 IEEE Symposium on Computers and Communications (ISCC), Kerkyra, 2011, pp. 1111-1116.

[12] https://onion.derby.ac.uk/

[13] Lavin, A. & Ahmad, S., 2015. Evaluating Real-time Anomaly Detection Algorithms—the Numenta Anomaly Benchmark. Miami, IEEE.