International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 - 5161

Available at http://inpressco.com/category/ijcet

General Article

Collation of Strategies for Click Fraud Detection using same IP Address

Bhavini Kanoongo[Å*], Puja Jagani[Å] and Khushali Deulkar[Å]

[Å]Computer Dept., Mumbai University, D.J.Sanghvi COE ,Vile Parle(W), Mumbai, India

Accepted 02 Sept 2014, Available online 01 Oct 2014, **Vol.4, No.5 (Oct 2014)**

Abstract

Due to the accelerated advancement of the Internet, online advertisement plays an essential role in the advertising market. One of the present and conventional revenue models for online advertising comprises of charging for each click depending on the reputation of keywords and the number of contending advertisers. This pay-per-click model gives a chance to individuals or rival companies to generate false clicks (i.e., click fraud), which act as a serious problem to the evolution of healthy online advertising market. Click fraud is the intended clicking on advertisements with no actual interest in the product or the service offered. It is one of the most critical problems in online advertising. It is thus very important to build detection methods for click fraud. Using a combination of statistical measures, timing window concept and zero access auto clicking as well as search hijacking, we document the distinct methods used by every module, the architecture they use, and how they operate to dupe online advertisers.

Keywords: *Data Mining, Fraud detection.*

1. Introduction

The voluntary act of clicking on a search engine sponsored listing or banner ad with the purpose of fraudulently increasing clicks while exhausting the advertiser's pay-per-click budgets is the fundamental of click fraud. While a few of the marketers feel that click fraud undermines the pay-per-click market, others agree that click fraud declines the success rate of many online marketing programs.

Click fraud can be committed for several reasons. In some cases, click fraud can occur independently. In other cases, it may be a malignant and well-structured method of producing boundless and fallacious click activity. Few search engines with pay-per-click or sponsored listings have established a network of associate sites that distribute listings based on search term results in what is known as contextual advertising. The ads or sponsored links are mixed in the related context of the website. For example, as search request for the purchase of a new Apple iPad may direct the user to an ecommerce Web site that displays a sponsored link along with the different versions of iPad on sale. Few of these culprits have developed sophisticated technologies and business processes in order to falsely generate clicks without getting detected. However, others which are less sophisticated also pose a risk and burden for accurate measurement of a campaigner's online success.

Unusual patterns become likely when visitors from pay per click marketing are monitored by the site and baseline visitor behavior is established. Immunity can become rare because everyone is a victim. The key is to track the subtle indicators which help to identify possible click fraud using conversion rates which means the amount of clicks required for a relevant activity, multiple clicks from a unique IP address, etc.

The targets of click fraud constitute of the search engine marketer, the consumer, advertiser and ultimately the search engine or pay-per-click network. This affects the expected returns of advertising which reduces the results and escalates the budgets unnecessarily. This ruins the client relations because the results are being delivered at a sub-optimal level. Return on ad dollars is analyzed carefully irrespective of the size of the organization. Many companies depend on sponsored listings to confirm their high rank in search engines. Sponsored listings influence customers in their web searches. Sponsored listings are economically dependent on click fraud income, hence a reduction in the same will lead to advertisers not spending and the search engines will suffer accordingly.

Section 2 discusses about the various methods by which click fraud occurs and how it can be detected. Section 3 deals with the review of different models and techniques to detect click fraud.

2. Review of literature

We first consider the problem of detecting duplicate clicks over jumping windows, and introduce group Bloom filters and timing bloom filters. The basic concepts to be known before we proceed are:

Landmark window (Richard Oentaryo *et al*, 2013): First N elements are considered to be landmark windows. Instead of maintaining all elements in current window a sketch can be used when processing data streams over landmark windows. The expired sketch can be replaced

*Corresponding author: **Bhavini Kanoongo**

Table 1 Comparative study parameters

Parameters for comparison	**Detecting duplicates over decaying window models**	**Detecting auto-click Using statistical methods**	**Detecting Zero Access Botnet fraud**
Basic Idea	It defines a timing threshold and only counts identical clicks once within the timing window.	It extracts publisher's feature from various statistics such as mean, standard deviation, count from different views and different time granularity of the publisher.	It studies the working of the Zero-Access botnet in detail to prevent attacks from it.
Working of the Method/ Model used	It uses the concept of timing windows where the identical clicks will not count if they are within short time interval, and will count if they happen sparsely.	It uses various statistics from various views and different time granularity of the publisher and chose the most discriminative ones to build an elective classifier.	The malware execution environment, manual analysis techniques, and the Zero-Access modules are studied so that such frauds can be detected.
Main purpose of the Method/ Model	Decaying windows eliminate expired information and only use fresh information of clicks.	To develop and crowd source data mining and machine learning methods capable of building elective predictive models to detect fraudulent publishers.	To be able to understand the working behind the botnet and track the techniques used so that similar fraud can be detected the next time.
Main approaches used	There are two common types of decaying windows used: • Count-based windows which maintain the last (most recent) N items in the data stream. • Time-based windows which maintain all items that arrived in the last T time units.	The main approach used is the linear blending of many predictive models. It consists of three main steps: • Creation of own validation set from the training set. •Optimization of meta-parameters of models. •Retraining the model on the combined train and internal validation sets.	There are two approaches used: • It detects the Auto-clicking of the botnet which is found to occur every two minutes. • It studies the mechanism of Search-hijacking and tries to find the loop holes so that it can be detected.
Models/attributes used	Decaying window models, such as landmark, jumping and sliding window models are used.	Click statistics of each publisher, unique count of attributes such as numericip, country, deviceua, referredurl, campaignid and total visits are used.	The models used are Auto-clicking module and the search-hijacking module of the Zero-Access botnet.
How does click fraud occur	An algorithm is used to simulate a real user.	It is generated using: • Botnets (where malware on the user's computer clicks on ads in the background). • Tricking or confusing users into clicking ads (e.g., on parked domains). • Directly paying users to click on ads.	It uses two methods: • Search-hijacking sends a real user to the advertiser. • In Auto-clicking Zero-Access module automatically clicks on advertisements.
Methodology of detection of click fraud	It uses Timing Bloom Filters (TBF) and Group Bloom Filters (GBF) algorithms.	It uses the classification method of linear blending which are tree/rule-based except for Bayesian network and resilient propagation (RPROP).	• Pseudo-domains and IP addresses are extracted from the search-hijacking module via malware executions and reverse engineering. • The complete method of auto-clicking is studied in detail.
Advantages/issues solved by the technique	It uses very little space and operation and makes only one pass over the click streams.	To prune inconsequential features and improve prediction performance, iterative feature elimination was performed.	• Auto-clicking requires no user participation, and is not visible to the user. • In search hijacking, the advertiser's site is relevant to the user's search query and hence the user may interact with the advertiser's site and trigger a conversion.
Disadvantages	It is not as effective and efficient as detecting click fraud using statistical methods.	It is not completely accurate.	It is very difficult to detect the fraud committed by the Zero Access botnet.

easily since all the elements last for the same amount of time.

Sliding window (Richard Oentaryo *et al*, 2013): A sliding window, first introduced by Datar *et al*, 2004, only contains the last N items, which is updated once a new element comes and an old element expires. To update the required statistics when then window slides, timing information is maintained as the elements expire 1 by 1.

Jumping window (Richard Oentaryo *et al*, 2013): The jumping window model was first proposed by Zhu and Shasha (R. Kohavi, 1996). The Landmark window and the Sliding window meet halfway to form the Jumping window.

Burton H. Bloom (A. Chao *et al*, 2003) introduced Bloom Filters in 1970, and they have been used in many areas such as networking and database (N. V. Chawla, 2002). A set of n elements, to reply to membership queries can be represented by a Bloom Filter which is a space-efficient data structure. Bloom filters can be directly deployed (I. Good, 1953), for detection of duplicates in click streams over a landmark window. Each click's identifier such as the source IP ad-dress, or the cookie, etc. is hashed into the Bloom filter. A click is reported duplicate, if its identifier is already present in the Bloom filter (i.e., all corresponding bits are 1s). The entire jumping window is equally divided into sub-windows and a Bloom filter is maintained for each sub-window, for the detection of duplicates. The same set of harsh functions is used by the Bloom filters to save operation time. For jumping windows with few sub-windows, the GBF algorithm works efficiently. The limitation of this algorithm is that it is not feasible in the sliding window model, or when the number of sub-windows Q is very large in a jumping window (Richard Oentaryo *et al*, 2013). Although, N Bloom filters can be kept, each to hold only one element, however the maintenance of N Bloom filters is very time consuming.

The ZeroAccess auto-clicking module performs click fraud by simulating normal Web browser behavior of a user clicking on a Web advertisement (Paul Pearce *et al*, 2013). The auto clicking module mimics a user "click" on an advertisement. The module takes the help of command-and-control servers with an appeal for click fraud jobs. An unassembled payload of a list of click jobs is returned by the C&C server. Each job is identified by a host name, a first hop URL and the HTTP Referer URL (Paul Pearce *et al*, 2013). An HTTP request is sent by the module to another CF-C&C server, which sets the Host header value according to the corresponding job. The first hop in the redirection chain is formed when the server redirects the request via an HTTP 303 redirect to a URL, the same URL as in the job. The Referer header is set by the retrieved URL to which it is redirected, by the module. The bot is then used to fetch the URL after different redirects which results in the advertiser being charged. The lack of activity in the foreground makes it tough for a user to detect any kind of click fraud occurrence. This same process repeats multiple times.

The Search-hijacking module keeps track of the data flow between the user and an infected PC and the browser waiting for the user to issue a search query to a search engine. The module identifies and confiscates web searches done using Google, Bing Yahoo, Ask and ICQ search, is confirmed. As the query goes through the destined search engine, the query terms are captured by the module. To recall a list of ad URLs useful for later hijacking, the search-hijacking module's C&C (SH-C&C) server also receives the query terms. Hijacking of the normal click and replacement of the destined URL with the ad URL fetched from the SH-C&C server is performed, when the user clicks on a search or ad result. The replacement URL is retrieved and delivered by the browser, instead of the search result URL. A chain of HTTP and JavaScript redirects may be involved in the retrieval of the replacement URL.

After studying the working of the above two models of ZeroAccess, related patterns are found when they occur again.

In order to capture the fraudulent visitors from the same IP address, calculations of the mean access, standard deviation, taking a count by grouping each IP for each publisher in different time granularity (by second, by min, by day) are performed. For example, let there be the full click log for the publisher \8kxij", the average access by IP in minute granularity is 8, standard deviation is 1, and the counting for the IP 2; 919; 155; 822 visit is 8. Identical statistics can be obtained in another time granularity such as by day, hour, second as well.

3. Comparative Study

It is given in table 1.

Conclusions

In this paper, we address the problem of detecting duplicate clicks in pay-per-click streams over jumping windows and sliding windows. We propose group Bloom filters which significantly reduces the memory operations when processing click streams, and GBF and TBF algorithms based on the Bloom Filters. We have also described two ZeroAccess modules, one of which is the auto-clicking module which achieves traditional click fraud by simulating user clicks on advertisements, and a much latest search-hijacking module, which intercedes upon user clicks on Web search results, rather than sending the user to an advertisement related to the search. The study of simple statistics such as average, count, etc. as well as the Analysis of time series at different granularity levels (e.g., sec, min, hour and day) helped in detecting click fraud.

In the future, we will continue to explore the issues of click fraud and try to prevent them. We will consider various sophisticated click fraud attacks, and study advertising network dynamics, new techniques of detection, economic and social impacts of click frauds.

References

Richard Oentaryo, Ee-Peng Lim, Michael Finegold, David Lo, Feida Zhu, Clifton Phua, Eng-Yeow Cheu, etc, (2013), Click Fraud in Online Advertising: A Data Mining, *Journal of Machine Learning Research*, 14 1-39.

Linfeng Zhang and Yong Guan (2008), Detecting Click Fraud in Pay-Per-Click Streams of Online Advertising Networks, *The 28th International Conference on Distributed Computing Systems*, 1063-6927/08 $25.00 © 2008 IEEE.

Paul Pearce, Chris Grier, Vern Paxson, Vacha Dave, Damon McCoy, Geoffrey M. Voelker, Stefan Savage (2013), The ZeroAccess Auto-Clicking and Search-Hijacking Click Fraud Modules, *Technical Report No. UCB/EECS-2013-211*.

Net Applications, Exposing click fraud, *Using Web Analytics To Identify Possible Click Fraud A white paper from net applications.com.*

C. Chen, A. Liaw, and L. Breiman (2004), Using random forests to learn imbalanced data, *Technical report, Technical Report No. 666*, Department of Statistics, University of California, Berkeley.

R. Kohavi (1996), Scaling up the accuracy of naive Bayes classifiers: A decision-tree hybrid, *Proceedings of the AAAI International Conference on Knowledge Discovery and Data Mining*, pages 202{207, Portland, OR, 1996.

A. Chao and T. Shen (2003), Nonparametric estimation of shannon's index of diversity when there are unseen species in sample, *Environmental and Ecological Statistics*, 10:429, 443, 2003.

N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer (2002), SMOTE: Synthetic minority over-sampling technique, *Journal of Artificial Intelligence Research*, 16:321{ 357, 2002.

I. Good (1953), The population frequencies of species and the estimation of population parameters, *Biometrika*, 40:237{264, 1953.