A Hybrid Abnormal Advertising Traffic Detection Method

Kun Wang, Guohai Xu, Chengyu Wang, Xiaofeng He*
Shanghai Key Laboratory of Trustworthy Computing,
School of Computer Science and Software Engineering, East China Normal University,
Shanghai, China
Email: wangkun940427@gmail.com, guohai.explorer@yahoo.com, chywang2013@gmail.com, xfhe@sei.ecnu.edu.cn

***Abstract*—Abnormal traffic is pervasive in the online advertising market. There are various cheating approaches while traditional anti-fraud methods are only effective for specific patterns. Combining the rule-based methods with supervised classification methods, we propose an abnormal traffic detection framework on both user layer and traffic layer. On the user layer, rule-based filters are designed to detect malicious users with duplicate clicks. We extract hybrid features under multi-granular time windows and train a user classifier to filter cheaters and complex spams indirectly. On traffic layer, we apply traffic filters to detect explicit fraudulent clicks and use a prediction model to detect malicious traffic with a higher precision. Extensive experiments on ground-truth data demonstrate the effectiveness of our detection method.**

I. Introduction

In recent years, online-advertising has been the leading sector of the advertising industry, which possesses the capability to target ads to proper online users[1]. However, fraudulent traffic exists widely in the online ad market. As shown in the anti-fraud research report of Admaster[2], on average, about 30.2% daily traffic is malicious, and more than 50% ad projects are suspected in varying degree. Malicious users create such fraudulent traffic for immoral profits. Click fraud is the most serious cheating approach, and many cheating clicks are generated by robot, DNS hijacking or hidden transcript[3]. Failing to detect such anormal traffic will damage the credibility of ad exchange or ad agency, discourage advertisers from participating in online advertising activities, and break the normal dealing order of advertising ecosystem at last.

In fact, some ad exchanges use certain filters to detect malicious users by setting reasonable statistic threshold or training a classifier with click features [4], [5]. Such methods mainly worked on user filtration but not detecting spam traffic directly. We can find those who have certain click patterns, but users' ID are required before using these filters. In previous work, many detection methods did not take full advantage of the impression and conversion information, and only paid attention to click behavior discovery. Further more, the abundant traffic attributes, such as ad projects, ad positions and media platform had not been utilized in feature extraction.

Considering such questions above, a hybrid abnormal traffic detection method is proposed in this paper. In order to discern spam clicks with different cheating patterns, we construct a series of rule-based filters and classifiers on both user layer and traffic layer.

On the user layer, we set up several rule-based filters to recognize explicit frauds by capturing the differences between the benign and suspected users on their click pattern. Then, a binary classifier is built to predict cheating users with complex behaviors. But these methods have small error rates on traffic filtration, because we discard all the traffic from malicious Cookies. However, the Cookie we used to identify user is a browser ID, which can be used by both benign and malicious users sometimes on public device.

To improve the precision of traffic filtration at user layer, a traffic layer is appended into our framework. On the traffic layer, we focus on malicious traffic detection directly. Several rules about behavior switch are used to detect inconsistent identity and untimely non-human clicks. A new classification model is introduced into the framework to predict abnormal traffic with hybrid features.

In this paper, we present a hybrid fraud detection framework. This framework is applied to large scale advertising logs acquired from an ad exchange platform. Specially, the key contributions of this paper are as follows:

- We address the fraud detection problem on both user layer and traffic layer. The former layer filters abnormal users and their clicks indirectly, and the latter layer detects suspicious traffic straightly.
- Rule-based filters and supervised classifiers are applied on each layer. Certain simple cheating behaviors can be captured by our rules and the hidden ones can be found in classification model.
- In addition to click data, we make use of impression and conversion logs in our methods, and we extract new features from multiple time windows and ad attributes.

The rest of this paper is organized as follows. We begin in Section 2 to discuss related work about spam ad traffic detection. Next, in Section 3, we present our framework. We validate and analyze the detection methods on ground-truth ad data in Section 4 and conclude in Section 5.

II. Related Work

There are a number of solutions for avoiding spam traffic in online advertising domain. Most of the previous research focus

*Corresponding author

978-1-5386-3120-1/17 $31.00 © 2017 IEEE
DOI 10.1109/ICBK.2017.50

on how to detect click frauds. The rule-based approaches and supervised learning models are the most common methods. Besides, spam detection in search advertising and search engine also makes good advances.

Statistic rule-based filters are widely used to find malicious users in click data stream. Lahiri[4] tracked continuous items in data stream by setting thresholds of counters. Bloom filter is an efficient data structure to detect cheating users with the same click behavior. Zhang[7] segmented the data stream with jumping window and sliding window models. Users who click in duplicate windows will be filtered. Similarly, Metwally applied Bloom filter to build a repeating click filtration model with landmark window and data stream rules[8]. They also worked on discovering cheating communities that consist of ad publisher or media platform, by computing the similarity of traffic[9].

Due to the good performance of machine learning algorithms on solving complicated problems, many researchers begin to utilize supervised classification models in their work. As switching the fraud detection problem to classification problem, we can get a reasonable user classifier by automatically learning different features of users. This method has been applied easily to detect spam e-mails, reviews and users in social media, because the published content, social attributes and social network can be used to model a user[10], [11], [12]. Gao[12] proposed to analyze the click behavior transition of users and predict cheating user on social media with session features and click categories.

However, advertisers and ad exchange can hardly obtain users' profiles and released content on various media platforms. They can only get the ad related information and device information, such as creative ID and IP, which limits the analysis of ad frauds. Haddadi[13] tried to place bluff ads on the sites and tracked the heavy clicks on these positions to filter suspected users. Perera[14] et al. utilized an ensemble method to detect malicious publishers on mobile client, which gained higher performance than single classifiers. Taneja[15] selected many time related features and used them to detect mobile ad click frauds, which solved the label imbalance problem of row data. At the aspect of feature extraction, these work inspired us to create new features from time slices and combine ad attributes with traditional click count features. Finally, we have verified the effectiveness of these features on our data set.

Spam detection research on search advertising also make good process. Duskin[5] built a user classifier to trace fraudulent search clicks. They studied query difference between human and robots and extracted some features from users' profiles, query attributes and click attributes. Certain researchers made use of click-through-rate to produce related features to classify spam contents in search engine[16]. In cases where the CTR information is not available, this method is not appropriate to utilize. At present, Google, Facebook and Yahoo have made great achievements in the field of abnormal traffic detection. AdWords system of Google adapted real-time rule-based and offline filters to discern malicious IP[6]. Google had integrated this system as a module of their search engine system, which played an important role in fraud detection.

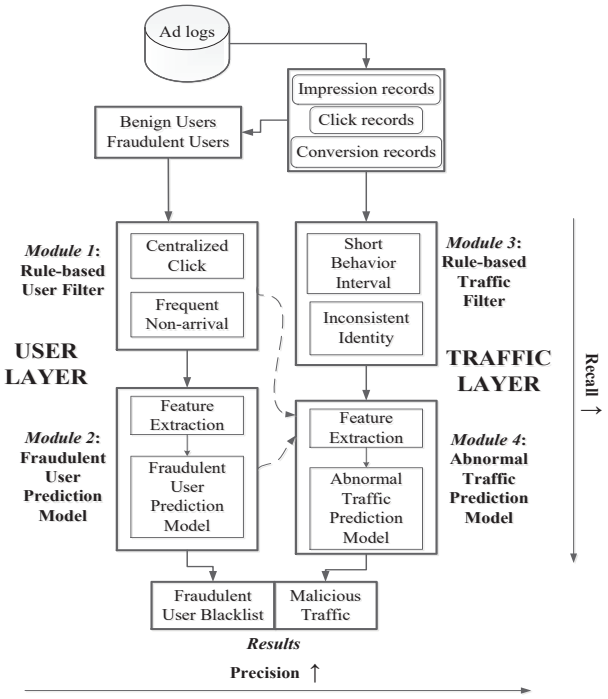


Fig. 1. Hybrid Abnormal Traffic Detection Framework. The solid line between modules represents the order of the hybrid method we use in this paper while the dash line is the feasible sequence we can use in practice.

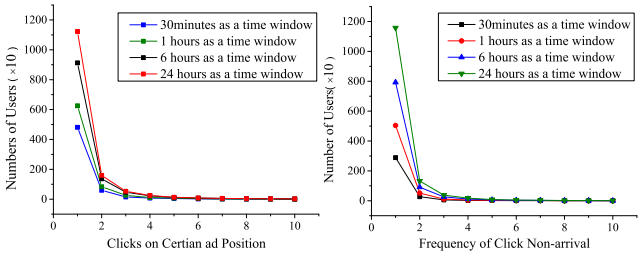
Similarly, early work[17] integrated statistic rules, classifier and clustering-based filters into a stage-wise architecture, which worked better than single method. However, this paper focused on discovering cheating users and media communities immediately, and the cluster analysis is not suitable for online system directly. In our framework, based on malicious user detection and fraudulent click filtration in disparate methods, we design a different abnormal detection architecture.

III. Hybrid Abnormal Detection Framework

A. Framework Overview

In common sense, relied on single detection method, we can filter a part of spam traffic in certain patterns. In order to detect more malicious clicks, we try to organize multiple methods in a reasonable way. Therefore, we present a multi-modules hybrid detection method. With such a detection structure, we can discover a great many of abnormal traffic and acquire accurate predictions.

As shown in Fig.1, the framework consists of following 4 modules. 1) Rule-based user filter. This module aims at detecting cheaters with heavy click behaviors. We set up two filters in this module to identify users with *frequent non-arrival* and *centralized click*. 2) Fraudulent user prediction model. This module applies a user classifier based on features extracted from multiple click attributes. 3) Rule-based traffic filter. New filters are used to find malicious clicks directly in this module. We analyze the cheating patterns by merging click with impression and conversion behaviors. 4) Abnormal traffic



(a) Centralized Click (b) Frequent Non-arrival

Fig. 2. Statistic Distribution of Normal User Clicks.

prediction model. This module constructs a traffic classifier that can predict suspicious clicks directly with similar features in module 2.

We parallelize traffic and user detection layers in our framework and list rule-based filters and supervised classifiers in a linear sequence on each layer. On each layer, classifiers process normal traffic or users predicted by rule-based filters and identify the missort frauds further. Modules on different layers return disparate results independently, as shown in Fig.1. We can filter traffic of cheating users in blacklist, but this result has a lower confidence than direct predictions in traffic layer. Modules on traffic layer are more suitable for online fraud detection because of faster processing speed. The framework minimizes the dependencies among these modules, we can arrange modules in other forms according to different demands. In addition, we can append new filters or classification models to every module flexibly according to diverse situations. For example, we can use multiple classifiers in *Module 4* and return an ensemble result of these models.

B. Module 1: Rule-based User Filter

By aggregating users' clicks together, we can effectively find an explicit or implicit click pattern from the click sequence. The common filters focused on those with duplicate clicks[7], [17]. We also think about the concentration of malicious users' clicks and draw up *centralized click* and *frequent non-arrival* strategies as described below.

1) Centralized Click Filter: In deep analysis, we find that the frequency of user's clicks on certain ad position follows a $Zipf$ distribution in different time frames, as shown in Fig.2a. This means that the frauds may click with a higher centrality on certain ad position than most normal users. For example, unreliable publishers or medias may click all the ads on a certain position of their sites no matter which ad is shown. In order to set a more precise threshold of click times, we use p-quantile value to get a reasonable upper bound λ_1 of normal users.

2) Frequent Non-arrival filter: Once a user's click on certain ad creative navigates to advertising page, there is the record of user's arrival in the conversion log. With observation of data, we find that the frequency of non-arrival in different time granularities also follows the $Zipf$ distribution, which is shown in Fig.2b. Therefore, there is a upper limit λ_2 of this behavior of normal users in a short time interval, and a mass of non-arrivals may be on behalf of malicious frauds.

C. Module 2: Fraudulent User Prediction Model

High-level cheaters will restrict click frequency as close as normal users, above filters may be unable to detect such cheaters. The classification model is an efficient method to detect frauds with complex patterns in computational advertising[14], [15]. Based on the observation of data, we find that 99.4 percent of Cookies have homogeneous clicks, which are all malicious or not. So we assume that the click traffic of a Cookie in time window T is homogeneous. Specially, we deem that all clicks of cheating users in time T are anomaly.

Comparing difference of click attributes between benign and fraudulent users, we model users with hybrid statistic click features, and these specific features are described as follows.

(1) Distinct Count Feature. Cheaters may change the IP or user agent to disguise as normal traffic. So we count the number of total clicks and distinct IP, ad project, ad position of users in period T as count features.

(2) Click Ratio Feature. This kind of feature is calculated with total click frequency and distinct count feature. For example, the average click frequency of normal users on ad position is less than frauds. Tab.I lists all the click ratio features and their descriptions used in our model.

TABLE I
FEATURES USED FOR USER CLASSIFIER

	Feature	Description
(1)	total clicks	frequency of user click
	distinct IP	number of distinct IP of user
	distinct project	number of distinct ad project
	distinct position	number of distinct ad position
	distinct media	number of distinct media platform
(2)	total clicks/distinct creative	average clicks on ad creative
	total click/distinct project	average clicks on ad project
	total clicks/distinct position	average clicks on ad position
	total clicks/distinct media	average clicks from media
(3)	clicks in night	total clicks from 00:00-06:00
	clicks in evening	total clicks from 18:00-24:00
	var of day parts	variance of clicks of 4 day parts
	mean(var) in hours	mean(variance) of clicks in hours
	mean(var) in minutes	mean(variance) of clicks in minutes
	mean(var) in seconds	mean(variance) of clicks in seconds
	mean(var) of interval	mean(variance) of click intervals

(3) Click Time Feature. Time related features are also important to describe continuity and centrality of users' behaviors. We divide large time window T into isometric subwindows $t(t \subseteq T)$ in different granularities and then gather users' click features on such multiple subwindows.

- Day Part window. We separate a day into four parts that six hours in each part. The variance of clicks among four parts, and click frequency of users in each part are used as features.
- Hour/Minute/Second window. The variance and mean of clicks of users among hour, minute or second windows are calculated respectively as features.
- Click Interval window. The time interval of contiguous click reflects the continuity of user click. We use the mean and variance of click intervals as features.

With the hybrid of 24 features in this module, we build user classifier to detect whether a user is fraud directly and filter suspicious clicks of frauds.

D. Module 3: Rule-based Traffic Filter

The rule-based user filters above are used to find cheating users with mass clicks. However, rules in this module focus on detecting malicious click record immediately.

The methods in previous work mainly studied click behaviors without considering information about impression and conversion. In this module, we link different behaviors of traffic above to filter spam with two types of rules: *short behavior interval* and *inconsistent identity*.

1) Short Behavior Interval Filter: Abnormal traffic may click no matter the ad is displayed or not, whose interval between impression and click is less than the human response time. After the page is displayed, the normal user needs appropriate time to browse the ad before making a click. But cheating web scripts may shorten such intervals and cause an induced click. The anormal interval in ad traffic log is less than a predefined threshold λ_3.

2) Inconsistent Identity Filter: Cheating users are likely to change the IP, Cookie and other device information meanwhile to disguise as normal clicks. In the absence of invariant identity, it is difficult to detect such malicious traffic only with click log. Linking the impression log with click log together by traffic ID, we can retrieve the traffic with inconsistent IP and Cookie in serial behaviors.

E. Module 4: Abnormal Traffic Prediction Model

Models in module2 may missort many users with vast normal traffic, so as to improve the precision of traffic detection, we build classifier to predict fraudulent click directly in this module. With the analysis of ad click data in depth, the cases that traffic is in the same class, which comes from certain device and targets on the same ad position, are more than 99.8 percent. Traffic on the same ad project or creative also shows the same pattern. Thus, we assume that clicks from the same client or device and locates in the same ad creative or ad position are homogeneous in a short time period T.

In this module, we group single device attribute together to create combined attributes, which are more accurate client identities. The original device attributes we used are IP and Cookie while click attributes contain creative ID, position ID, ad project ID and media ID. The 44 hybrid features extracted within different time windows based on single and the combined attributes are listed as below.

(a) Category Feature. According to statistic analysis[2], the volume of cheating traffic on PC clients is higher than mobile clients. So we take mobile device ID to distinguish PC platform and mobile platform, which is named *platform category* feature. Additionally, many frauds come from video or e-business sites, so *media category* feature can reflect the possibility of traffic to cheat from different websites.

(b) Time Related Feature. Fig.3 states how we extract features on multi-size time windows. The details of time related features we use in this module are shown below.

- Period Window Features. In period window T(i.e. 1 hour$< T \leq$1 day), we count the frequency of single device ID or combined attributes, such as, frequency of Cookie in T, frequency of Cookie-IP co-occurrence, frequency of IP on ad position and etc.
- Hour/Minute/Second Window Features. The mean and variance of clicks of device ID among multiple hour, minute, second windows are calculated as features, such as mean of Cookie among hours, variance of Cookie among seconds.
- History Window Features. History information is also useful for prediction. So we inherit click features in former time window as historical features.
- Timestamp Features. We count the frequency of different attributes on a timestamp as time moment features, such as total clicks of Cookie or IP on a certain timestamp.

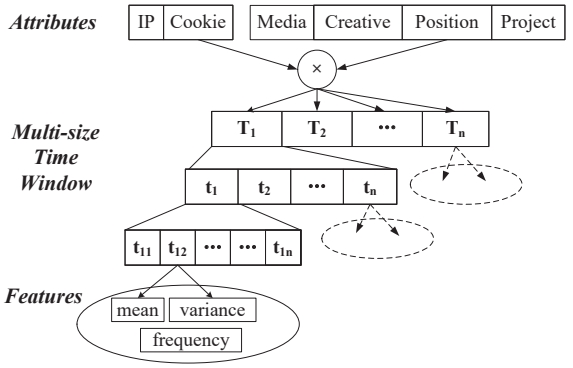


Fig. 3. Time Related Features in Traffic Prediction Model.

(c) Click Centrality Features. Comparing to the normal, cheaters may click in a more concentrated way for certain purpose, which may have a higher click ratio on certain ad position or project. Besides, the most frequent ad attributes in different time windows also represent click centrality of traffic, which can be used as categorical features.

IV. Experiments

A. Dataset

The data we used in experiments is entire ad log over 1 day which we get from an ad exchange company. There are 0.13 billion impression records, 8.5 million click records and about 1.85 million conversion records. The click logs are labeled by professional staff of that company with the feedback of publishers and advertisers, while other logs are unlabelled. The traffic has a distinct ID in all the logs, so that we can join the impression, click and conversion logs together to find new patterns. The timestamps of these logs are unbiased because the data comes from a distributed file system with synchronized clock.

The click log contains attributes as IP, Cookie, timestamps, ad creative ID, ad position ID, ad project ID, media ID, user agent, media category, mobile device ID (i.e..idfa, *if the traffic is from a mobile*), os and etc. We choose Cookie to identify a user temporarily in a short time due to the relative stability and veracity. At the same time, we assume that Cookies without fraudulent traffic are the benign while others are fraudulent. A brief description of data set for training and testing is shown in Tab.II.

TABLE II
BRIEF DESCRIPTION OF DATA SET

Description	data set (#)
Abnormal clicks	2938215
Normal clicks	5553738
Fraudulent users(Cookie)	1071770
Benign users(Cookie)	2153569

B. Experimental Setup

With the help of distributed database *Hive*[18], we can finish some simple statistic tasks, and deal with the feature engineering, training and testing work on multi-nodes cluster by using *Python*.

In *Module 1* and *Module 3*, we use p-quantile values to set relatively accurate thresholds of these filters. Considering the distribution of data in Fig.2 and counting the proportion of spam in click logs, finally, we set λ_1=5, λ_2=3 and λ_3=0.5. In fact, most public networks can allocate dynamic IP with the same net segment, so we only filter the traffic with completely different IP net segments and Cookies in filter *Inconsistent Identity Filter*.

In the module of *Fraud User Prediction Model*, we extract 24 features upon data attributes as mentioned before. To avoid adverse influence of useless features, we select top 20 features which have great influence to the classification, as shown in Tab.I. In this module, we try 5 classification algorithms as follows: Logistic Regression, Decision Tree, Naive Bayes, Random Forest and GBDT (gradient boosting decision tree, an ensemble algorithm). The Random Forest and GBDT algorithms both use 10 subtrees. We compare the performances of above 5 algorithms on traffic detection indirectly based on homogeneity observation of users' traffic. For comparision, We train a GBDT model as baseline with 16 features in [17], which we can extracted from our data.

In the module of *Abnormal Traffic Prediction Model*. We choose the GBDT algorithm as classifier to test effects of different feature combination on traffic prediction. The GBDT model consists of 5 subtrees in an ensemble way, and each subtree learns 80% features in training. We choose (c) *Click Centrality Features* in **test 1** and (b) *Time-Related Features* in **test 2**. In **test 3** we combine (a) *Category Features* and (c) *Click Centrality Features* and **test 4** uses (a) *Category Features* and (b) *Time-Related Features*. Finally, we use all features in **test 5** after feature selection.

The problem of abnormal detection needs to ensure high precision and recall to find spam traffic as much as possible. Hence, we evaluate our models with precision, recall and F1-score. We use 5-fold cross validation over data set in classifiers to acquire stable models.

C. Results analysis

Tab.III illustrates results of different classification algorithms in module of *Fraud User Prediction Model*. As shown in Tab.III, we can know that features we use in *Module2* work better than the baseline. The baseline has a high recall but low precision. Meanwhile, the hybrid features in our work can enhance the precision effectively. As known, Decision Tree is insensitive to magnitude of features and has a balanced performance on both precision and recall. Random Forest and GBDT model choose Decision Tree as weak learner in the ensemble learning algorithm. Compared with Decision Tree, the recall of Random Forest and GBDT model improves significantly in the experiments, which indicates that ensemble learning is helpful to find more fraudulent traffic. Further, GBDT model has the highest recall while its precision is similar to Random Forest.

TABLE III
PERFORMANCE OF USER CLASSIFIER ON TRAFFIC DETECTION

Classifier	Precision (%)	Recall (%)	F1-score (%)
Baseline	74.96	95.41	83.96
Naive Bayes	64.61	98.17	77.93
Logistic Regression	82.53	87.39	84.89
Decision Tree	83.36	88.20	85.71
Random Forest	85.39	92.31	88.72
GBDT	**85.02**	**98.31**	**91.18**

In *Module2* and *Module4*, we select top-k features in classifier by computing the **average impurity loss** to measure the importance of features. The result also shows the contribution of these features in our classifier. We will not discuss the computation of feature importance since the space limit, but you can get more details at the github page[1].

TABLE IV
PERFORMANCE OF TRAFFIC PREDICTION MODEL WITH DIFFERENT FEATURE COMBINATION

Test	Features	Precision (%)	Recall (%)	F1-score (%)
test 1	(c)	89.04	80.56	84.59
test 2	(b)	94.08	56.00	70.21
test 3	(a) + (c)	91.18	89.30	90.23
test 4	(a) + (b)	95.51	65.32	77.59
test 5	(a) + (b) + (c)	**96.53**	**94.89**	**95.70**

We compare the performance of different feature combinations in module *Abnormal Traffic Prediction Model* and the results are listed in Tab.IV. *Click Centrality Features* have a balanced result of precision and recall, while *Time-Related Features* can effectively distinguish the normal and the abnormal. But many frauds have similar time related features to normal traffic yet. The results have been improved by combining with category features. Finally, we can get the best performance with three type features after feature selection. In contrast to user classifier, traffic prediction model has a higher precision because multi-dimensional device attributes make traffic features more separable. Better results of traffic prediction model certify the effectiveness of our assumption in this module. But there remains certain traffic that our model can not recognize, because the malicious may have extremely similar features like normal traffic sometimes.

We use four modules of our framework to detect abnormal traffic over the whole data set and compare the results of these modules in Fig.4. We can conclude that modules in traffic layer own higher precision but lower recall than user layer. Additionally, rule-based filters has lower recall than classifier in each layer. The rules take use of centrality and time features of deceptive behaviors and the combination of multiple filters work better than single one. In addition, rules in traffic layer are more confident than those in user layer.

[1] https://github.com/KunWangR/hybrid_spam_detection

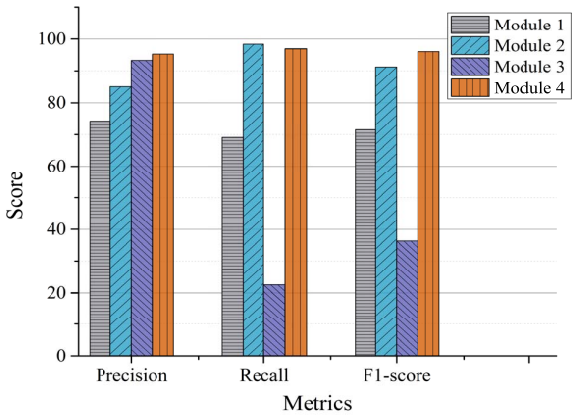


Fig. 4. Performance of 4 Detection Modules on Testing Data.

With the changes of cheating approach, cheaters will pretend as normal traffic and simulate benign behavior, which makes rule-based abnormal detection more and more difficult. In this case, rule-based filters can only be used to detect frauds with simple malicious patterns, while supervised models can find complex traffic. We choose GBDT model as prediction model in both classifier-based modules due to the best performance on our data.

According to Fig.4, although our assumption in classifier-based modules has a bit bias to reality, these two modules still work well. However, traffic prediction model has higher precision because the combined device attributes divide traffic into finer client clusters and hybrid features promote the performance of prediction. The user prediction model has really high recall because we assume anyone with fraudulent click is a fraud, but in fact, some clicks from the same Cookie are not with the same labels. For instance, traffic from a computer in public Internet cafe is not generated by the same user. So the user prediction module can recognize almost abnormal traffic from cheaters but also remove many normal clicks from a suspicious Cookie. With the complementarity of modules above, we propose such a hybrid method to detect abnormal traffic on both user layer and traffic layer.

V. Conclusion

The abnormal traffic seriously disturbs the normal deal of online advertising system. We combine the rule-based filters and classification models together on both user layer and traffic layer. A hybrid abnormal traffic detection framework is proposed in our work. Several statistic rules are applied to detect traffic and cheaters with explicit abnormal patterns. While we build a user classification model and an traffic prediction model with hybrid features, which perform well on the ground-truth data set.

Limited by the experimental data, we can not fully use the information related to the search, web content and so on. Considering the difference of frauds on PC end and mobile, we will build more customized detection module in the future. What's more, we will use graph model to find cheating communities and design a data compression process further to accelerate the calculation speed.

Acknowledgment

This work is supported by the National Key Research and Development Program of China under Grant No. 2016YFB1000904.

References

[1] Evgeniy G, Vanja J, Bo P. Introduction to Computational Advertising. ACL:1, 2008.
[2] Admaster. Adertising fraud research report of 2017.http://www.admaster.com.cn/?c=downloads&a=view&id=101.
[3] Daswani N, Mysen C, Rao V, et al. Online Advertising Fraud. Epfl, 2011.
[4] Lahiri B, Chandrashekar J, Tirthapura S. Space-efficient tracking of persistent items in a massive data stream// ACM International Conference on Distributed Event-Based Systems, Debs 2011, New York, Ny, Usa, July. DBLP, 255-266, 2011.
[5] Duskin O, Feitelson D G. Distinguishing humans from robots in web search logs: preliminary results using query rates and intervals// The Workshop on Web Search Click Data. ACM, 15-19, 2009.
[6] Tuzhilin A. The Lane's Gifts v. Google Report. Official Google Blog Findings on Invalid Clicks.
[7] Zhang L, Guan Y. Detecting Click Fraud in Pay-Per-Click Streams of Online Advertising Networks. 77-84, 2008.
[8] Metwally A, Agrawal D, El Abbadi A. Duplicate detection in click streams// International Conference on World Wide Web, WWW 2005, Chiba, Japan, May. DBLP, 12-21 , 2005.
[9] Metwally A, Agrawal D, Abbadi A E. Detectives:detecting coalition hit inflation attacks in advertising networks streams// International Conference on World Wide Web. 241-250, 2007.
[10] Chakraborty M, Pal S, Pramanik R, et al. Recent developments in social spam detection and combating techniques: A survey. Information Processing & Management, 2016, 52(6):1053-1073.
[11] Wang G, Konolige T, Wilson C, et al. You are how you click: clickstream analysis for Sybil detection// Usenix Conference on Security. 241-256, 2013.
[12] Gao H, Hu J, Wilson C, et al. Detecting and characterizing social spam campaigns// ACM Conference on Computer and Communications Security. ACM, 681-683, 2010.
[13] Haddadi H. Fighting online click-fraud using bluff ads. Acm Sigcomm Computer Communication Review, 40(2):21-25, 2010.
[14] Perera K S, Neupane B, Faisal M A, et al. A Novel Ensemble Learning-Based Approach for Click Fraud Detection in Mobile Advertising// Mining Intelligence and Knowledge Exploration. 370-382, 2013.
[15] Taneja M, Garg K, Purwar A, et al. Prediction of click frauds in mobile advertising// Eighth International Conference on Contemporary Computing. IEEE, 162-166, 2015.
[16] Wei C, Liu Y, Zhang M, et al. Fighting against web spam:a novel propagation method based on click-through data// Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval. ACM, 395-404, 2012.
[17] Song L, Gong X, He X, et al. Multi-Stage Malicious Click Detection on Large Scale Web Advertising Data// Proceedings of the First International Workshop on Big Dynamic Distributed Data.VLDB, 67-72, 2013.
[18] Thusoo A, Sen Sarma J, Jain N, et al. Hive: a warehousing solution over a map-reduce framework. Proc. VLDB Endow, 1626-1629, 2009.