SPRINGER BRIEFS IN COMPUTER SCIENCE

Xingquan Zhu
Haicheng Tao
Zhiang Wu
Jie Cao
Kristopher Kalish
Jeremy Kayne

Fraud Prevention in Online Digital Advertising

Springer

SpringerBriefs in Computer Science

More information about this series at http://www.springer.com/series/10028

Xingquan Zhu • Haicheng Tao • Zhiang Wu
Jie Cao • Kristopher Kalish • Jeremy Kayne

Fraud Prevention in Online Digital Advertising

Springer

Xingquan Zhu
Dept. of Computer & Electrical Engineering and Computer Science
Florida Atlantic University
Boca Raton, FL, USA

Haicheng Tao
Dept. of Computer & Electrical Engineering and Computer Science
Florida Atlantic University
Boca Raton, FL, USA

Zhiang Wu
College of Information Engineering
Nanjing University of Finance and Economics
Nanjing, China

Jie Cao
College of Information Engineering
Nanjing University of Finance and Economics
Nanjing, China

Kristopher Kalish
Bidtellect, Inc.
Delray Beach, FL, USA

Jeremy Kayne
Bidtellect, Inc.
Delray Beach, FL, USA

ISSN 2191-5768 ISSN 2191-5776 (electronic)
SpringerBriefs in Computer Science
ISBN 978-3-319-56792-1 ISBN 978-3-319-56793-8 (eBook)
DOI 10.1007/978-3-319-56793-8

Library of Congress Control Number: 2017942765

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Foreword

Advertising has a long history which can be traced back to ancient civilization. Four thousand years ago in early Egypt, Greece, China, and India, people used wall or rock paintings for commercial advertising. In England during the eighteenth century, advertisements started to appear in weekly newspapers which marked the start of modern advertising. Advertising continued to advance with technology and offered increased audience targeting and reach via newspapers, radio, TV, and many other modern devices. The birth of the Internet in the twentieth century was a groundbreaking platform allowing advertisers to immediately reach out to individuals, collecting their response and feedback in real time. Such evolution marks a new discipline, computational advertising, which uses computing technology to offer more efficient and effective advertising.

In digital advertising, advertisers, publishers, and audiences are connected over the Internet. The lack of physical interaction and the requirement of real-time response make our advertising ecosystem vulnerable to fraud. These fraudulent attacks come in many forms including non-human traffic, Bot clicks, unintended clicks, or stacked and unviewable Ad banners. The need for countermeasures has never been so urgent and challenging given that both advertisers and publishers are facing huge volumes of traffic on a daily basis and fraudsters are stealing a significant portion of revenue and continuously deteriorating the whole ecosystem (online ad fraud driven by bots costs brands over $7 billion globally in 2016 alone).

This book offers the first comprehensive study of fraud in digital advertising. The fraud map, or taxonomy as noted in the book, provides a clear view of most known fraud in the industry. The study of fraud prevention and commercial systems allows academic researchers and industrial developers to fully understand the problems from both scientific and practical perspectives. As we are embracing the big data era, fraud detection and prevention represents a big challenge that the whole industry is facing. I am glad that this book timely captures such challenges and is moving the industry ahead to provide better and safer advertising.

Delray Beach, FL, USA — Lon Otremba
January 17, 2017

Preface

Computational advertising refers to computing platforms or systems, which use computational approaches to optimally match audiences and advertisement. In advertising ecosystems, revenue is the driving force fueling the whole system. Publishers, advertisers, and third-party service providers are all trying to maximize the return by following business rules and procedures. Yet, the highly specialized business models and the nature of the cyberspace information switch without physical interactions make it difficult to directly evaluate whether the advertisements are indeed served to genuine human viewers, or clicks are actually generated by human users. As a result, it creates an environment for fraudsters to use deceptive approaches to attack advertising systems for illicit returns. This reality has raised serious integrity concerns, because a fraud flooded advertising market hardly has any value to stakeholders, and fraud must be well controlled, if not completely removed, for the healthy growth of the digital advertising market.

In this monograph, we systematically review forms of online digital advertising (Ad) fraud and the techniques to prevent and defeat them. We categorize Ad fraud into three major categories, including (1) placement fraud, (2) traffic fraud, and (3) action fraud. We summarize major features of each type of fraud and also outline measures and resources to detect each type of fraud.

The book provides a comprehensive guide to help researchers comprehend the state of the art in Ad fraud detection. It also serves as a technical reference for industry to design new techniques and solutions to win the battle against fraud.

Boca Raton, FL, USA — Xingquan Zhu
Boca Raton, FL, USA — Haicheng Tao
Delray Beach, FL, USA — Kristopher Kalish
Delray Beach, FL, USA — Jeremy Kayne
January 15

Acknowledgements

On January 26, 2016, Florida Atlantic University's College of Engineering and Computer Science announced that it received a $300,000 gift from Bidtellect, a global leader in native advertising technologies and solutions. The gift will allow the Department of Computer and Electrical Engineering and Computer Science (CEECS) to engage in research, develop and establish curriculum, facilitate laboratory improvements, fund research assistantships, and provide an annual award to the top Bidtellect Laboratory researcher.

We are grateful to the Bidtellect for their continuous support in establishing FAU Bidtellect Laboratory and funding provided for the research projects. "This generous gift from Bidtellect enforces the mission of the College of Engineering and Computer Science to offer innovative educational experiences to its students alongside our faculty researchers," said Mohammad Ilyas, Ph.D., dean of the College of Engineering and Computer Science.

H. Tao, Z. Wu, and J. Cao are partially supported by National Natural Science Foundation of China (91646204, 71571093, 71372188) and National Center for International Joint Research on E-Business Information Processing (2013B01035).

Contents

1 Introduction 1
1.1 Computational Advertising 1
1.2 Display Advertising 2
1.3 Fraud in Digital Advertising 3
1.4 Book Objective 5
References 5

2 Ad Ecosystems and Key Components 7
2.1 Ad Banner and Viewability 8
2.2 Ad Business Model 12
2.3 Revenue Models 13
2.4 Ad Advertisement Concept Flow 14
References 17

3 Ad Fraud Taxonomy and Prevention Mechanisms 19
3.1 Ad Fraud Taxonomy 19
3.2 Ad Fraud Prevention Mechanism 21
References 22

4 Ad Fraud Categorization and Detection Methods 25
4.1 Placement Fraud 25
4.1.1 Stuffing or Stacking 26
4.1.2 Fake Sites 26
4.1.3 Domain Spoofing 27
4.1.4 Ad Injection and Malware 28
4.2 Traffic Fraud 29
4.2.1 Impression Fraud 30
4.2.2 Click Fraud 30

4.3 Action Fraud 33
4.3.1 Conversion Fraud 33
4.3.2 Re-targeting Fraud 34
4.3.3 Affiliate Fraud 35
References 36

5 Ad Fraud Measure and Benchmark 39
5.1 Measures with Ground Truth 39
5.2 Measures Without Ground Truth 41
5.3 Real World Datasets 42
References 43

6 Ad Fraud Detection Tools and Systems 45
6.1 Commercial Ad Fraud Detection Systems 45
6.2 Ad Fraud Detection Systems in Academia 47
References 48

7 Conclusion 51

Glossary 53

Acronyms

Ad	The abbreviation for advertising or advertisement.
CPM	Cost Per Mille (CPM) is the unit price based on one thousand impression.
CPC	Cost Per Click (CPC) is the unit price based on one click event.
CPA	Cost Per Action (CPA) is the unit price based on one conversion event.
CTR	Click Through Rate (CTR) is the ratio of clicks on an Ad (or a page, site etc.) divided by the number of impressions of the Ad (or the page, site etc).
DSP	Demand Side Platform (DSP) is for advertisers to manage/plan advertising campaigns.
DMP	Data Management Platform (DMP) is a system for creating audiences based on first-party and third-party data so that advertisers may target those audiences.
eCPC	Effective Cost Per Click (eCPC) is calculated using the costs of buying the impressions divided by the number of clicks generated from the impressions. CPC is normally used for defining price of buying clicks, whereas eCPC is a metric defining the equivalent CPC prices of buying impressions.
eCPM	Effective Cost Per Mille (eCPM) is calculated using the total costs spent on an Ad campaign divided by number of impressions resulted from such costs (normalized in 1000 impressions). CPM is normally used for defining price of buying impressions, whereas eCPM is a metric defining the equivalent CPM prices of buying methods (e.g. fixed price, CPM, CPC, or CPA).
KPI	Key Performance Indicator (KPI) may refer to any metric, such as CPA or CTR, used to evaluate the performance.
SSP	Supplier Side Platform (SSP) is an analytics tool for publishers to manage advertisement placements.

Chapter 1
Introduction

Half the money I spend on advertising is wasted; the trouble is I don't know which half.

– John Wanamaker (1838–1922)

Abstract In this chapter, we briefly introduce the computational advertising, including search advertising and display advertising. We explain the reality of fraud in digital advertising, and summarize types of fraud methods commonly observed in the industry for making illicit returns.

1.1 Computational Advertising

Recent advancement in networking and communication technologies have witnessed a rapid growth of digital advertising [1], which uses the Internet to promote and deliver advertisements (Ad) to consumers [24]. Compared to traditional media, such as Radio, TV, or news papers, the Internet offers tremendous advantages such as real-time interaction, consumer information availability, transparent user engagement, and effective assessment of the campaign results etc. As a result, on-line digital advertising is quickly dominating the advertising market, and its market revenue is projected to reach over $250 billions in 2018 [22]. One of the prominent and most sought characteristics of the Internet is that it allows the Ad industry to obtain fine-grained information from specific geographic locations, regions, households, or even individual users, and is able to serve highly customized advertisement to users in real-time. Such tools and methods, used in digital advertising, are commonly referred to as *computational advertising*, which mainly covers two types of advertising: search advertising and display advertising. The essential goal is to identify the user context, such that the Ads most interesting to the users are served with minimum advertising costs.

In search advertising [9, 14] (also referred to as sponsored search), the context information is obtained through the search query users provided to the system. More specifically, the search keywords users entered in the search engine are used to find users' interests and the best matching Ads are then displayed to the users, along

X. Zhu et al., *Fraud Prevention in Online Digital Advertising*, SpringerBriefs in Computer Science, DOI 10.1007/978-3-319-56793-8_1

with the generic query results from the search engines. For search advertising, the challenge is to identify users' search goal through the entered query key-words, which are brief, noisy, and often inaccurate. For this purpose, a lot of research [7] has been conducted to understand the relationship between users' search queries and their information goals.

In display advertising [2], the Ads are displayed on the Ad banners [5, 12] which are part of the web page content displayed to the viewers (In this book, viewers and audience are equivalent terms, which both refer to end users viewing the Ad impressions). Under such circumstance, the contextual information about users is very limited, compared to search advertising. The most common way of obtaining the contextual information is to use cookie synchronization [18] which links users across different domains and sites to identify users' interests for targeting. Some methods also utilize the content information of the web page to identify user interests, which is often referred to as content-targeted advertising [23]. For example, native advertising [3] intends to show Ads which are closely related (or visually related) to the content displayed in the current page. Existing research [10, 19] has studied Ads matching website content vs. Ads with increased obtrusiveness, and observed that Ads doing both (Ads matching both website content and having increased obtrusiveness) have worse performance at increasing purchase intent than Ads that only focus on one or the other.

1.2 Display Advertising

In this book, we mainly focus on online display advertisement on desktops. The general concepts and business process discussed in the paper are also applied to other types of online advertising, such as online search advertising [17], online mobile advertising [4, 15], or specific media types, such as video content portals [16].

In computational advertising ecosystems, each Internet user is a potential Ad customer or target. When an Internet user sends a web URL to request content from a web server, the publisher (which hosts the web services), upon receiving the user's request, immediately obtains necessary user information, such as IP address, operating systems, and web browser types etc. Combining user information with the content of the web page requested by the user, the publisher is in a position to find advertisers who are interested in serving their advertisement to the user. To this end, the publisher submits user information as a *bid request* to a bidding market place, called *Ad Exchange* where advertisers are connected to place their bids in response to the bid request. Upon receiving bids from advertisers, the Ad exchange chooses the winner and passes the Ad link of the winning advertiser (who submitted the highest bid) to the publisher. The Ad links are immediately embedded to the web page, and the publisher returns Ad embedded web page to the users, which results in an *Impression* indicating that the advertisement is displayed once to the audience. This process concludes an Ad serving circle from audience to publisher, Ad exchange, advertisers, and then back to the audience. It is worth noting that

the whole process, from users sending out URL request to receiving the web page, happens in real-time with typical delay less than 100 millisecond (ms). An example of the online advertisement generated from the Ad system is shown in Fig. 2.2, where the web page shown on the user's web browser contains three advertisements (dashed rectangle boxes).

During an Ad bidding process, computational advertising plays essential roles for different parties, including publishers and advertisers. Advertisers want to serve their advertisements to users potentially interested in their products, while minimizing the cost, by working with a DSP. Meanwhile, publishers also want to maximize their revenue by selling Ad inventory (or traffic) to the advertisers whose Ads are most suitable to their audience, by working with an SSP which will attempt to optimize yield on the publishers behalf by choosing the right exchange or DSP to auction up the impression. The SSP may even serve the impression opportunity up to multiple buyers if hardware permits. In reality, because the whole bidding process requires real-time response, all stakeholders employ a programmatic bidding strategy to automate their bidding process. As a result, it makes the Ad system vulnerable to different types of attacks or fraudulent activities, which use unethical or illicit ways to gain revenue.

In online advertising business models, a publisher sells Ad inventory to advertisers, allowing them to display their advertisement on publishers' web pages which are visited by audiences through different devices, such as desktop computers, tablets, or mobiles. The buying and selling of the Ad inventory is made through an Ad exchange which brings buyers and sellers together to finish the transactions in real-time. While advertisers are only interested in serving Ads to human viewers, an unethical publisher may submit a fake impression (e.g. machine generated impression) to the bidding engine resulting in advertisers' advertisement being displayed to non-human user agents. This is, in fact, one of the most common forms of fraud (*Impression Fraud*) in digital advertising. Other types of fraud may involve sophisticated programming and coordinated user participation, interactions, etc., but all of them have an ultimate goal of obtaining financial revenue. Such fraudulent activities have become a significant burden to the digital advertising. For example, the Association of National Advertisers [21] projected that Ad fraud will cost $7.2 billions in 2016. For video advertising, Videology and WhiteOPS projected that 8–25% of online video Ad inventory is consumed by bots [20].

1.3 Fraud in Digital Advertising

Arguably, fraud attacks are one of the major threads of any business or financial systems, and online digital advertising is only worse. This is mainly because online digital advertising has an open platform and real-time transaction requirement, but the underlying foundation on which digital advertising technology is built is not designed with fraud prevention in mind [11].

For example, in 2011, a study [13] has shown that some pornographic sites secretly redirect users to non-pornographic sites without users' knowledge and then sell advertising spots of those visits for revenue (e.g. five million dollars in fraudulent revenue in 8 months [25]).

On December 2016, WhiteOPS [26] reported their discovery and solutions for fighting Methbot, which is a "bot farm" controlled by a single group based in Russia and operating out of data centers in the US and Netherlands, generating "$3 to $5 million in fraudulent revenue per day by targeting the premium video advertising ecosystem" [27].

For fraud in digital advertising systems, they may be either actively generated or passively generated. For example, a user may deliberately perform fraudulent actions to obtain financial gain, resulting in actively generated fraud which is typically malicious and has significant financial impact. On the other hand, digital advertising systems essentially live in a computing and networking environment, and are subject to the impact of network activities. For example, web crawlers are actively visiting websites to collect content to serve search engines. When a web crawler visits a web page, its actions resemble to a human user, so web servers may send traffic generated by the crawler to the Ad exchange for bidding. Under such circumstance, the underlying impression is essentially a fraudulent but is passively generated by a web crawler (which does not have the intention to cheat the Ad systems). This type of passively generated fraud is typically benign and has very little financial impact to the Ad systems. In this book, we mainly focus on actively generated fraud in digital advertising systems.

For actively generated fraudulent activities, they can be observed at nearly every corner of the Ad system, where different fraud objectives often result in different types of fraud behaviors. For example, at the network traffic level, frauds can be observed as fake impressions, in the sense that the traffic is not generated by human users and therefore does not merit severing or displaying any advertisement. At the action level, fraudsters can use scripting language to fire a mouse click event to simulate that a user is viewing and has clicked on the advertisement, and then claim commission from advertisers.

In a recent technical review [11], the experts from Index Exchange [8] and DoubleVerify [6] summarized following four types of fraud methods for making illicit returns:

- **Phony Traffic Brokers:** This type of fraud activity intends to simulate human traffic, and bring simulated traffic to a targeted site.
- **Ghost in the Machine:** This type of fraud activity uses malware or bots to generate fake audience or traffic.
- **Masking URLs in Bidstreams:** This type of fraud activity uses information in the bid requests, e.g. wrongly declaring the domain name to the exchange.
- **Hiding Ads:** This type of fraud activity generates impressions where Ads are never actually displayed to the audience.

While malicious fraudulent activities are becoming increasingly common in the Ad ecosystem, very little research work exists to systematically summarize fraud and their major characteristics. Many fundamental questions remain unclear for both academia researchers and industry practitioners, despite of the sheer capital size of the online advertising market and the overwhelming presence of fraudulent activities. What are the major types of fraud in Ad systems? what are the major methods used to detect such fraud? what are the measures or tools available to assess fraud and fraud behaviors? what are the ground truth or benchmarks available for research in this area etc.

1.4 Book Objective

Motivated by the above questions, in this book, we systematically study research activities and industry progress in tackling online advertising fraud. Our main objective is to provide a clear Ad fraud taxonomy, covering majority type of fraud in Ad systems. Our categorization will provide a tiered view of the system, by investigating major characteristics of different types of fraud, methods available to detect fraud, and resources available to stimulate the research and development in the area. We expect that this book will provide a guide for researchers in online advertising, as well as serving as a technical reference for industry practitioners or developers to design their own systems for fraud prevention.

The remainder of the book is structured as follows. Chapter 2 introduces Ad ecosystem, its main components, and key terminology. Chapter 3 summarises Ad fraud as a taxonomy, and Chap. 4 elaborates on fraud in different categories and explains corresponding detection methods. Chapter 5 addresses Ad fraud measures and benchmark, followed by Ad fraud detection tools and systems in Chap. 6. We conclude the book in Chap. 7.

References

1. Broder A, Josifovski V (2011) Introduction to computational advertising. https://web.stanford.edu/class/msande239/, Stanford University
2. Bureau IA (2016) Iab display advertising guidelines. https://www.iab.com/guidelines/iab-display-advertising-guidelines/
3. Carlson M (2015) When news sites go native: Redefining the advertising-editorial divide in response to native advertising. Journalism 16:849–865
4. Crussell J, Stevens R, Chen H (2014) Madfraud: investigating ad fraud in android applications. In: Proceeding MobiSys '14 Proceedings of the 12th annual international conference on Mobile systems, applications, and services, pp 123–134
5. Dichter E (1966) How word-of-mouth advertising works. Harvard Business Review 44:147–66
6. DoubleVerify D (2016) http://www.doubleverify.com/

7. Downey D, Dumais S, Liebling D, Horvitz E (2008) Understanding the relationship between searchers' queries and information goals. In: Proceedings of the 17th ACM conference on Information and knowledge management (CIKM), pp 449–458
8. Exchange I (2016) http://www.indexexchange.com/
9. Fain DC, Pedersen JO (2006) Sponsored search: a brief history. Bulletin of the American Society for Information Science and Technology 32:12–13
10. Goldfarb A, Tucker C (2011) Online display advertising: Targeting and obtrusiveness. Marketing Science 30:389–404
11. Heine C (2016) Here are 4 common methods that ad fraudsters use to make their ill-gotten money. http://www.adweek.com/news/technology/here-are-4-common-methods-ad-fraudsters-use-make-their-ill-gotten-money-169285
12. Heinz S, Hug M, Nugaeva C, Opwis K (2013) Online ad banners: the effects of goal orientation and content congruence on memory. In: Proceedings of CHI '13 Extended Abstracts on Human Factors in Computing Systems, pp 1875–1880
13. Ipeirotis P (2011) Uncovering an advertising fraud scheme. or the internet is for porn. http://www.behind-the-enemy-lines.com/2011/03/uncovering-advertising-fraud-scheme.html
14. Jansen BJ, Mullen T (2008) Sponsored search: An overview of the concept, history, and technology, international journal of electronic business. International Journal of Electronic Business 6:114–131
15. Liu B, Nath S, Govindan R, Liu J (2014) Decaf: Detecting and characterizing ad fraud in mobile apps. In: 11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14), USENIX Association, pp 57–70, URL https://www.usenix.org/conference/nsdi14/technical-sessions/presentation/liu_bin
16. Marciely M, Cuevas R, Banchsz A, Gonzalezy R, Traverso S, Ahmedy M, Azcorra A (2016) Understanding the detection of view fraud in video content portals. In: Proceedings of the 25th International Conference on World Wide Web, pp 357–368
17. Mladenow A, Novak NM, Strauss C (2015) Online ad-fraud in search engine advertising campaigns. Lecture Notes in Computer Science: Information and Communication Technology 9357:109–118
18. Olejnik L, Castelluccia C (2016) Analysis of openx-publishers cooperation. http://lukaszolejnik.com/OpenX-hotpets14.pdf
19. P H, B T (2014) Native advertising and digital natives: The effects of age and advertisement format on news website credibility judgments. ISOJ Journal 4:61–77
20. Paper W (2015) Eradicating bot fraud: The path to zero-tolerance. In: Videology and WhiteOPS
21. Paper W (2016) Bot baseline: Fraud in digital advertising. In: WhiteOPS and Association of National Advertising
22. Portal STS (2016) Digital advertising spending worldwide from 2012 to 2018. http://www.statista.com/statistics/237974/online-advertising-spending-worldwide/
23. Ribeiro-Neto B, Cristo M, Golgher PB (2005) Impedance coupling in content-targeted advertising. In: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, pp 409–503
24. Roth DW, Salisbury D (2001) Internet advertising system. US Patent US 6285987 B1
25. Stitelman O, Perlich C, Dalessandro B, Hook R, Raeder T, Provost F (2013) Using co-visitation networks for detecting large scale online display advertising exchange fraud. In: Proceeding KDD '13 Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD), pp 1240–1248
26. Whiteops (2016) http://www.whiteops.com/
27. Whiteops (2016) Learn about methbot and how to combat it. http://www.whiteops.com/methbot

Chapter 2
Ad Ecosystems and Key Components

Abstract In this chapter, we briefly describe the digital advertising ecosystem, mainly from the display advertising perspective. We will first describe the real-time bidding framework for online digital advertising, including technical platforms for publishers, advertisers, and the market place for online Ad inventory buying and selling. After that, we will describe major business model of online advertising, and explain three types of revenue models commonly used in Ad systems, including impression-based revenue model (CPM), click-based revenue model (CPC), and action based revenue model (CPA).

The content described in this chapter provides an overview of the online digital advertising ecosystem for better understanding fraud activities, their major characteristics, and corresponding detection mechanisms, which will be described in the following chapters.

An Ad system, as shown in Fig. 2.1, consists of two major parties: (1) sellers who provide Ad traffic inventory, and (2) buyers who buy traffics in order to deliver their advertisements to the audience. The subsystems corresponding to these two parties are called supplier side platform (SSP) and demand side platform (DSP), respectively. The key stakeholders on the seller side are publishers which provide web services to the public, and an Ad traffic inventory is generated whenever a user visits their websites. In other words, a user's request to visit the publisher's website creates an opportunity for the publisher to display one or multiple Ads to the user, and the publisher will therefore sell the opportunity to the advertisers who are interested in showing their Ads to the user.

At the buyer side, advertisers are the ones actively seeking to buy online traffic in order to serve advertisement to end users. An Ad exchange acts as a broker to connect buyers and sellers for them to exchange information, such that buyers and sellers can negotiate price and deliver advertisement to end devices in real-time. In the following, we will briefly explain online advertising business model, which provides essential background knowledge for understanding fraud in the Ad system.

X. Zhu et al., *Fraud Prevention in Online Digital Advertising*, SpringerBriefs in Computer Science, DOI 10.1007/978-3-319-56793-8_2

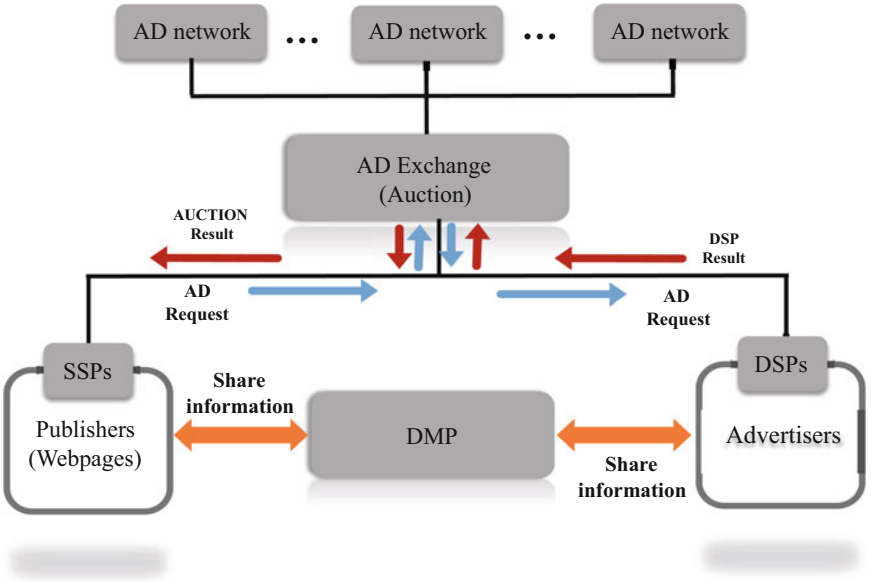

Fig. 2.1 An overview of the online digital advertising Real-time Bidding (RTB) framework. Supplier side platform (SSP) generates Internet traffic (or Ad inventory). Demand side platform (DSP) buys Internet traffic in order to display advertisements to audience. Ad exchange acts as a broker for buyer (DSP) and sellers (SSP) to exchange information and make buying/selling transactions in real-time. Both SSP and DSP may use data management platform (DMP) or third party data aggregator to analyze the data for better advertising. E.g. by using DMP, an SSP can direct their Ad inventory to the DSP best match their audiences' interests

2.1 Ad Banner and Viewability

In an Ad system, an Internet user is the one generating traffic (or inventory) and is also the advertising target. The eventual goal of advertising is to deliver and display relevant advertisements to users with genuine interest on their devices, ideally within the active viewport of the users' web browsers.

As shown in Fig. 2.2, each rectangle box used to display advertisement is called an Ad *banner*. For each webpage, the arrangement of the Ad banner on the page is called an Ad *placement*, which is typically characterized by its location on the page (e.g. on the top or the bottom of the page), sizes (width and height), etc. A placement can be as specific as an Ad banner on a particular page or broadly represent an entire website. Advertisers and publishers use Ad placement to define where Ads are displayed on the page, allowing placement targeting. Ad placements are differentiated by placement IDs, where a placement ID can be shared across multiple pages.

The position of the Ad banner on the page plays an important role in determining whether a displayed Ad will likely be clicked by users or not [1]. This is mainly because that when viewing a web page, users often have different visual attention on the page, as shown in Fig. 2.3. Typically, the content showing on the left side of

Fig. 2.2 Examples of online advertisement displayed on a user's web browser. Each *dashed rectangle box* denotes one Ad banner dedicated to display online advertisements. The Ad surrounded by *blue dashed line* is from Google AdSense and the *red one* is from other native advertising platform. The content (i.e. the actual Ad) displayed in the Ad banner is dynamically generated from a real-time bidding platform, so two users may view different advertisements even though they are visiting the same page at the same time and at the same geographic location

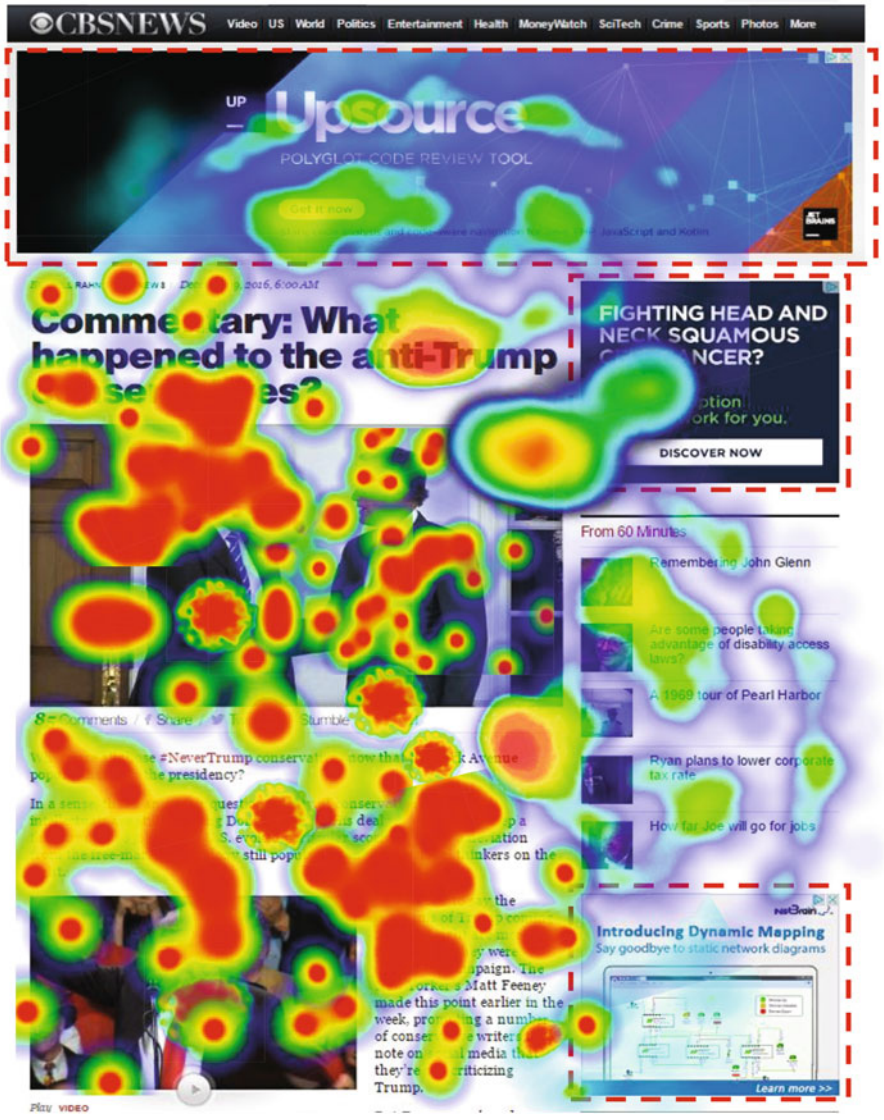

Fig. 2.3 User eye-tracking heatmap when viewing a web page. The *red-dashed rectangle boxes* denote Ad banners. Color coded regions denote visual attention heatmap. From *red colored regions*, to *yellow, green, and blue regions*, user visual attention will gradually decrease

the page will receive much more visual attention than the content showing on the right side. On the other hand, because users prefer viewing content without much visual interruption, Ad banners are often placed on the top or on the right panels of the page.

While the position of the Ad banner on the page is important, an Ad banner placed on a good position does not guarantee that the displayed advertisement is viewable to the audience. This is because that whether an advertisement is viewable or not is determined by many factors, such as the size of the Ad banner, its visibility etc. A recent study has shown that about 40–50% of online Ads served by publishers may be never seen by Internet users [9]. This is consistent with the observations from [8] which found that approximately 45% of Ads are in view for less than 1 s. Therefore, they referred to such Ads as "unviewable", and impressions that were in view for at least 1 s will be called viewable. Buying Ads unviewable to the audience only result in ineffective branding campaigns and a waste of money for advertisers.

Formally, an advertisement is called *viewable* if it is visually observable in user's active window, as shown in Fig. 2.4. An unviewable advertisement can be the one placed in a very small placement window, or never displayed in the active viewable window of the users' browsers. If an advertisement is not viewable, it has very little value to the advertiser. Therefore, accessing and predicting the viewability of the advertisement served to the users is an important, yet challenging, task [5, 11].

A common way of assessing the viewability is to obtain vertical location of Ad on the page, and a research from [8] has studied the frequency of Ads served at

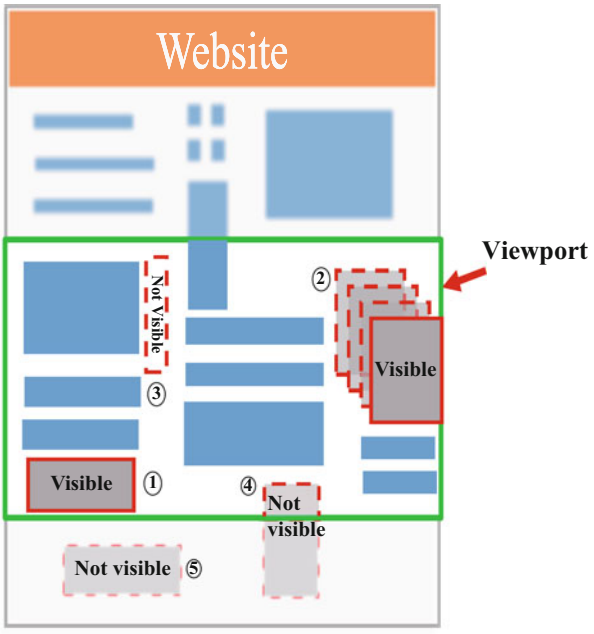

Fig. 2.4 A conceptual view of the Ad viewability. The *green solid rectangle box* denotes the active viewport on the user device. Each *rectangle box* denotes one Ad banner dedicated to display online advertisement. The Ad banner is viewable if its size is larger than a certain value (e.g. 100 × 100), and over 50% of its size is displayed for a certain period (say 2 s). The Ad surrounded by *red line* is viewable, whereas *red dashed box* is not viewable. Stacked Ads are in ② while the Ads are transparent in ③. Banners outside the browser window is "below the fold" and therefore considered not viewable

different locations on a page, and found that the probability of the impression being viewable drops quickly around the median location of the bottom of the browser window. By using locations, device type, scroll depth, and other features, it is possible to computationally estimate the viewability of an Ad on a particular page. For example, in [11], the authors studied a number of features that impact the scroll depth for a given user and a page, and proposed to use probabilistic latent class model to predict the viewability of any given scroll depth for a user-page pair.

2.2 Ad Business Model

When a company, say Nike or Dell, decides to engage to online advertisement for one or multiple of their products, they will first approach to an advertising company which will act as the advertiser on behalf of the company. For ease of understanding, we will refer to this company as the branding company through out the whole paper. The advertisement is therefore regarded as a *campaign* by the advertiser, in the sense that the advertisement will have a pre-specified advertising theme and objective defined by the branding company, such as geographic regions or user groups the advertisement aiming for. In addition, the advertiser and the branding company should also identify the type of campaigns, depending on their advertising objectives. For example, if the branding company only aims to carry out branding and shows their product to potential customers, it may choose CPM campaign with a negotiated price, say $5.0 CPM. Then the advertiser will receive $5.0 for every 1000 times the advertisement is displayed to viewers. On the other hand, if the company is interested in user participation, they may choose a $1.0 CPC campaign, so advertiser will receive $1.0 from the banding company, for each time a user clicks on the advertisement. The type of campaign determines the advertising goal, and fraud is often highly customized to target different types of campaigns.

Assume an advertiser has successfully launched a campaign, the next step is to find Ad inventory to display the advertisement, with minimum cost, such that they can claim maximum revenue from the branding company (under the agreed campaign objectives and KPIs). For this purpose, the advertiser will connect to Ad exchanges, through demand side platform (DSP), to participate real-time bidding to find valuable users and display advertisements. Such a biding platform is essentially triggered by an Internet user who requests to visit a web page owned by a publisher. As soon as the publisher receives an HTTP request from the user, it will send user information, as a bidding request, to the Ad exchange to call for advertisers for bidding. The type of bidding can vary, but normally follows either (1) *first-price auction*, or (2) *second-price auction*. In a first price auction, each advertiser will place a bid, and the winner (who submits the highest price) will win the bid, and pay the price as indicated in its bid. The winner of a second price auction, on the other hand, will only pay the price of the second highest bid plus a minor gap. In reality, second-price auctions are more common, because research has shown that the price

paid by second-price auction is close to the true value of the merchandise [4]. For example, as of this writing, over 80% of auctions received/participated by Bidtellect.com are second-price auctions.

It is worth noting that an advertiser is essentially a broker or agent in the sense that it will pay for buying the impression, and is also get paid by the branding company. The amount of commission an advertiser receiving from Ad branding companies is called *revenue*, and the amount of payment the advertiser spending for buying the impressions is called *cost*. The difference between revenue and cost will then form *gross profit*, which is an important factor to assess an advertiser.

2.3 Revenue Models

Overall, three types of revenue models are commonly used in Ad systems: impression-based (CPM), click-based (CPC), and action based (CPA) [10]. For impression based revenue models, both publishers and advertisers will receive revenue for displaying advertisement for a certain number of times (normally calculated per 1000 times), i.e. a CPM revenue model. The difference is that publishers will claim commission from advertisers, whereas advertisers will receive revenue from the branding company. For CPC based business model, a publisher will receive revenue for displaying the advertisement, whereas an advertiser will not receive revenue unless the displayed Ad results in a click event. For CPA based revenue model, an advertiser will receive revenue each time a customer completing a predefined business action, such as filling an online application form or placing an online order. For all three types of revenue models, CPC is the most common one in the Ad ecosystem.

In addition to CPC, CPM, and CPA revenues models, Ad systems also use effective CPM (eCPM), effective CPC (eCPC), and effective CPA (eCPA) to assess the actual performance of the system when using different types of models. For example, in an Ad campaign x, a branding company agrees to pay an advertiser CPC_x dollars for each user click on their advertisements. This is a typical CPC revenue model. After launching the campaign for a certain period of time, the advertiser has bought Imp_x impressions with an average CPM cost equal to CPM_x. In addition, all I_x impressions result in Clk_x clicks. As a result, the effective CPC, eCPC, of campaign x is calculated as follows

$$eCPC_x = \frac{Imp_x \times CPM_x}{1000 \times Clk_x} \tag{2.1}$$

Indeed, while CPC_x defines the revenue the publisher will receive from the advertiser for each user click, $eCPC_x$ defines the actual costs the advertiser spent on campaign x, normalized using the clicks. By comparing CPC_x *vs.* $eCPC_x$, advertiser or branding companies can directly assess the effectiveness of their advertising activities in terms of the costs and revenue. Similar principles also apply to *eCPM* and *eCPA* as well.

The unique business model where publishers receive revenue for displaying advertisements and user actions on the advertisements will subsequently bring revenue to the advertisers, provides motivation of using fraudulent activities to exaggerate impressions and user clicks which are strictly prohibited by stakeholders. For example, Google AdMob's "Invalid clicks and impressions" rules state that "Publishers may not click their own Ads or use any means to inflate impressions and/or clicks artificially, including manual methods. Testing your own Ads by clicking on them is not allowed" [6].

2.4 Ad Advertisement Concept Flow

We now detail a typical online advertising cycle, with a focus on interactions between different parties in an Ad ecosystem. When an Internet user visits a web page, and sends an HTTP request to the web server, showing on Step 1 of Fig. 2.5, this will trigger an Ad impression, if the requested web page contains any Ad banner, managed by publisher Ad servers.

In Fig. 2.5, as soon as a user sends an HTTP request to the publisher content server to request access to the content, e.g. a web page containing one or multiple banners, the content server will contact their Ad server to request advertisement

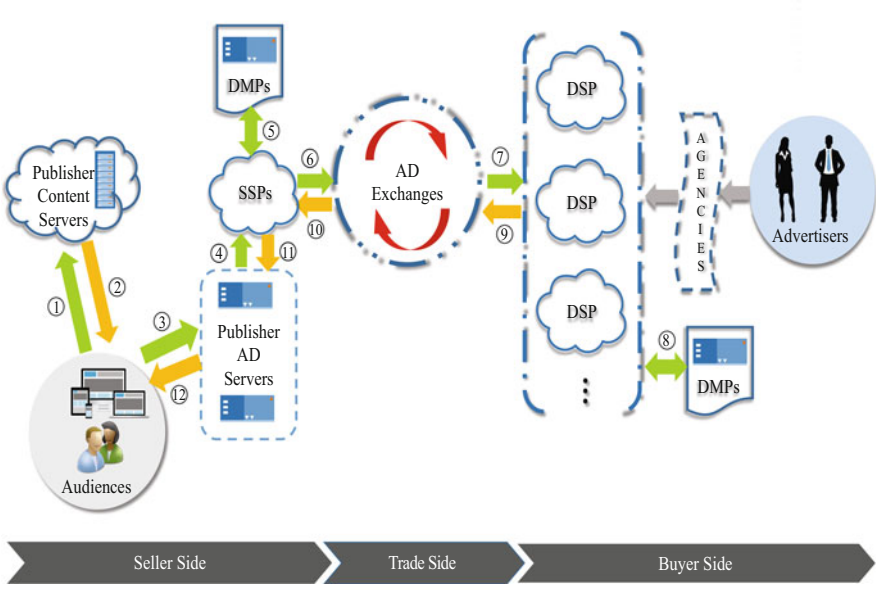

Fig. 2.5 A flow chart of the online display advertising system. The user traffic flow (or Ad inventory) is from Seller, Trade site, to Buyer, whereas the monetary flow is from Buyer, Trade site, to Seller

information to be embedded into the content page. In some cases, the Ad server may directly return Ad information to the content page. As a result, the web pages, containing advertisements, are returned to the audience. This often happens in direct-buy advertising where publishers and advertisers have signed up direct sale contacts. So Ads are delivered to audiences without any real-time bidding process.

In majority cases (or in real-time bidding scenarios), the Ad servers will contact (server side platform) SSP to preparing for bidding. SSP is essentially an automated programmatic technology platform allowing Ad publishers to manage their advertising space inventory, optimizing the selling of their online media space etc. The key role of the SSP is to allow publishers to connect their inventory to multiple Ad exchanges, DSPs, and networks at once, such that publishers can maximize their gain and control the selling price of their inventory with respect to different advertiser groups. Therefore, an important aspect of the SSP is to properly understand their inventory, Ad exchanges, and advertisers. For example, from the online user perspectives, SSP may want to know who are the audience of each publisher website, where are the audience from, and whether the audience have participated in any Ad campaigns before. From the Ad exchange perspective, SSP may want to know the fluctuation of the bidding/winning price of different Ad exchanges, with respect to different time of the day, so they can send their inventory to the most profiting Ad exchanges. The answers to all these questions are essentially resolved by using a data manage platform (DMP), which provides data support to the SSP. Upon necessary preparation, SSP is now ready to submit an auction to selected Ad exchanges, showing on Step 6 of Fig. 2.5.

An example of a bid auction request is shown in Fig. 2.6, which is essentially a JSON object containing a number of fields including site, page, Ad banner, impression, and user device information, etc. The actual format of the bid request is specified by the IAB OpenRTB specification [2] for programmatic Ad buying/selling.

The information provided in the bid request allows Ad Exchange and DSPs to understand the context of the Ad banners to be served to the audience, in addition to the bidding information such as bid floor price and bid currency etc. For example, the device information allows advertisers to determine creative and campaigns to be displayed at different types of devices. The site information, including site domain and page URLs, allows advertisers to know the destination of the Ads to be served, so they can ensure that their Ads are delivered to certain user groups.

While a bid request is meant to provide genuine information for auction, fraudsters may modify site and page information in the bid request [7], so advertisers' Ads may end up being displayed on sites different from the ones showing on the bid request. Such fraudulent activities have resulted in significant brand safety concerns for advertisers because they are trying to avoid showing Ads on certain sites [3].

Once an Ad exchange receives an ad request from publishers, it will broadcast the request as an ad auction to all demand side platforms (DSP) connected to the exchange. Similar to an SSP, a DSP is an automated programmatic advertising platform allowing Ad advertiser to manage and optimize the buying of online impressions. DSPs are crucial because they incorporate vital facets previously

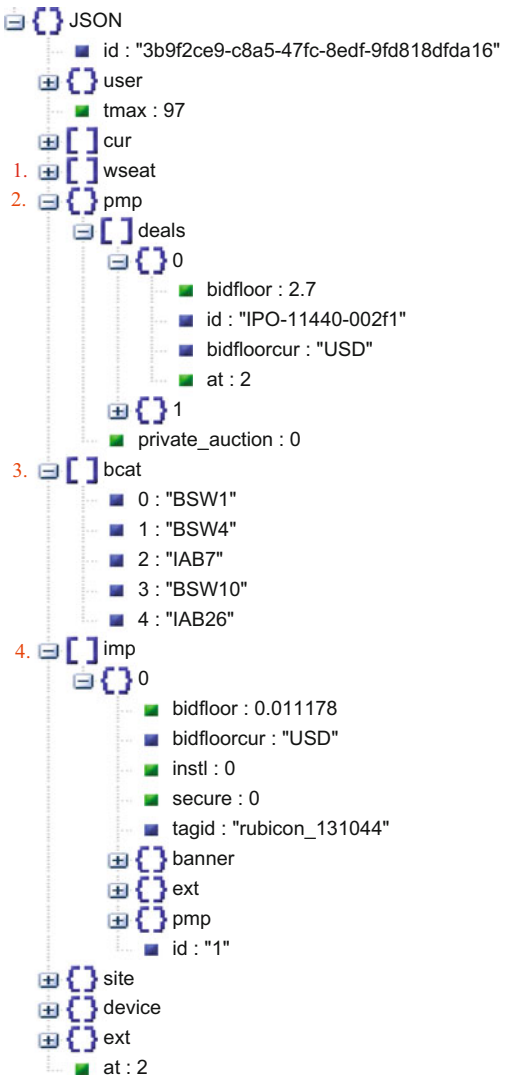

Fig. 2.6 A typical example of a bid request following the rules of openRTB. The data format of the bid request is JSON. Some optional parameters are: *1* "wseat" for the whitelist of seats which are advertisers or agencies. *2* "pmp" for a private marketplace of selected advertisers including minimum bid price "bidfloor". *3* "bcat" for blocked advertiser categories. *4* "imp" for the auctioned impression. A bid request must require at least one "imp"

offered by advertising networks, such as the information of the impressions from different publishers, the click through rate (CTR) of different Ad placements, the winning chance with respect to different publisher etc. Accordingly, DSPs offer wide access to inventory and vertical and lateral targeting, with the ability to serve Ads, real-time bid on Ads, track the Ads, and optimize the revenue. For example, after receiving an auction which typically includes the placement ID and publisher ID, the DSP can check the previous click through rate of the same placement, determine the placement's potential advertising value, and eventually help DSP put a suitable bidding price. For this purpose, a DSP will often rely on data management

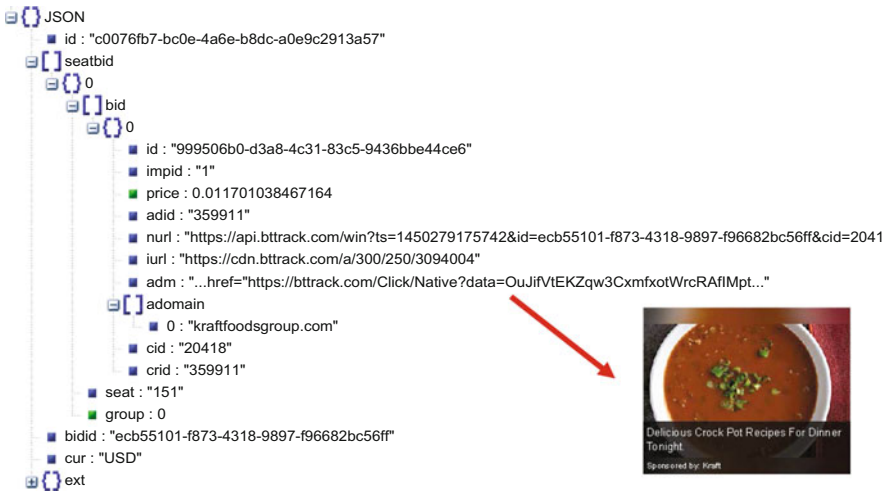

Fig. 2.7 A typical example of a bid response following the rules of openRTB. The data format of the bid response is also a JSON object. The main object of the bid repose is "seatbid" which is consisted of multiply bids with "price" for bid price, "nurl" for win notice URL and "adm" for Ad markup. The decoded Ad markup can be seen in the *lower right corner*

platforms (DMP) to provide effective data support. For example, before making a bid on an auction, DSP can pass cookie or page information in the bid request to the DMP, to obtain statistical information such as the number of impressions originated from a cookie. Such information can help the DSP assess whether the impression in the bid request is a fraud, and avoid biding on fraudulent impressions.

After the Ad exchange receives bid responses from all DSPs within the cutoff time limitations (typically less than 100 ms from casting the bid), showing on Step 9 of Fig. 2.5, it will select the one with the highest bidding price. An example of a bidding response from a DSP is shown in Fig. 2.7. The advertiser winning the bid will pass the information, such as the URL of their advertisement along with the script code, e.g. the adm filed in the bidding response in Fig. 2.7, to the SSP and then to the publisher Ad servers, showing on Steps 10 and 11 of Fig. 2.5. The publisher web content server and the Ad server then respond to the Internet user, with the web page and advertisements being served to the client devices.

The above process concludes a single Ad transaction, and in practice, this process happens in real-time with less than 100 ms delay. So users do not experience uncomfortable latency.

References

1. Agarwal A, Hosanagar K, Smith MD (2011) Real-time bidding benchmarking with ipinyou dataset. Journal of Marketing Research 49(16):1057–1073
2. Bureau IA (2016) Openrtb api specification version 2.5. http://www.iab.com/wp-content/uploads/2016/03/OpenRTB-API-Specification-Version-2-5-FINAL.pdf

3. Callejo P, Cuevas R, Cuevas n, Kotila M (2016) Independent auditing of online display advertising campaigns. In: Proceedings of the 15th ACM Workshop on Hot Topics in Networks (HotNets)
4. Edelman B, Ostrovsky M, Schwarz M (2007) Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. American Economic Review 97:242–259
5. Flosi S, Fulgoni G, Vollman A (2013) If an advertisement runs online and no one sees it, is it still an ad? empirical generalizations in digital advertising. Journal of Advertising Research 53(2):192–199
6. Google (2015) Google admob behavioral policies
7. Heine C (2016) Here are 4 common methods that ad fraudsters use to make their ill-gotten money. http://www.adweek.com/news/technology/here-are-4-common-methods-ad-fraudsters-use-make-their-ill-gotten-money-169285
8. Hill DN, Moakler R, Hubbard AE, Tsemekhman V, Provost F, Tsemekhman K (2015) Measuring causal impact of online actions via natural experiments: Application to display advertising. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pp 1839–1847
9. Bounie D, Valérie M, Quinn M (2016) Do you see what i see? ad viewability and the economics of online advertising. https://ssrn.com/abstract=2854265
10. Vratonjic N, Manshaei MH, Hubaux JP (2012) Online advertising fraud. In: Death Of the Internet, Markus Jakobsson edit.
11. Wang C, Kalra A, Borcea C, Chen Y (2015) Viewability prediction for online display ads. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pp 413–4222

Chapter 3
Ad Fraud Taxonomy and Prevention Mechanisms

Abstract In this chapter, we first propose a taxonomy to summarize fraud in online digital advertising. The taxonomy provides a complete view of major fraudulent activities in answering questions related to Who does What to Whom, and How. The proposed fraud taxonomy includes three major categories: placement oriented fraud, network traffic oriented fraud, and action oriented fraud. Placement oriented fraud mainly intends to manipulate or modify publisher pages or modify the web pages showing on the user's devices in order to increase the number of impressions or clicks. Traffic oriented fraud generates fake traffic to inflate the number of impressions or clicks generated from individual sites or placements. Action oriented fraud aims to target users' actions in order to generate revenue.

After that, we summarize four types of mechanisms for online Ad fraud prevention, including signature-based prevention mechanism, anomaly-based prevention mechanism, honeypot-based prevention mechanism, and credential-based prevention mechanism.

3.1 Ad Fraud Taxonomy

Pursuing high return on investment is the eventual goal for both advertisers and publishers. Unfortunately, to increase revenue, not all parties of the digital advertising system follow proper advertising strategies but may use unethical fraudulent actions or approaches for different purposes. For example, some advertisers may deplete rivals marketing budgets by committing fraud. Likewise, some publishers stealthily make various traps to lure users to view or click Ads, in which users are not really interested. Under such circumstances, both malicious advertisers and publishers are fraudsters. Aiming to target different pricing models, i.e. CPM, CPC and CPA, fraudsters constantly trying to inject fraud into the Ad system, where the majority fraud is impression-based, click-based, or action-based. In this chapter, we specifically describe Ad fraud following a set of W3H questions: Who does What to Whom, and How.

X. Zhu et al., *Fraud Prevention in Online Digital Advertising*, SpringerBriefs in Computer Science, DOI 10.1007/978-3-319-56793-8_3

"Who" answers whether a fraud is directly generated from human users or non-human. In certain under developed countries/regions, the cost of hiring human labors is relatively low. So fraudsters may recruit and paid human users to view and click advertisements, and produce conversions. on the other hand, a botnet [3, 7], one of most pervasive implementation for non-human fraud, may consist of thousands of bots or malware injected to infected computers, and can be manipulated by Command and Control center (C&C). They can automatically issue HTTP requests and even mimic a human users' behavior.

"What" intends to categorize the fraud and answers whether a fraud is impression-based, click-based, or action-based. Apparently, each fraud is targeting respective price model, such as impression-based fraud will try to increase the number of views so fraudsters can claim revenue from CPM campaigns. An accurate categorization of the fraud is the most fundamental step for developing solutions to prevent fraud in Ad systems.

"Whom" answers whom are the platform or devices a fraud is targeting: PCs, tablets, or mobile users. A study from a mobile Ad company AppLift and a mobile security company Forensiq indicated that "a surprising proportion of programmatic mobile Ad impressions – 34 percent to be exact–is at risk of being fraudulent" [11].

"How" resolves technical questions on how does a fraudster deliberate the fraudulent actions for certain types of fraud, and how to prevent such fraud in Ad systems.

Based on the above specifications, we categorize Ad fraud into a taxonomy with three major types, as shown in Fig. 3.1, where the categorization is mainly based on whether fraud is used to target Ad placements, Ad traffic, or Ad user actions.

The first type of fraud is placement fraud whose main objective is to manipulate/modify publisher websites or content showing on users' devices in order to increase the number of impression or clicks. Many mobile applications are attacked by this type of fraud.

The second type is traffic fraud which intends to generate fake traffic to inflate the number of impressions or clicks generated from individual sites or placements. For example, by using botnet or crowd, fraudsters can increase the number of impression and clicks on publishers' websites. To avoid passive traffic fraud, Google AdMob's [5] terms dictate that "Ads should not be placed very close to or underneath buttons or any other object which users may accidentally click while interacting with your application" and "Ads should not be placed in areas where users will randomly click or place their fingers on the screen".

The third type is action fraud which targets users' actions in order to generate revenue. For example, fraudsters may hire people to download or submit forms to produce conversions, or make fake cookies to earn commissions as affiliates by using bots. For re-targeting fraud, fraudsters can use bots to intimate users behavior to pretend being potential customers for valuable traffic on advertisers websites.

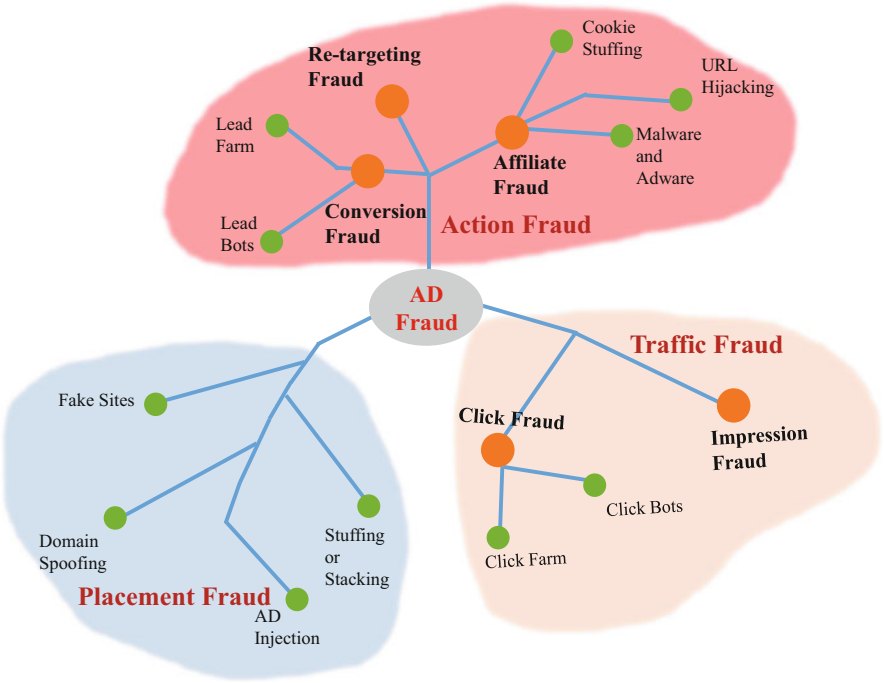

Fig. 3.1 The Ad fraud map. We categorize Ad fraud into three major categories: placement fraud, traffic fraud, and action fraud, depending on whether the fraud is targeting the page content, Ad traffic, or user actions, respectively. For each category, we further categorize fraud into multiple subgroups, with fraud inside each subgroup sharing similar objective and behaviors

3.2 Ad Fraud Prevention Mechanism

In order to fight Ad fraud, existing solutions commonly rely on the following four types of mechanisms [10]

- **Signature-based Prevention Mechanism:** This type of method uses predefined features/patterns to find malicious impression or traffic [4]. For example, research has found that if a client-side code execution is inconsistent with known code execution models (such as JavaScript), it is very likely that the traffic is not generated by real human users but by a bot [9]. Therefore, by testing code execution environment, such as JavaScript support or mouse event test [12], it is possible to filter out a significant portion of fraudulent traffic. Such behavior modeling methods have been extensively studied for Clickbots [8].
- **Anomaly-based Prevention Mechanism:** This type of approach uses statistical analysis and historical data to find suspicious placements, web sites, or publishers, whose traffics are considered abnormal compared to generic user traffic. For example, as of April 2016, the average probability of click events in displaying

advertisement is roughly 0.17% which means that, on average, there are about 1.7 click events on every 1000 impressions [2]. A placement or publisher website showing significantly higher click through rates will be deemed as abnormality and implies fraudulent activities deserving further investigation [13].

- **Honeypot-based Prevention Mechanism:** In order to detect fraudulent activities, Ad servers (such as advertisers) can intentionally serve a number of carefully defined bluff Ads to publishers, where the bluff/honeypot Ads are known to be unrecognizable (e.g. the size is too small or transparent) by human users, and if bluff Ads result in interactions, such as a click event, it will contradict to the assumption and therefore imply fraud activities [6]. Such a honeypot approach has been applied by Traffic Trafficker to examine traffic for better Ad serving [1].
- **Credential-based Prevention Mechanism:** The credential or creditability of publishers or websites is directly correlated to potential fraud activities. In order to assess the credential of publishers, DSPs or advertisers can use reverse-crawling to find the content of the web pages and check whether its content is consistent with the tags associated with the impression when submitting for auctions. In addition, one can also use the number of impressions generated from a publisher, and compare the value with trusted website rankings such as Alexa or RageRank. A publisher whose impression is much more than its traffic ranking would clearly imply potential fraudulent activities.

References

1. Blog TT (2015) Using honeypot banners to detect click fraud. http://www.traffictraffickers.com/blog/index.php/2015/07/
2. Chaffey D (April 26, 2016) Us, Europe and worldwide display ad clickthrough rates statistics summary. http://www.smartinsights.com/internet-advertising/internet-advertising-analytics/display-advertising-clickthrough-rates/
3. Cho CY, Caballero J, Grier C, Paxson V, Song D (2010) Insights from the inside: A view of botnet management from infiltration. LEET 10:1–1
4. Dave V, Guha S, Zhang Y (2012) Measuring and fingerprinting click-spam in ad networks. In: ACM SIGCOMM Computer Communication Review - Special October issue SIGCOMM '12, pp 175–186
5. Google (2015) Google admob behavioral policies
6. Haddadi H (2010) Fighting online click-fraud using bluff ads. ACM SIGCOMM Computer Communication Review 40:22–25
7. Miller B, Pearce P, Grier C, Kreibich C, Paxson V (2011) What's clicking what? techniques and innovations of today's clickbots. In: Detection of Intrusions and Malware, and Vulnerability Assessment, Springer, pp 164–183
8. Miller B, Pearce P, Grier C, Kreibich C, Paxson V (2011) What's clicking what? techniques and innovations of today's clickbots. In: Detection of Intrusions and Malware, and Vulnerability Assessment, pp 164–183
9. Neal A, Kouwenhoven S (2015) Quantifying online advertising fraud: Ad-click bots vs humans. In: Technical Report, Oxford Bio Chronometrics
10. Stone-Gross B, Stevens R, Zarras A, Kemmerer R, Kruegel C, Vigna G (2011) Understanding fraudulent activities in online ad exchanges. In: Proceeding IMC '11 Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference, pp 279–294

11. Sullivan M (2015) 33% of all programmatic ad impressions on mobile could be fake. http://venturebeat.com/2015/12/10/33-of-all-programmatic-ad-impressions-on-mobile-could-be-fake/, accessed December 10, 2015
12. Xu H, Liu D, Koehl A, Wang H, Stavrou A (2014) Click fraud detection on the advertiser side. In: Prof. of European Symposium on Research in Computer Security, pp 419–438
13. Yu F, Xie Y, Ke Q (2010) Sbotminer: Large scale search bot detection. In: Prof. of the Third ACM International Conference on Web Search and Data Mining

Chapter 4
Ad Fraud Categorization and Detection Methods

Abstract This chapter provides a comprehensive review of Ad fraud in three major categories: placement fraud, traffic fraud, and action fraud, which are at different levels of online advertising. Placement fraud mainly focuses on the pages which displaying the Ads. For placement oriented fraudulent activities, they often modify publisher pages or the web pages showing on the users' devices to increase impressions or clicks. Traffic fraud mainly tries to manipulate the network traffic to inflate the number of impressions generated from individual sites or placements. Action fraud targets users' meaningful business actions, such as filling an online form or survey, completing an online purchase order, or use users' previous actions or behaviors to re-target valuable customers. For each type of fraud, we will also review detection methods and approaches for online Ad fraud prevention.

4.1 Placement Fraud

An advertising placement is often an iframe containing Ads which are creative content with texts, pictures or videos. The placement can be set anywhere, i.e. left/right side, top/bottom side or even mixed with content. Floating or fixed position are also two options for the placement. Google AdSense suggests that for Ad placement [11], advertisers should consider to: (1) Put yourself in users' position and make the site easy to navigate. (2) Put the Ads close to the content that users are interested in. (3) Do not misguide users and keep Ads looks like Ads. (4) Do not put too many Ad placement in one page.

Placement fraud is defined as fraudulent actions or activities which intend to manipulate or modify publishers' web pages or modify the web pages showing on the user's devices to increase impressions or clicks. Such fraudulent activities involve a variety of actions, from simple keywords stuffing [5], misrepresenting Ad placement location so a placement is placed in the invisible frames and is never viewable to the audience [8] to Malvertising [30] which injects advertising malware by luring users to register and then redirecting traffic to malicious sites, in order to generate inflated impressions.

In the following, we categorize placement fraud into four groups, with each group focusing on one aspect of fraudulent actions, and review solutions to detect each type of placement fraud.

X. Zhu et al., *Fraud Prevention in Online Digital Advertising*, SpringerBriefs in Computer Science, DOI 10.1007/978-3-319-56793-8_4

```
(a) <iframe src =http://…domain.com width=1 height=1 frameborder=0 > </iframe>

(b) <iframe src =http://…domain.com width=100 height=100 frameborder=0
    Scrolling=no style="display:none;"> </iframe>
```

Fig. 4.1 General examples of stuffing Ads. (**a**) Stuffing Ads in a small iframe (e.g. 1 × 1 pixel). (**b**) Although the size of iframe is sufficiently large, the *display* : *none* command takes precedence

4.1.1 Stuffing or Stacking

Stuffing (either keyword stuffing or pixel stuffing) is a way of displaying content incapable of being viewed by naked eyes. It is commonly used for both keyword stuffing [5] and placement stuffing. In Ad keyword stuffing, Ad keywords are hidden in the HTML tags that are not visibly displayed, or they are shown as the same color as the background, so they are incapable of being viewed by naked eyes. Although hidden keywords cannot be viewed by naked eyes, they are in fact visible to Ad network agents when they crawl the web page content in order to determine the relevant pages correlated to specific Ads. Such techniques, in fact, are commonly seen in search engine cloaking [38] or using search engine optimization to increase visibility [15].

Similar to keyword stuffing, placement stuffing stuffs a large number of placements, not intended to be viewed by naked eyes, in a web page. An example of stuffing Ads is shown in Fig. 4.1 where the placement has a reasonable size but the visibility is set as "none" therefore cannot be viewed. Similarly, stacking entails layering Ads on top of one other in the same Ad slot but only the Ad on the top layer is viewable. By doing so, fraudsters try to stuff or stack these invisible Ads as many as possible to inflate the number of impressions.

In order to detect stuffing or stacking fraud, DoubleVerify Inc. [20] proposes several ways to detect hidden or invisible advertisements. One method is to compare the advertisements with graphic images extracted from html codes with snapshot of this web page. Using image analysis technology [45], any advertisement with the image which is not found in snapshot will be marked invisible. Another method is based on the geometric analysis. In order to check whether the advertisement is viewable, the code snippet embedded into the page will calculate the location of the advertisement, location of the viewable areas of the browser, and size of the open browser window.

4.1.2 Fake Sites

Fake site fraud involves two distinct forms. One is to create sites with legitimate domain names but contain only Ad slots [3]. Then by joining in large Ad networks

or Ad exchanges, fraudsters can obtain considerable revenue from fake sites. For example, an investigation from online Ad exchange [34] reveals that many fake sites have Wordpress blog template style web pages with posts only by an "admin" user with no comments, but a large number of Ads (from several different Ad exchanges) embedded in the pages. In many cases, the content of such fake sites are either meaningless or containing content stolen from other websites.

Another way of the fake sites is to mislead visitors by copying the content from well known sites or registering a similar domain name. Such fake sites are also found to deliver Ad tags which automatically redirect users' web browsers to fake anti-virus campaigns [18], or scare and trick users to download malware (approaches, commonly seen in malvertising [30]).

There are two common methods for fake sites detection, i.e. lookup blacklist and distinguish fake sites using machine learning methods. Almost every browser toolbar maintains a blacklist to check whether the site the user is visiting is fake or not [44]. The sites on the blacklist can be from sites identified by toolbar itself or reported by users as well as from third party communities, e.g. Anti-Phishing Working Group.

By using data mining and machine learning approaches, Abbasi and Chen [1] proposed a classifier system to detect fake sites. The system is based on a large number of features (nearly 6000) and a support vector machine (SVM) classifier. The features include body text, HTML design, images, linkage, and URLs. Given a site, each page a in that site can be represented by two vectors (i.e. maximum similarity vector and average similarity vector) which are calculated between a and the labeled pages. Then the inner product of two vectors are the input for SVM classifier. Finally, the fake site can be determined by the number of classified pages as well as the number and percentage of fake pages.

4.1.3 Domain Spoofing

Web spoofing [6] is commonly known in the Internet where fraudsters create websites mimicking real sites in order to carry out fraud activities, such as stealing identity information or account credentials.

In Ad networks, most advertisers maintain a whitelist for reputed premium publishers and a blacklist for fraudulent publishers such as porn sites and fake sites with no credibility or low quality content. For brand safety and many other concerns, displaying advertisement on blacklisted sites is inherently prohibited. So blacklisted websites would be least preferred by the advertisers. Because advertisers are willing to place their Ads and even bid higher price on higher quality sites, fraudsters spoof their domains to avoid being placed on the blacklists. Accordingly, domain spoofing refers to fraudulent activities trying to falsify the domain as if the traffic are from publishers in the whitelist.

In Ad networks, domain spoofing is normally carried out through the following two major approaches:

- **Malware & Tool-bars:** If a user's computer were infected by malware or user installed malicious tool-bars, fraudsters can inject Ad windows onto web pages the user is viewing. This creates an impression, which appears to be on a premium publisher's site, because the user may actually be on the premium domain, whereas, in fact, the Ads actually originate from a toolbar. The fraudster can send the impression for bidding on Ad exchanges, with heavily discounted price for desirable sites, but the money generated from the Ad (which was originated from malware or tool-bars) is collected by the fraudster but not the premium publishers. Because users are genuinely on the premium sites when the Ad impression is sent out for auction, as a result, this type of domain spoofing is difficult to detect [14].
- **Ad Tag Misrepresentation:** By modifying Ad tags in the bid auction, fraudsters can also spoof their domains to pretend offering impression at premium web sites, but the Ad is in fact displayed on a blacklist site. For example, Rubicon (an SSP) often provides their publishers with a JavaScript snippet to put on their site to show Ads and monetize their content. When the script makes a call to Rubicon's servers, it passes some information about the page it's on to the server. A nefarious publisher can modify the script, or add an additional script to the page which rewrite some of the functionality of the script so that it passes false information. Without spoofing about their website contents, a fraudulent publisher would not receive higher value Ads on their pages than they would receive [34].

4.1.4 Ad Injection and Malware

Ad injection and malware are more aggressive fraud activities which directly affect client web browsers to either modify the advertisement [35] or display advertisement on the current web page which does not have any Ad placement at all. An Ad injection example is that in 2014, a Target Ad popped up on the Walmart.com Web site, which Walmart did not sell the placement to Target but was caused by fraud Ad injection advertising. In addition, Internet service providers, such WiFi service providers may also tamper with in-transit HTTP content to inject Ads [16]. Another type of Ad injection is brought by advertising software, such as malicious adware, a program running on client computers to display unintended advertisement. For example, adware has known to target the Ad businesses of Web giants including Facebook Inc., Google Inc. and Yahoo Inc., by inserting a layer of Ads on web browser's current websites or covering up other paying Ads [32].

Formally, Google [35] defines Ad injection as any binary, extension, or network ISP that modifies page content to insert or replace advertisements, irrespective of user consent. In Ad injection, Ads are injected into websites which users

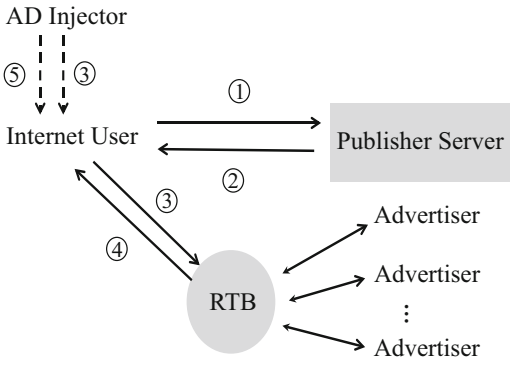

Fig. 4.2 Flows of Ad injection. ① A user sends a request for the web page from a premium publisher. ② The web page is sent back by the publisher content server. ③ The Ad injector (malware) installed on user's computer acts like premium publisher who own the web page to bid on the real time bidding platform. ④ The advertiser offers a high value for the Ad injector's Ad slot. ⑤ Ad injector modifiers the web page on user's computer to insert or replace the Ad

are viewing, through malware installed on users' computers. Alternatively, Ad injection can also be done through browser extension software which is often used to increase/enhance the functionality of the web browser [41]. For example, compromised browser extension programs may be uploaded to Chrome web store or Mozilla AdOns store. Because browser extensions are typically written using HTML, JavaScript, and CSS, they will interact with the web page currently loaded in the browser, allowing JavaScript or other script codes to modify DOM structures of the web page, and also leveraging with browser APIs to communicate with external servers. As a result, a web browser will display injected advertisement which is never meant to be part of the current web page. In Fig. 4.2, we list the major steps of Ad injection explaining how Ad injection is carried out in online digital advertising.

Tampering with existing elements and inserting rogue elements in clients' HTML pages are two tricks of Ad injection. To get Ad injection scripts from HTML pages, Thomas et al. [35] developed a detection method based on client side DOM with Google websites. They first scan the clients' DOM to identify suspicious elements and fraudulent domains. Then by excluding normal programs such as browser toolbars and antivirus engines, they filter scripts which are not caused by Ad injection. Finally, they manually review the scripts based on the scripts' content.

4.2 Traffic Fraud

Increasing traffic is a natural way of increasing revenue, particularly for publishers. Therefore, traffic fraud mainly aims to inflate the number of impressions generated from individual sites or placements, by manipulating the network traffic. In addition,

for CPC based campaigns, only the click action of the users on the displayed advertisement can result in a revenue, so click fraud [39, 43] is also commonly observed, and is one of the most common fraudulent activities.

4.2.1 Impression Fraud

Impression fraud aims to directly increase the website traffic and subsequently generate more impressions for auction. This type of fraud has the most significant impact to the CPM based campaigns, because inflated impressions provide little or no value to advertisers for their advertising benefits [5]. In addition, it also impacts on CPC and CPA based campaigns because most impression fraud cannot result in click or conversion events, and therefore will lower the click-through-rate (CTR), because the denominator of the CTR calculation is the number of page views.

Indeed, other type of fraud, such as placement fraud etc., may also result in increased impressions and therefore share similar objective. In reality, impression fraud is generated from three common approaches: hiring human labor to manually view pages, design different types of bots to generate impressions for auctions, and using expired domains to divert users to third-party pages [31]. For example, by using a Bot to repetitively send HTTP request to a web server, it will trigger a large number of page display request, and therefore results in inflated impression which has zero business value to the advertiser. Alternatively, one could use genuine human labor to manually review/refresh websites in order to trigger Ad auction from the sites, or direct traffic from some expired sites to third party pages, and subsequently increase the impression volumes.

Because hiring human labor is considered too expensive for generating web traffic, whereas bot often has less intelligent power to mimic human traffic, some hybrid approaches intend to increase website traffic by combining genuine human actions and automated bot functions. For example, in pay-per-view (PPV) networks [31], publishers gain impressions in an invisible way. Whenever a user views one publisher of this network and clicks anywhere in that site, it will trigger an invisible frame containing other publishers. The authors introduced three counter measures, i.e. filtering zero-sized viewports, blocking traffic from PPV networks using referral blacklists, and stopping running advertisement on publishers in blacklist.

4.2.2 Click Fraud

A click event, on an advertisement, is a clear signal indicating that a viewer is potentially interested in an advertisement, and therefore may become a purchasing customer. Therefore, click through rate (CTR) is often used to assess the effectiveness at different levels, such as at the placement level, site level, or publisher level

etc. Arguably, Click fraud is the most popular fraud in the Ad ecosystem, mainly because that CPC based campaigns dominate Ad networks. In a click fraud attack, fraudsters use different types of approaches, either manually or using bots, to click on an advertisement. Because fake clicks cannot converge to meaningful business actions, this type of fraud results in direct advertising loss to the branding company.

A click fraud may be rooted from two parties, publishers and advertisers, with two motivations, publisher click inflation or advertiser competition, respectively [37].

- **Publisher Click Inflation:** From publisher perspective, click events can bring immediate revenue, because publishers are rewarded based on the percentage of advertised impressions clicked by viewers. Therefore, publishers are intuitively tolerating click fraud attacks, if not encouraging or participating such activities. To do so, clicks are produced either by using automated programs or using human labors.
- **Advertiser Competition Clicks:** Most, if not all, advertising campaign have a budget, which supports the advertising activities for a certain period of time, say one or multiple months. Under the CPC revenue model, each click will consume a small amount of advertising budget. Therefore, by producing artificial clicks on competitor's advertisements, the competitor's advertising budget may be exhausted within a short time period. As a result, fraudulent advertiser's Ads would have the advantage of being served to legitimate users with a better chance of being clicked by users and resulting in a better conversion rate, and satisfying the branding company.

  To defend click attacks, most advertisers employ a pacing rate control, which specifies daily or hourly advertising spending cup for smooth budget delivery [17]. This can avoid whole campaign budget being exhausted within a short amount of time, whereas the low quality clicks with no advertising value still add extra burden to any campaigns.

To generate fake clicks, two commonly used approaches are click farms or click bots [26], where the former is generated by human viewers and the later is generated by computer programs.

- **Click Farm:** A click farm consists of a large number of hired human labor who manually click on the advertisement [12]. Although this type of click is genuinely produced by human viewers, they have very little, or no intention, to be converted to purchasing customers, and therefore their clicks are generated with malicious or fraudulent intent.
- **Click Bots:** A click bot refers to an automated computer program/system (either stand-alone or distributed as a bot net) which automatically and repeatedly retrieves URLs associated with Ads to generate mouse click events, resemble to genuine human viewers [4]. In a click fraud bot net, such as ZeroAccess [40], a bot infected host may be coordinated/controlled by a master bot to fetch on-line advertisements and click on the Ads without host user's awareness.

In order to detect click fraud which are either generated by intentionally manipulated fraud activities or by ClickBots [24], common approaches are to use signature type of mechanisms to examine whether clicks follow specific pattern. For example, one can check whether two or more clicks are always generated from the same client machine within a short period of time, or whether there are periodicity or correlations between clicks and other items. For example, research [42] has proposed to use simple window-based approaches to detect duplicate clicks (or click frauds) within a short time window frame. Others [21] proposed to use association rule for fraud detection in web advertising networks, where publishers may construct their web pages to automatically click the advertisements, whenever the page is loaded/displayed on customers' browsers, which is considered inflation attack (or click fraud).

Stitelman et al. [33] proposed a two stage step to find non-intentional traffic (NIT) which are not generated by genuine users' interest. In the first stage, they build a bi-partite graph $G =< B, W, E >$, in which B denotes browsers, W represents websites and E are edges denoting that B visited the W. Then this bi-partite graph is projected to co-visiting network

$$G_w^n =< V_W \subseteq W, E = (x, y) : x, y \in W, [\Gamma_G(x) \bigcap \Gamma_G(y)]/\Gamma_G(x) \geq n > \quad (4.1)$$

where $\Gamma_G(x)$ is the set of neighbors of x. If the number of first-degree neighbors of the website is sufficiently large, this website is considered as having high level of non-intentional traffic.

The second stage is for tracking browsers, where the browsers will be placed in a "penalty box" if they visit websites which have high level of non-intentional traffic in the recent past. Any traffic generated by browsers in the "penalty box" will be recognized as non-intentional traffic. But penalized browsers will be released if they don't visit those sites for a period of time.

Tian et al. [36] proposed a detection method against crowd fraud as well as hiring labors fraud for search engine advertising. Three general characteristics are observed, i.e. moderateness for moderate hit frequencies, synchronicity for attacking a common set of advertisers and dispersivity for searching unrelated queries. This method consists of three stages based on three general characteristics, respectively. The constructing stage will eliminate extremely small or large queries and build surfer-advertiser bipartite graph. Then in the clustering stage, synchronization similarity is defined to count the number of common advertisers between each pair of two click histories in a time window. After that, a nonparametric clustering method based on DP-means is provided to detect suspicious crowd fraud clusters. The final stage is to filter normal clusters with less disparity based on a domain coherence coefficient (DCC).

4.3 Action Fraud

Action fraud intends to target users' meaningful business actions, such as filling an online form or survey, placing an online purchase order, or use users' previous actions or behaviors to re-target valuable customers [13]. Because advertisers are strongly interested in using cost-per-action (CPA) to assess their advertising costs *vs.* revenue, action fraud has direct impact to the Ad pricing, campaign planning, and many other major components of the Ad ecosystem.

4.3.1 Conversion Fraud

A conversion in Ad network denotes one or a set of meaningful business actions taken by the site visitors which they are converting to paying (or potential paying) customers. Alternatively, a conversion can also be defined as "agreed-upon action taken by a user" [23]. For example, a simple conversion event can be a downloading of a file or filling out a form, or a completion of an online purchase order. Such fraud is also called conversion spam [5].

As shown in Fig. 4.3, a conversion normally requires a number of actions from users, and a conversion event typically takes place minutes, hours, or even days after the original Ad click. It is worth noting that a conversion is typically tracked on the branding sites (or the landing page) through the placed pixel, whereas a click is typically tracked on the publishers' sites. The mapping (i.e. identifying users whose click resulting in a conversion) is often done by matching user cookie information.

After users clicking on the displayed advertisement, they are normally directed to a landing page which shows summarized information about the advertised product or services. The purpose of the landing page is to either (1) persuade users to click through another pages towards making a purchasing decision (click through landing pages); or (2) collect user information or request actions, with necessary description of what users will get, in return for submitting their personal data or actions (lead generation landing pages). For most click through landing pages based conversions, a purchase is required so users will need to provide name, credit card, and other important information. Because this process requires sophisticated interactions and financial commitment, very few fraudulent activities exist to target this type of conversion.

For lead generation landing page based conversions, users are only required to provide simple information or taking simple actions, such as filling in user name, household address, or downloading a file (such as a trial version of a software package) from the advertisers' site. All these actions can be carried out with minimum, or next to nothing, financial costs. Therefore, majority conversion fraud targets this type of conversion.

Similar to the click fraud, conversion fraud is often committed through two types of activities [27]:

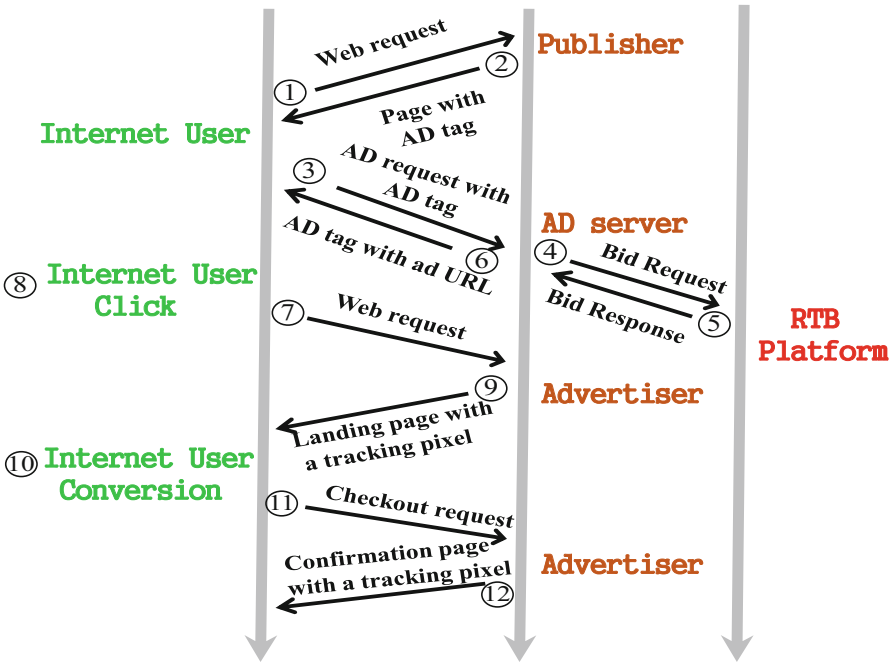

Fig. 4.3 A typical Ad click example from AppNexus advertising platform. Notice that impressions are tracked by the Ad server while clicks and conversions are tracked by the pixel on the landing page and confirmation page, respectively

- **Lead Bots:** A lead bot is a computer agent which automatically fills out lead forms with either randomly generated, or partial correct information. In addition to filling and submitting user forms, a lead bot can also commit simple actions, such as clicking a link to download a file etc.
- **Lead Farm:** Because not all conversions can be processed by bots, and sites with a higher conversion rate are mostly favored by the advertisers, fraudsters are willing to hire people to produce conversions from under-developed countries with lower labor costs. This results in conversion fraud activities using lead farm, which consists of genuine human labors to fill lead forms or commit other required activities, in order to convert a click to a conversion.

4.3.2 Re-targeting Fraud

Re-targeting (also called re-marketing) is a very effective form of online advertising, which intends to accurately target valuable customers based on their previous Internet actions, such as purchasing history the customers have made before or web browsing history/behaviors of the customers [19]. This can be done by checking

past transaction records, or tracking cookies from users visiting sites before and showing interest in certain products, to determine whether a user is interested in certain product or not.

In order to re-target customers, advertisers commonly use techniques, such as "cookie" or "pixel" as the snippet code. If an audience has visited a website, opened any email messages, or browsed any other pages on which the snippet code have been previously inserted, the advertiser can then use the cookie information to understand user behaviors and serve most relevant advertisement to them [10]. In the AdWords, re-targeting is called "remarketing", referring to the ability of serving Ads through the Google display network to anyone who lands on a page that has an audience tag. In Bing, similar techniques are called remessaging [10].

For re-targeting fraud, the main objective of the fraudsters is to mimic genuine customers' specific behaviors and make them behave like desirable users. This is commonly achieved by using computer generated agents, such as DeceptiBots [25], to mimic a human's intentions and behaviors and pretend that they are interested in a specific product or a brand. As a result, the bots deceive advertisers into believing that bots are valuable potential customers, and therefore put a higher price on auctions/impressions generated by the bots.

4.3.3 Affiliate Fraud

Affiliate marketing is a type of performance based marketing approaches, where an affiliate (i.e. a business entity) will receive rewards for each visitor or customer brought by their marketing efforts. In affiliate marketing, affiliates use different types of advertising methods, including search engine optimization (SEO), e-mail marketing, or display advertising to attract visitors. Affiliate fraud is referred to activities which deceive the system for claiming commission/revenue which the affiliate does not qualify.

For most affiliate marketing, commission is paid only if a user makes a purchase, so affiliate can only claim commission after a conversion occurs. As a result, affiliate fraud mainly focuses on finding users who are already on the verge of making purchases [9].

An affiliate fraud [29] is commonly carried out through the following three types of approaches:

- **Malware and Adware:** When audiences are navigating to a branding company's site without affiliate's assistance/referral, the affiliate should not be qualified to claim commission. However, if the audiences' computers are infected by affiliates' malware or adware, which see that the audience is visiting a sponsor's website, the adware will then redirect the user again through an affiliates' marketing link, acting like that the user were being referred by the affiliate. If the user happens to subsequently make a purchase, the affiliate will be credited as the putative cause of that purchase, and therefore render an affiliate fraud [7].

- **Cookie Stuffing:** In cookie stuffing based affiliate fraud, an affiliate will try to attract audience to visit a website and then stuff cookies to the audience's computer. Later on, if the audience makes any purchase from the advertiser's site, the affiliate can claim commission. This is normally done by designing a web page to attract audience potentially interested in a certain brand or product, e.g., by repeatedly and falsely promising coupons or discount of a certain brand/product. Once the audience visits the affiliates' web site, the affiliate will stuff a large number of cookies on his/her computers (web browsers), which typically remain alive for 7–30 days. If the viewer happens to make purchase on advertisers' sites, before the cookie expires, the affiliate will claim commission for referring a customer to the advertisers [2].
- **URL Hijacking:** An URL hijacking is also called typosquatting [22], which happens by mistakes such as typographical errors made by audience when incorrectly inputting a website address into a web browser (e.g. typing "Walmat.com" instead of "Walmart.com"). Once a typosquatting occurs, the site hijacks the URL users typed in the web browser, and directs the traffic to specific sites and therefore claims commission. For example, a research study carried out in 2010 [22] has shown that about one million (938,000) typosquatting domains target the top 3264.com sites, and about 80% of such typosquatting domains are supported by pay-per-click Ads, either by advertising the correctly spelled domains or by its competitors.

Shekhter [28] proposed a framework to detect fraudulent affiliates by using three steps: The first step is to separate and group traffic based on affiliate ID. After that, they use algorithms to find suspicious traffic which coincide with fraudulent activity in each group/affiliate. Finally, if the number of suspicious traffic is beyond a predefined percentage in each group, that affiliate is determined to be fraudulent.

References

1. Abbasi A, Chen H (2009) A comparison of tools for detecting fake websites. Computer (10):78–86
2. Chachra N, Savage S, Voelker GM (2015) Affiliate crookies: Characterizing affiliate marketing abuse. In: Proceedings of the 2015 Internet Measurement Conference (IMC)
3. Conti M, Cozza V, Petrocchi M, Spognardi A (2015) Trap: using targeted ads to unveil google personal profiles. In: Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS), IEEE
4. Daswani N, Stoppelman M (2007) The anatomy of clickbot.a. In: Proceedings of the Fist Workshop on Hot Topics in Understanding Botnets (HotBots'07), USENIX
5. Daswani N, Mysen C, Rao V, Weis S, Gharachorloo K, Ghosemajumder S, the Google Ad Traffic Quality Team (2015) Online advertising fraud. Crimeware: understanding new attacks and defenses
6. Dinev T (2006) Why spoofing is serious internet fraud. Communication of the ACM pp 77–82
7. Edelman B (2005) How affiliate programs fund spyware. http://www.benedelman.org/news/091405-1.html

8. Edelman B (2014) Accountable? the problems and solutions of online ad optimization. IEEE Security & Privacy 12(6)
9. Edelman B, Brandi W (2015) Risk, information, and incentives in online affiliate marketing. Journal of Marketing Research 52:1–12
10. Ellis S (2012) The future of retargeting, remarketing and remessaging. http://marketingland.com/the-future-of-retargeting-remarketing-and-remessaging-7643, marketing Land
11. Google (2016) Best practices for ad placement. https://support.google.com/adsense/answer/1282097
12. Google (2016) Google ads: Ad traffic quality resource center. http://www.google.com/ads/adtrafficquality/, google
13. Hoofnagle CJ, Soltani A, Good N, Wambach DJ, Ayenson MD (2012) Behavioral advertising: The offer you cannot refuse. Harvard Law & Policy Review 6:273–296
14. Hubspotcom (2015–2016) Advertising: digital advertising fraud. In: Innovation in Magazine Media
15. Killoran JB (2013) How to use search engine optimization techniques to increase website visibility. IEEE Transactions on Professional Communication 56(1):50–66
16. Kravets D (2014) Comcast wi-fi serving self-promotional ads via javascript injection. ttp://arstechnica.com/tech-policy/2014/09/why-comcasts-javascript-ad-injections-threaten-security-net-neutrality/
17. Lee KC, Jalali A, Dasdan A (2013) Real time bid optimization with smooth budget delivery in online advertising. In: Proceedings of the Seventh International Workshop on Data Mining for Online Advertising (ADKDD '13), ACM
18. Li Z, Zhang K, Xie Y, Yu F, Wang X (2012) Knowing your enemy: Understanding and detecting malicious web advertising. In: Proceeding CCS '11 Proceedings of the 18th ACM conference on Computer and communications security, ACM, pp 674–686
19. Liu B, Shethz A, Weinsbergz U, Chandrashekarz J, Govindan R (2013) Adreveal: Improving transparency and control in online targeted advertising. In: Proceedings of the Twelfth ACM Workshop on Hot Topics in Networks
20. Mclaughlin M, Rosenfeld RK, Abu LM, Simon L (2015) System and method for identifying hidden content
21. Metwally A, Agrawal D, Abbadi AE (2005) Using association rules for fraud detection in web advertising networks. In: Proceedings of the 31st VLDB Conference
22. Moore T, Edelman B (2010) Measuring the perpetrators and funders of typosquatting. In: Financial Cryptography and Data Security: Lecture Notes in Computer Science
23. Mungamuru B, StephenWeis (2008) Competition and fraud in online advertising markets. In: Financial Cryptography
24. Neal A, Kouwenhoven S (2015) Quantifying online advertising fraud: Ad-click bots vs humans. In: Technical Report, Oxford Bio Chronometrics
25. Nowak P (2012) Deceptibots: when machines go bad. New Scientist 214:45–47
26. Pearce P, Dave V, Grier C, Levchenko K, Guha S, McCoy D, Paxson V, Savage S, Voelker GM (2014) Characterizing large-scale click fraud in zeroaccess. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, pp 141–152
27. Performics (2014) Digital advertising fraud & abuse: Strategies & recommendations for mitigation. http://blog.performics.com/digital-advertising-fraud-abuse-strategies-recommendations-mitigation/, performics
28. Shekhter H (2011) System and method for detecting fraudulent affiliate marketing in an online environment. US Patent App. 13/082,554
29. Snyder P, Kanich C (2015) No please, after you: Detecting fraud in affiliate marketing networks. In: Proceedings of the Workshop on the Economics of Information Security (WEIS)
30. Sood AK, Enbody RJ (2011) Malvertising-exploiting web advertising. Computer Fraud and Security
31. Springborn K, Barford P (2013) Impression fraud in on-line advertising via pay-per-view networks. In: USENIX Security, pp 211–226

32. Steel E (2011) New 'adware' apps bug facebook, google. http://www.wsj.com/articles/SB10001424052970203413304577086463731021828, the Wall Street Journal
33. Stitelman O, Perlich C, Dalessandro B, Hook R, Raeder T, Provost F (2013) Using co-visitation networks for detecting large scale online display advertising exchange fraud. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, pp 1240–1248
34. Stone-Gross B, Stevens R, Zarras A, Kemmerer R, Kruegel C, Vigna G (2011) Understanding fraudulent activities in online ad exchanges. In: Proceeding IMC '11 Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference, pp 279–294
35. Thomas K, Bursztein E, Grier C, Ho G, Jagpal N, Kapravelos A, McCoy D, Nappa A, Paxson V, Pearce P, et al (2015) Ad injection at scale: Assessing deceptive advertisement modifications. In: 2015 IEEE Symposium on Security and Privacy (SP), pp 151–167
36. Tian T, Zhu J, Xia F, Zhuang X, Zhang T (2015) Crowd fraud detection in internet advertising. In: Proceedings of the 24th International Conference on World Wide Web, pp 1100–1110
37. Vratonjic N, Manshaei MH, Hubaux JP (2012) Online advertising fraud. In: Death Of the Internet, Markus Jakobsson edit.
38. Wang DY, Savage S, Voelker GM (2011) Cloak and dagger: dynamics of web search cloaking. In: Proceeding CCS '11 Proceedings of the 18th ACM conference on Computer and communications security, ACM, pp 477–490
39. Wilbur KC (2008) Click fraud. Marketing Science 28:293–308
40. Wyke J (2012) The zeroaccess botnet: Mining and fraud for massive financial gain. In: Technical Report, Sophos
41. Xing X, Meng W, Lee B, Weinsberg U, Sheth A, Perdisci R, Lee W (2015) Understanding malvertising through ad-injecting browser extensions. In: Proceeding of the 24th International World Wide Web Conference, ACM
42. Zhang L, Guan Y (2008) Detecting click fraud in pay-per-click streams of online advertising networks. In: Proceedings of the 28th International Conference on Distributed Computing Systems
43. Zhang Q, Ristenpart T, Savage S, Voelker GM (2011) Got traffic?: an evaluation of click traffic providers. In: Proceedings of the 2011 Joint WICOW/AIRWeb Workshop on Web Quality, ACM, pp 19–26
44. Zhang Y, Egelman S, Cranor L, Hong J (2006) Phinding phish: Evaluating anti-phishing tools. In: Proceedings of the 14th annual network and distributed system security symposium
45. Zheng Y, Jeon B, Xu D, Wu Q, Zhang H (2015) Image segmentation by generalized hierarchical fuzzy c-means algorithm. Journal of Intelligent & Fuzzy Systems 28(2):961–973

Chapter 5
Ad Fraud Measure and Benchmark

Abstract In this chapter, we discuss measures and benchmark datasets commonly used for Ad fraud detection. The measures include fraud detection accuracy, precision, recall, F-measure, and AUC scores which are commonly used to validate the performance of classifiers for classification. In addition, we also summarize several real-world datasets which are currently available for Ad detection and computational advertising research in general.

Correct measure and representative benchmark data are two critical components to validate the effectiveness of any fraud detection algorithms or systems. On one hand, real-world Ad fraud data are hard to obtain, partially because verifying whether an impression is genuinely a fraud or not is time consuming and requires special skills, not to mention the privacy issues which often forbid any company or third party to publish data online. On the other hand, compared to normal traffic and impressions, fraud is only a small percentage. So fraud detection systems and algorithms must be able to detect small probability events effectively. Accordingly, this chapter review measure and benchmark data available for Ad fraud prevention research and development.

5.1 Measures with Ground Truth

When labels indicating fraud are provided, the most common measures for evaluating algorithms or models are Accuracy, Precision, Recall, and F-measure, etc.

Table 5.1 shows the confusion matrix about a two class classifier for fraud detection. Many measures for fraud detection evaluation can be derived from a confusion matrix.

- Accuracy is the ratio between fraud that are correctly classified and the total number of samples, as calculated using the equation:

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{5.1}$$

X. Zhu et al., *Fraud Prevention in Online Digital Advertising*, SpringerBriefs in Computer Science, DOI 10.1007/978-3-319-56793-8_5

Table 5.1 A Confusion Matrix

Detected			
		Positive(Fraud)	Negative(Non Fraud)
True	Positive(Fraud)	TP : $TruePositive$	FN : $FalseNegative$
	Negative(Non Fraud)	FP : $FalsePositive$	TN : $TrueNegative$

Each row denotes the true label and each column denotes the detected label. In a 2×2 confusion matrix, the true labels are either positive (fraud) or negative (non fraud), and the detected labels are also either positive (fraud) or negative (no fraud). True Positive (TP) denotes the number of samples which are labeled as positive (fraud) and are also detected as positive. True Negative (TN) denotes the number of samples which are labeled as negative (non fraud) and are also detected as negative. False Positive (FP) denotes the number of samples which are labeled as negative (non fraud) but are falsely detected as positive. False Negative (FN) denotes the number of samples which are labeled as positive (fraud) but are falsely detected as negative

- Precision is the ratio between fraud that are correctly classified and the detected fraud, as calculated using the equation:

$$Precision = \frac{TP}{TP + FP} \tag{5.2}$$

- Recall or True Positive Rate (TPR) is the ratio between fraud that are correctly classified and the number of true fraud, as calculated using the equation:

$$Recall(TPR) = \frac{TP}{TP + FN} \tag{5.3}$$

- False Positive Rate (FPR) is the ratio between Non Fraud that are incorrectly classified as Fraud and the number of true non fraud, as calculated using the equation:

$$FPR = \frac{FP}{FP + TN} \tag{5.4}$$

- F-measure is the harmonic mean of Precision and Recall:

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{5.5}$$

In reality, because fraudulent instances, especially fraudulent publishers, are much less than normal instances [9]. If a dataset has 1% fraudulent impressions and the rest are normal, a classification model can classify all impressions as normal and result in 99% classification accuracy. However, this model is hardly useful because it leaves all fraud unidentified. Receiver Operating Characteristic (ROC) curve is more suitable for evaluating the performance of any models on such skewed sample distributions. For example, the ROC curves are used to assess the performance of click fraud detection classifiers [10] in a fraud detection in mobile advertising (FDMA) competition in 2012.

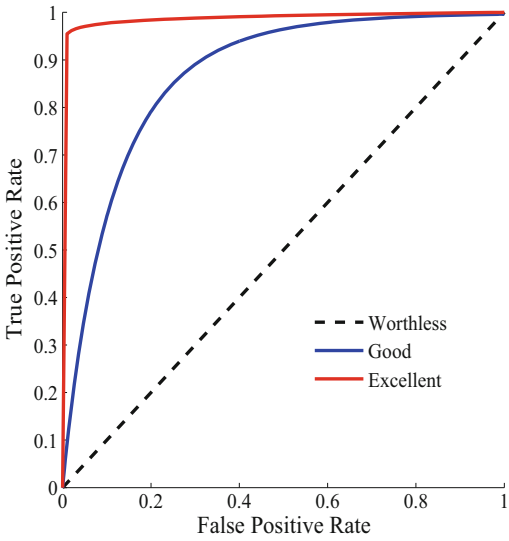

Fig. 5.1 Receiver operating characteristic (ROC) curve. The *x*-axis denotes false positive rate, the *y*-axis denotes true positive rate, and the *dash line* denotes random estimation. The better the performance of a prediction model, the closer its ROC curve is to the *upper left corner*. The areas under the ROC curve is called the AUC value, which is commonly used to assess a prediction model's performance on imbalanced sample distributions

ROC curve is plotted base on the true positive rate (TPR) and the false positive rate (FPR) for different cut points as shown in Fig. 5.1. The dashed diagonal line denotes the random estimation, where true positive rate and false positive rate are equal. The closer the ROC curve is to the upper left corner, the better the performance of the method is.

The area under the ROC curve, commonly referred to as the AUC value, is a measure to evaluate the performance of a model on severely imbalanced sample distributions.

In addition to the ROC curve and AUC value, Average Precision (AP) [10], a measure considering both Precision and Recall, has also be employed to evaluate models on imbalanced data distributions. Because Precision can be expressed a function of Recall [14], AP can also be described by Precision-Recall curve which is similar to the ROC curve shown in Fig. 5.2. The AP of A or B is the area under the respective curve. The larger the AP value, the better the performance of the method is.

5.2 Measures Without Ground Truth

If no ground truth is available, the alternative evaluation is to compare the results from a proposed model with results from existing fraud detection systems. From the advertisers' perspective, Dave et al. [5] created advertisers' websites and validate their Bayesian method for detecting clicking spam by signing up with ten major Ad netwoks, such as Google search, Bing search and Google Adsense. Tian et al. [12]

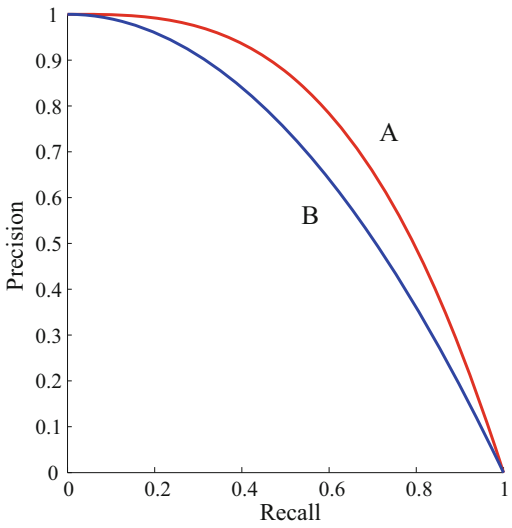

Fig. 5.2 Average Precision (AP). The *x*-axis denotes the recall and the *y*-axis denotes the precision. The better the performance of a prediction model, the closer its AP curve is to the *upper right corner*. The areas under the AP curve (or AP score) is equivalent to the AUC denoted in Fig. 5.1. The larger the AP score, the better the model is in classifying data with imbalanced sample distributions

first evaluated the results by using a rule based system which detects simple fraud as a baseline. Then given a large collection of logs, they manually distinguish fraud by sampling methods.

5.3 Real World Datasets

Indeed, real-world online advertising fraud datasets are rare. In the following, we list several useful benchmark data sources for fraud detection, and computational advertising research in general.

One useful dataset [10] previously used for fraud detection in mobile advertising competition was provided by BuzzCity Pte. Ltd. The data were first released in a Fraud Detection in Mobile Advertising (FDMA) 2012 Competition to improve the detection accuracy using data mining methods (the data are available at http://research.larc.smu.edu.sg/fdma2012). Two types of data are included in the dataset: publisher database and click database. In publisher database, profiles about publishers are listed such as the identifier of a publisher, publishers' bank accounts, addresses and the labels of publishers indicating whether the publisher is fraud, suspicious or not. There are many missing values about bank accounts and addresses. Meanwhile, the click database records specify information about click events such as identifiers of publishers, click time, device models, IP addresses etc.

Kaggle click-through rate prediction (CTR) is a series of data science challenges of CTR prediction for display advertising [8] and also for mobile users [7]. For the latter, the challenge provides 11 days worth of Avazu data to build and test click-through rate prediction models for mobile devices. The dataset contains

impression level samples with 24 data fields. Each instance denotes one impression, and the 24 data fields include information, such as impression ID, time-stamp, Ad banner position, site ID, site categorization, device type, etc. Because this is a CTR challenge, a clicked impression is explicitly marked (labeled). Therefore, one can use supervised learning to build classification models for CTR prediction. While the Kaggle challenge does provide a real-world testbed for CTR prediction, the percentage of impressions with clicks is over 20% of all impressions. This is considered much higher than the average industry standard, where only about 0.17% of impressions may result in a click (i.e. 1.7 clicks in 1000 impressions) [6]. The high CTR rate of the Kaggle is partially because that the dataset is collected from mobile users, which normally have a higher CTR values on average.

KDD Cup [11] is an annual data mining and knowledge discovery competition series organized by ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD). The KDD CUP 2012 challenge Track 2 [4] is a data science challenge for search advertising. Users are given the training instances derived from session logs of the Tencent proprietary search engine [1], and the challenge requires an accurate prediction of the probability of an Ad being clicked by users. The dataset includes a number of key fields for search advertising, including query terms, position of the Ad (i.e. the order of an Ad in the impression), and Ad properties, such as title and descriptions of the Ad.

Zhang et al. [13] released a dataset from a Chinese advertising company iPinYou which aims to improve the performance of DSP bidding algorithm. The dataset consists of detailed log information from four types of events, including bids, impression, clicks, and conversions. These logs are divided into three parts consisted of training and testing data from 3 months in different seasons. There data have 24 features and their semantic descriptions are available in [13].

The other two datasets are all from Criteo Labs INC. The first one [2] is for CTR prediction which contains traffic over a period of 24 days. There are 39 features, among which 13 features are integer and 26 features are categorical. The second dataset [3] is about conversion traffic over a period of 2 months with about 17 features. Unfortunately, no semantic information about these features from the two datasets are provided.

References

1. (2012) Tencent proprietary search engine. http://www.soso.com/
2. Criteo (2016) http://labs.criteo.com/downloads/download-terabyte-click-logs/
3. Criteo (2016) http://labs.criteo.com/downloads/2014-conversion-logs-dataset/
4. Cup K (2012) Predict the click-through rate of ads given the query and user information. http://www.kddcup2012.org/c/kddcup2012-track2
5. Dave V, Guha S, Zhang Y (2012) Measuring and fingerprinting click-spam in ad networks. In: ACM SIGCOMM Computer Communication Review - Special October issue SIGCOMM '12, pp 175–186
6. Exchange I (2016) http://www.indexexchange.com/

7. Kaggle (2015) Click-through rate prediction: Predict whether a mobile ad will be clicked. https://www.kaggle.com/c/avazu-ctr-prediction
8. Kaggle (2015) Display advertising challenge: Predict click-through rates on display ads. https://www.kaggle.com/c/criteo-display-ad-challenge
9. KS P, B N, MA F, Z A, WL W (2013) A novel ensemble learning-based approach for click fraud detection in mobile advertising. In: Prasath R., Kathirvalavakumar T. (eds) Mining Intelligence and Knowledge Exploration. Lecture Notes in Computer Science, vol 8284
10. Oentaryo R, Lim EP, Finegold M, Lo D, Zhu F, Phua C, Cheu EY, Yap GE, Sim K, Nguyen MN, Perera K, Neupane B, Faisal M, Aung Z, Woon WL, Chen W, Patel D, Berrar D (2014) Detecting click fraud in online advertising: a data mining approach. The Journal of Machine Learning Research 15
11. SIGKDD A (1997) Kdd data mining and knowledge discovery competition. http://www.kdd.org/kdd-cup
12. Tian T, Zhu J, Xia F, Zhuang X, Zhang T (2015) Crowd fraud detection in internet advertising. In: Proceedings of the 24th International Conference on World Wide Web, pp 1100–1110
13. Zhang W, Yuan S, Wang J, Shen X (2014) Real-time bidding benchmarking with ipinyou dataset. arXiv preprint arXiv:14077073
14. Zhu M (2004) Recall, precision and average precision. Department of Statistics and Actuarial Science, University of Waterloo, Waterloo 2

Chapter 6
Ad Fraud Detection Tools and Systems

Abstract This chapter reviews both commercial Ad fraud detection and prevention systems and the ones developed in academia. For commercial systems, they mainly emphasize on the efficiency, so fraud detection can be achieved at pre-auction level (e.g. less than 10 ms). The systems developed in academia are often more sophisticated in their designs and mathematical models. Yet the efficiency of such systems for online usages are often not strictly evaluated.

6.1 Commercial Ad Fraud Detection Systems

Most systems from commercial companies for fraud detection follow the similar flows such as Clicklab, Click Defense, and Validclick. They all try to add programs such as Javascript or iframe codes into client computers. Then they track the information fed back by these codes and determine fraudulent clients.

"Are You A Human" (AYAH) Inc. [1] provides a human verification tool to differentiate human generated traffic from bot traffic, so third parties can effectively block the traffic generated by bots. The tool collects and analyzes users' behavior by putting code on millions of sites. Once a user is verified as a human, information will be added into a "Verified Human Whitelist" and re-verified from day to day. In order to verify genuine human users, a unique feature used by AYAH is a game-based CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) [16]. For example, ATAH CAPTCHA may ask users to select and place eyes or mouths on a carton face. The task is easy, yet fun, for genuine human users, but rather difficult for non-human users, such as bots. While CAPTCHA is an effective way of differentiating human users, recently, research has investigated the possibility of using machine learning and artificial intelligence techniques to crack the CAPTCHA [4], and conclude that its is possible to solve CAPTCHA by using machine learning to attack the segmentation and the recognition problems simultaneously.

DoubleVerify (DV) Inc. [5], DV offers a unified service and performance platform, Pinnacle, which evaluates the quality of each impression delivered and the net result of each quality measure, such as the percentage of viewable impressions, fraud free impressions, brand safe impressions, and in geo targeting impressions etc.

X. Zhu et al., *Fraud Prevention in Online Digital Advertising*, SpringerBriefs in Computer Science, DOI 10.1007/978-3-319-56793-8_6

This allows DV to offer customized services for all stakeholders in the Ad ecosystem. For advertisers and agencies, DV Pinnacle pinpoints optimization actions to drive the return of investment (ROI) for in-flight campaigns. For publishers and Ad networks, Pinnacle allows optimization of inventory yield by delivering quality traffic to clients. For DSP and Ad exchanges, Pinnacle authenticates the quality of pre-bid decisions with the transparency necessary for quality inventory control. A recent study has shown that DoubleVerify uses extensive cookie-based and fingerprinting-based tracking for impression analysis and validation [6].

Forensiq Inc. [7] provides a fraud detection system working for all stages of Ad campaign, i.e. pre-campaign, in-campaign, and post-campaign. For pre-campaign fraud detection, the system will score each impression to determine the level of risk for this impression within 10 ms before DSP sending out the bid. This kind of pre-bid fraud detection is based on aggregate data and IP reputation from an evolving fraud intelligence database. While the Ad campaign is serving, a forensiq tag, which is a javascript tag inserted into publishers' websites or ADs, will track the impression and obtain real-time scores. The stage of post-campaign is for reporting after the campaign is running. During this stage, data will be collected and visualized for analyzing the risk score, fraud trends and so on.

Integral Ad Science (IAS) Inc. [13], a media valuation company, was founded in 2009. The main business of the company is to validate the quality of online Ad placements for both media buyers and sellers, by using Ad verification, optimization, and analytics solutions, including massive-scale web page classification using "active testing" [2]. IAS offers a variety of products for marketers, programmatic players, and media sellers, and the company is known for addressing issues around fraud, Ad viewability, brand risk, and true advertising quality (TRAQ). The TRAQ is a unique Ad quality scoring system for buyers and sellers to value media by assessing the following metrics: brand safety, Ad fraud, page content and structure, time viewed, share of view, and Ad clutter among others. From the fraud detection and prevention perspective, IAS uses large scale data analytics and session-based signal analysis to measure and block fraud at the impression level, in real-time.

Moat Inc. [12] was originally an Ad search engine for display advertising which aggregates online advertising information, such as commercial brands, creative, campaigns etc, to support user search [3]. For generic search engines, like Google, they often index the underlying web pages, and strip out the display Ads. Moat, on the other hand, provides an indexing and search mechanisms for online Ads. For example, when typing a keyword, such "Apple", it will help users search all creative Ad units currently running across the web. As business evolves, Moat is now offering real-time Ad analytics and provides a variety of Ad performance metrics, including viewability assessment, non-human traffic detection, audience characterization, and audience attention and engagement evaluation. These metrics not only validate the Ad impression, but also assess the audience, so both advertisers and publishers know who they are reaching and whether creative delivers interactions and captures the audience's attention, or results in audience engagement.

ValidClick Inc. [10, 14] develops a real-time click fraud detection system for the affiliate network. The system consists of four parts, i.e. visitors, affiliate web

sites, click verification web server, and advertisers. Every time a visitor makes a request on the affiliate web site, the affiliate will make another request to the click verification web server with visitor's request information, IP address and agent browser information. Then, click verification web server will send visitor's request information to advertiser and obtain the relative advertisement. Click verification web server will generate and store a verification ID for each advertisement and meanwhile, URL of the advertisement, visitor's IP address and agent browser information will be stored in click verification web server's database. Afterwards, the advertisement will be on the affiliate web site through the click verification web server. A client side script provided from click verification web server will be executed on the affiliate web site when certain events are triggered by visitors such as onmouseover event. Thus, verification ID, visitor's IP address and agent browser information, size of the browser window and name of the web page will be sent and stored in click verification web server's database. When a visitor clicks the advertisement, the request information along with the URL of the advertisement will be sent to the click verification web server. Information stored in the database will be retrieved according to URL of advertisement. Thereby, based on these information, click verification web server will check the validity of this click by examining rules set up for affiliates. The rules can be checked to see if the browser window is sufficiently large or the IP address is from suspicious countries or regions.

White Ops Inc. [9, 15] proposes a remote control detection system for preventing fraudulent traffic. A remote control can communicate with malware installed on computers of local users and will guide malware to execute commands, e.g. mimic human behavior. This system consists of three stages. By inserting the code snippet into the requested web pages, the first stage is to collect performance metric such as frequency data which refer to the frequency of updating events such as mouse movement movements. Limited by the network bandwidth, remote control agents will cause lower frequency than local users. The second stage is to compare the collected performance metric with characteristics for human activity and remote control activity. Finally, record the result based on the second stage and collect more results by repeating the three stages to form the report for local users and remote controls.

6.2 Ad Fraud Detection Systems in Academia

In addition to the commercial industry, Ad fraud detection and prevention has also received significant attentions in academia, with many prototype systems being proposed to fight fraud.

Ge et al. [8] developed a collaborative click fraud detection Ad prevention system (CCFDP). The main advantage of this system is collaboration between server side logs and client side logs. The server side logs includes tracking ID, Client IP, Client User Agent and cookies. Mouse movement such as mouse over and scroll bar movement and clicked link are from client side logs. Three roles are employed

in this system, which are (1) Global Fraudulent Database (GFD) for storing both server side and client side logs. (2) Monitored web server for ignoring the request from fraudulent sites. (3) Client computers. The major process of detecting fraud is as follows:

- Client computers send a request to a website.
- Monitored web server will check the request first by sending server side logs to GFD. If the score of the request is higher than the default threshold in GFD's score system, the request will be identified as fraudulent. Then monitored web server will abandon this request.
- After monitored web server responds to the client computer with tracking program and ID, the program will send the client side logs to GFD continuously for detecting fraud.

In this system, simple tricks are applied to detect action fraud such as repeated clicks for affiliate fraud and the client's IP from under-developed countries for conversion fraud. For bot and malware fraud, the system will check the mouse movement, page view time and other activities from client side logs and score them respectively. The client with overall score higher than the default threshold will be identified as fraud.

Liu et al. [11] implemented a system called DECAF to efficiently detect placement fraud in mobile apps from app stores. DECAF consists of UI Action channel and UI Extraction channel. UI Action channel employs the Monkey which is an automation tool to trigger actions such as clicking and scrolling. Different Actions will cause different states, i.e. different pages. Then UI Extraction is used to detect fraud based on the structure and content of these pages.

References

1. Areyouahuman (2016) http://areyouahuman.com/
2. Attenberg J, Ipeirotis P, Provost F (2015) Beat the machine: Challenging humans to find a predictive model's "unknown unknowns". Journal of Data and Information Quality (6):1–17
3. Barnard L, Kreiss D (2013) A research agenda for online political advertising: Surveying campaign practices, 2000–2012. International Journal of Communication (7):2024–2066
4. Bursztein E, Aigrain J, Moscicki A, Mitchell JC (2014) The end is nigh: Generic solving of text-based captchas. In: 8th USENIX Workshop on Offensive Technologies (WOOT 14), USENIX Association, San Diego, CA, URL https://www.usenix.org/conference/woot14/workshop-program/presentation/bursztein
5. DoubleVerify D (2016) http://www.doubleverify.com/
6. Englehardt S, Narayanan A (2016) Online tracking: A 1-million-site measurement and analysis. In: Proceedings the 23rd ACM Conference on Computer and Communications Security
7. Forensiq (2016) https://forensiq.com/
8. Ge L, King D, Kantardzic M (2005) Collaborative click fraud detection and prevention system (ccfdp) improves monitoring of software-based click fraud. E-COMMERCE 2005 p 34
9. Kaminsky D (2015) Detection and prevention of online user interface manipulation via remote control. US Patent App. 14/620,115

10. Linden J, Teeter T (2012) Method for performing real-time click fraud detection, prevention and reporting for online advertising. US Patent 8,321,269
11. Liu B, Nath S, Govindan R, Liu J (2014) Decaf: Detecting and characterizing ad fraud in mobile apps. In: 11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14), USENIX Association, pp 57–70, URL https://www.usenix.org/conference/nsdi14/technical-sessions/presentation/liu_bin
12. Moat (2016) https://moat.com/
13. Science IIA (2016) https://integralads.com/
14. Validclick (2016) http://validclick.com/
15. Whiteops (2016) http://www.whiteops.com/
16. Yu H, Riedl M (2015) Automatic generation of game-base captchas. In: Proceedings of the 2015 Workshop on Procedural Content Generation

Chapter 7
Conclusion

Online advertising fraud represents a significant portion of deceiving actions in digital advertising systems which use numerous technologies to derive illicit returns. Even the most conservative estimation has shown that more than 10% of Ad inventory is consumed by bot or fraud impressions. Despite of the fast growth of the computational advertising in modern communication networks, no comprehensive literature review or research documentation exists to summarize forms of fraud in Ad systems. In this book, we provided a comprehensive review of fraud activities in Ad systems, by using a tiered taxonomy to summarizes Ad fraud at different levels and from different perspectives. Our taxonomy categorizes Ad fraud into three major categories, including (1) placement fraud, (2) traffic fraud, and (3) action fraud, with each category focusing on publisher web sites/pages, network traffic, and user actions, respectively. Our literature review provides direct answers to key questions such as the major types of frauds in Ad systems, key approaches and characteristics of different types of fraud, major methods used to detect Ad frauds, and ground truth, measures, tools available to assess fraud and support research in this domain. This book delivers a first hand research guidance for online Ad fraud prevention. It also serves as technical reference for industry practitioners or developers to design their own fraud defending systems.

X. Zhu et al., *Fraud Prevention in Online Digital Advertising*, SpringerBriefs in Computer Science, DOI 10.1007/978-3-319-56793-8_7

Glossary

Publisher The owner of websites for selling advertisement placement.

Advertiser The buyer of advertisement placement.

Ad Network The bridge between publisher and advertiser.

Campaign Advertising plans and actions centred around a pre-specified advertising theme and objective.

Impression A request of the advertisement display sending from a client side to the publisher website.

Placement A specific arrangement of Ad banners on a web page, such as the position (top, bottom etc.) of the Ad banner on the page. A placement can be as specific as an Ad banner on a particular page or broadly represent an entire website. Advertisers and publishers use Ad placement to define where Ads are displayed on the page, allowing placement targeting. A placement ID can be shared across multiple pages.

Creative The artwork or the content of the advertisement intended to be displayed in the Placement.

Ad Exchange The Exchange platform of different Ad networks.

Landing Page A webpage appearing in response to the viewers' clicks on the advertisement.

Tracking Pixel HTML code containing an (1×1 pixel) image inserted in the landing pages to track user information.

Click A click event of the mouse or other input devices on the advertisement displaying at the client side.

Conversion Valuable business actions, such as purchase, sign-up, registration, followed by the Ad clicks.

X. Zhu et al., *Fraud Prevention in Online Digital Advertising*, SpringerBriefs in Computer Science, DOI 10.1007/978-3-319-56793-8