Yang DH, Li ZY, Wang XH *et al.* Exploiting the community structure of fraudulent keywords for fraud detection in web search. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 36(5): 1167–1183 Sept. 2021. DOI 10.1007/s11390-021-0218-2

Exploiting the Community Structure of Fraudulent Keywords for Fraud Detection in Web Search

Dong-Hui Yang[1,2], Zhen-Yu Li[1,2,*], *Member*, *CCF*, *ACM*, *IEEE*, Xiao-Hui Wang[3], Kavé Salamatian[4], and Gao-Gang Xie[2,5], *Member*, *CCF*, *ACM*, *IEEE*

[1] *Network Technology Research Center, Institute of Computing Technology, Chinese Academy of Sciences Beijing 100190, China*

[2] *University of Chinese Academy of Sciences, Beijing 100049, China*

[3] *Global Energy Interconnection Research Institute Co., Ltd., Beijing 102209, China*

[4] *LISTIC Laboratory of Computer Science, Systems, Information and Knowledge Processing, Université Savoie Mont Blanc, Chambéry 73011, France*

[5] *Computer Network Information Center, Chinese Academy of Sciences, Beijing 100190, China*

E-mail: {yangdonghui, zyli}@ict.ac.cn; 18744021638@sina.cn; kave.salamatian@univ-smb.fr; xie@ict.ac.cn

Received December 11, 2019; accepted July 1, 2021.

Abstract Internet users heavily rely on web search engines for their intended information. The major revenue of search engines is advertisements (or ads). However, the search advertising suffers from fraud. Fraudsters generate fake traffic which does not reach the intended audience, and increases the cost of the advertisers. Therefore, it is critical to detect fraud in web search. Previous studies solve this problem through fraudster detection (especially bots) by leveraging fraudsters' unique behaviors. However, they may fail to detect new means of fraud, such as crowdsourcing fraud, since crowd workers behave in part like normal users. To this end, this paper proposes an approach to detecting fraud in web search from the perspective of fraudulent keywords. We begin by using a unique dataset of 150 million web search logs to examine the discriminating features of fraudulent keywords. Specifically, we model the temporal correlation of fraudulent keywords as a graph, which reveals a very well-connected community structure. Next, we design DFW (detection of fraudulent keywords) that mines the temporal correlations between candidate fraudulent keywords and a given list of seeds. In particular, DFW leverages several refinements to filter out non-fraudulent keywords that co-occur with seeds occasionally. The evaluation using the search logs shows that DFW achieves high fraud detection precision (99%) and accuracy (93%). A further analysis reveals several typical temporal evolution patterns of fraudulent keywords and the co-existence of both bots and crowd workers as fraudsters for web search fraud.

Keywords community structure, fraud analysis, fraudulent keyword detection, web search

1 Introduction

Web search engines provide Internet users with a simple portal to search for information quickly, and finally redirect users to the targets. Most major search engines make revenue via paid search advertising, where advertisers pay to search engines to display ads (i.e., sponsored results) alongside non-sponsored (a.k.a, organic, non-paid, algorithmic) web search results on the search result pages when a user searches a particular keyword (which may consist of more than one word) [1–3]. Advertisers bid on keywords that they want to use to trigger the display of their ads, and pay to search engines according to the revenue model. Search advertising is effective because it captures and satisfies users' need for relevant search results [4]. Advertisers

Regular Paper

This work was supported by the National Key Research and Development Program of China under Grant No. 2018YFB1800205, the National Natural Science Foundation of China under Grant Nos. 61725206 and U20A20180, and CAS-Austria Project under Grant No. GJHZ202114.

*Corresponding Author

use search engine advertising to improve their visibility among the search engine results and thus to attract potential customers. Indeed, search advertising constitutes the largest source of revenues for search engines, and therefore it has become an integral part of business model [1].

Fraud emerged with the rapid increase in money earned through search advertising. Fraudsters usually start by submitting carefully selected query keywords to search engines and making use of the results pages. The anomalous searches that are intentionally generated to increase fake traffic for a specific purpose are usually money-driven. For instance, a fraudster can generate false impressions and increase the search volume of certain keywords in order to boost their bidding price and sell them. Fraudsters can also click on rival's ads to exhaust their pay-per-click advertising budget either manually or by an automated script. This is known as click fraud. Another example is that fraudsters can repeatedly search keywords relevant to rivals' ads to generate impressions without clicks aiming at lowering the click through rates (CTR). As CTR is a key factor that influences an ad's quality score, which is further used to determine the rank order of sponsored links, fraudsters can lower the performance rank of rivals' ads on the search engine results page. Moreover, they can bid less for the advertising slots [5]. In addition, fraudsters can send certain query keywords for purposes such as reverse engineering the search engine's index, poisoning its ranking algorithm, discovering vulnerable web servers or increasing the load of search engine by forming DDoS attacks [6]. Fraudulent behavior in web search causes huge amount of financial losses① [7]. Therefore, it is important to detect such anomalous money-driven searches that are intentionally generated for a specific purpose.

The most common way of performing fraud search is to leverage bots that generate searches using the given keywords. A distinguishing search behavior pattern of bots is that they generate huge volumes of searches on a limited number of fraudulent keywords. Previous studies rely on this pattern for fraudster detection or fraudulent click detection [8–14]. Nevertheless, some new means of fraud search have been emerging, such as crowd fraud in some malicious crowdsourcing platforms, where fraudsters hire web workers to search designated keywords and/or click on designated advertisements on a certain search engine [15]. The crowd workers may not generate as many searches as bots and may search for many other keywords. The approaches on fraudster detection based on the behavior patterns of bots, as a consequence, may fail to detect the fraud made by crowd workers.

In this paper, we aim at the detection of both bots and crowd fraud workers. We achieve this goal by solving the problem from a different perspective, i.e., the perspective of fraudulent keywords. We leverage the insight that fundamental requirements for fraudsters' success can be used as discriminating features for fraud detection. Specifically, a successful fraud campaign on a target will use a bag of relevant keywords for the generation of queries, rather than individual keywords. Using a dataset consisting of 150 million web search logs of a popular search engine, we investigate and verify that this rationale can be taken as a discriminating feature of fraudulent keywords.

We next design DFW (detection of fraudulent keywords), a simple yet effective approach for the fraudulent keyword detection using the above rationale. We then evaluate the accuracy of DFW using our dataset, and finally examine the typical patterns of fraudulent keywords and fraudsters.

To sum up, we make the following main contributions.

• We model the temporal correlation of fraudulent keywords as a graph, where an edge is formed between two keywords (nodes) if they were searched by a bot within a short time period. By analyzing this graph, we discover that fraudulent keywords form very well-connected communities: the clustering coefficient of the graph is as high as 0.91.

• We propose DFW to detect web search fraud by exploiting the community structure of fraudulent keywords. Specifically, DFW mines the temporal correlations between candidate fraudulent keywords and a given list of seeds, and leverages several refinements to filter out non-fraudulent keywords that co-occur with seeds occasionally. The experiments using the dataset show that DFW outperforms the baseline approaches, and achieves high fraud detection precision (99%) and accuracy (93%).

• We comprehensively study the characteristics of fraudulent keywords and fraudsters. Surprisingly, 262 out of the top 500 popular keywords in the dataset are fraudulent keywords. The time series analysis reveals both continuous fraud search behavior and diurnal behavior for fraudulent keywords. More importantly,

① https://www.cnbc.com/2017/03/15/businesses-could-lose-164-billion-to-online-advert-fraud-in-2017.html, July 2021.

we observe that 34.25% of fraudsters show search patterns other than bots, implying the emergence of new means of fraud that existing approaches on fraud detection may not be able to capture.

The rest of the paper is organized as follows. Section 2 provides related work. Section 3 describes the background of this study and the dataset we use in this study. Section 4 elaborates on the temporal correlation in fraudulent web searches. Our detection system of identifying fraudulent keywords is presented in Section 5, and Section 6 shows the evaluation process and further analysis based on the detection result. Section 7 discusses some points of our approach. Finally, Section 8 concludes the paper.

2 Related Work

2.1 Bot Detection

Prior studies on fraud detection in web search mainly rely on behavioral characteristics to detect bots. Buehrer *et al.* [12] developed a set of features modelling the physical interaction of a user as well as the behavior of automated traffic to distinguish between searches generated by humans and bots. Sadagopan and Li [13] used Markov chain to model user sessions and computed a score by normalizing the log-likelihood by the number of transitions. They combined the score with other user session characteristics to detect outliers as user sessions generated by bots using Mahalanobis distance. SBotMiner [9, 10] leveraged the similarity of bot-generated activities to capture groups of distributed search bots. Duskin and Feitelson [14] used query rate and the minimal interval of time between different queries to distinguish humans and bots. Kang *et al.* [16] made use of CAPTCHA to extract data logs of genuine human users as training data, and then proposed a semi-supervised learning approach to classifying bot generated web search traffic from genuine human users. Haidar and Elbassuoni [17] proposed a bot detection approach based on local website navigation behavior. Guo *et al.* [18] proposed a traffic-based quasi-real-time method for CloudBot detection, where CloudBot denotes the malicious bot deployed on the hosts of data centers. Toffalini *et al.* [19] studied the problem of Google Dorking which is at the core of many automated exploitation bots, where attackers craft special search engine queries for facilitating their attacks. Shakiba *et al.* [6] proposed a semi-supervised method to detect bot-generated spam queries submitted by malicious users to facilitate their attacks.

In our work, we solve the fraud detection problem from the perspective of fraudulent keywords, given the emergence of new means of fraud and that we do not focus on a certain kind of fraud (e.g., click fraud).

2.2 Fraud Detection in Internet Advertising

Metwally *et al.* [20, 21] studied the type of fraud in Internet advertising which involved coalitions among fraudsters. They first proposed a simple approach based on Bloom filters to detecting duplicates in click streams, then modeled the detection of fraud coalitions in terms of the set similarity problem and proposed an algorithm to uncover coalitions of fraudster pairs. Immorlica *et al.* [22] studied pay-per-click marketplaces and proved that a particular class of learning algorithms can reduce click fraud. Dave *et al.* [23] studied click-spam in online advertising. They leveraged the invariant that click-spammers delivered high return on investment to offset the risk of getting caught, and presented an approach to catching click-spam in search ad networks. Li *et al.* [24] proposed an approach to detecting fraudulent clicks, which were used by some websites to obtain a higher rank. They modeled user sessions, constructed bipartite graphs to describe the relations between users and sessions as well as patterns and sessions, and used the bipartite graph propagation algorithm to detect fraudulent clicks based on the seed cheating session modes. We also present the comparison between our approach and the bipartite graph propagation algorithm that has been widely used in web spam and click spam detection. Tian *et al.* [15] studied crowd fraud which emerged with the rise of crowdsourcing platforms. They examined the characteristics of the group behaviors of crowd fraud and identified three patterns including moderateness, synchronicity and dispersivity. Based on the identified patterns, they built a parallel detection system which can find fraudulent clicks with a high accuracy. Nagaraja and Shah [25] proposed two kinds of defence against click fraud instituted via malware. They detected click fraud based on timing characteristics of click traffic. DeBlasio *et al.* [26] explored deceptive advertising and characterized the fraudulent advertiser ecosystem at Bing search engine. Wei *et al.* [27] proposed a label propagation algorithm on click-through bipartite graph to detect web spam. Haidar *et al.* [28] studied ad frauds in mobile advertising caused by false display requests or clicks. They proposed an ensemble based method to identify fraudulent ad displays. Dong *et al.* [29] investigated mo-

bile ad frauds and proposed a hybrid approach to detecting ad frauds in mobile Android apps.

3 Data: Web Search Logs

We collect web search logs from a popular search engine in China. We randomly select several front-end servers that host the web search service, and dump their HTTP-level logs for one month. The dataset consists of 150 million logs. Each log contains the timestamp (in second), user's IP address (anonymized), user agent information, the request URL and some performance related metrics (e.g., upstream response time).

Then the submitted query keywords are extracted from the log, and similar to [30], we use the request's IP address combined with User-agent as a fingerprint (i.e., user ID) to distinguish users, as it is less sensitive to the effect of network address translation (NAT) which renders IP addresses not precise to distinguish different users. We make our dataset publicly available for the community②. Each line corresponds to a log that contains the anonymized user ID, the timestamp and the query keyword ID, where each query keyword is mapped to a unique keyword ID for anonymization. In addition to the data, we also share the list of detected fraudulent keywords③, where each line corresponds to the ID of a detected query keyword. We hope the dataset will be used by other researchers to further investigate the fraud in web search.

4 Community Structure of Fraudulent Keywords

This section examines the community structure of fraudulent keywords. This requires adequate fraudulent keywords and their relevant queries. To this end, we first choose strictly a few misbehaving bots that may be a small part of all fraudsters that conduct fraud. Then we regard their query keywords as fraudulent ones. Finally, we model the temporal correlation of these keywords and analyze the graph properties.

4.1 Bots Detection

Search bots often submit large numbers of queries in long-duration sessions of continuous activities, where a session refers to a short period of contiguous time the user spends querying and examining results [14, 31]. As such, we first separate web search logs into sessions. Usually an inter-activity time interval threshold value is set to break logs into sessions. If the time interval between two searches of a user exceeds the threshold, the user starts a new session. Inspired by [30], we learn the threshold empirically from the dataset. Specifically, we compute users' inter-search time on a logarithmic scale (see Fig.1). We next fit the intervals with a mixture of Gaussian model using the expectation maximization (EM) algorithm. The fitting yields a two-component Gaussian mixture model (see Fig.1), where the first component is relevant to the intra-session behavior and the second captures the inter-session behavior. The threshold for session separation is the point where the two components' curves intersect, which represents that the probability of inter-activity time belonging to two components is equal. This threshold is 30 seconds in our data. Applying this threshold, we obtain 38 373 621 sessions, of which 1 637 932 sessions contain at least two query keywords.

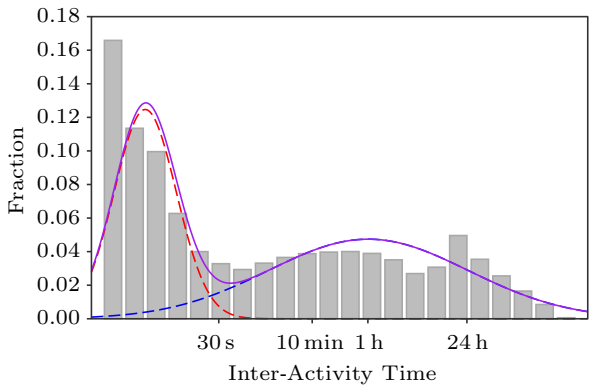

Fig.1. Distribution of query intervals with the Gaussian mixture model. The two Gaussian components are colored in red and blue, respectively, and their mixture is colored in purple.

We compute for each user the length of the longest session and the number of queries in that session, and use the Gaussian mixture model to cluster users. The Akaike information criterion (AIC) [32] determines a two-component mixture model. One cluster shows a long average session length (560 s) and a large number of queries in a session (390 on average), which are the distinctive features of bots (either crawler bots or fraud bots). In contrast, the average session length of the other cluster is 19 s and the average number of queries is only 2.4, which are the characteristics of normal users.

②https://bit.ly/2If0jak, July 2021.
③https://bit.ly/3oLZoxT, July 2021.

To further distinguish the crawler bots and the fraudulent bots in the first cluster, we leverage two statistics: the number of queries and the number of unique keywords searched by each user in the cluster. Our rationale is that crawlers gather information by submitting large quantities of query keywords with each keyword queried only once or twice. On the contrary, bots involved in fraud should conduct large quantities of queries aiming at certain targets. We again use a Gaussian mixture model to cluster these users based on the two statistics. The AIC metric indicates a model of four components. Table 1 lists their features.

Table 1. Features of the Four Clusters of Bots

Cluster	Avg. Q	Avg. K	Label
1	109	43	Light fraud bots
2	4 239	245	Light crawlers
3	16 132	10 832	Heavy crawlers
4	257 140	32	Heavy fraud bots

Note: Avg. Q and Avg. K are the abbreviations for average number of queries and average number of keywords, respectively.

We notice that the fourth cluster, which consists of 80 users, exhibits the exact behavior of heavy fraudulent bots. The 831 query keywords from these bots can be safely regarded as fraudulent ones.

4.2 Community Structure Analysis

We model the temporal correlation of the query keywords of the identified fraudulent bots as a graph, where nodes are the keywords and an edge between two keywords reflects their co-occurrence within a short period of time. Specifically, for each keyword k queried by a fraudulent bot, we associate a small time window w_k (2 minutes in our setup), where k's appearance is centered in w_k. An edge between k and each keyword searched in the time window w_k is added in the graph. Fig.2 depicts the graph built with the 831 keywords searched by the identified bots using a force-directed graph drawing algorithm [33]. We can clearly see that these keywords form several well-connected components. A further manual examination reveals that the keywords in the same component seem to serve the same fraud target.

To quantify the community structure of the graph, we further compute the clustering coefficient of the graph [34]. The clustering coefficient is a measure of the degree to which nodes in a graph have the tendency to cluster together into tightly connected neighborhoods. The clustering coefficient of a graph is derived from that of individual nodes, which can be computed as the ratio of the number of edges between the node and its neighbors to the number of edges that could possibly exist between them.

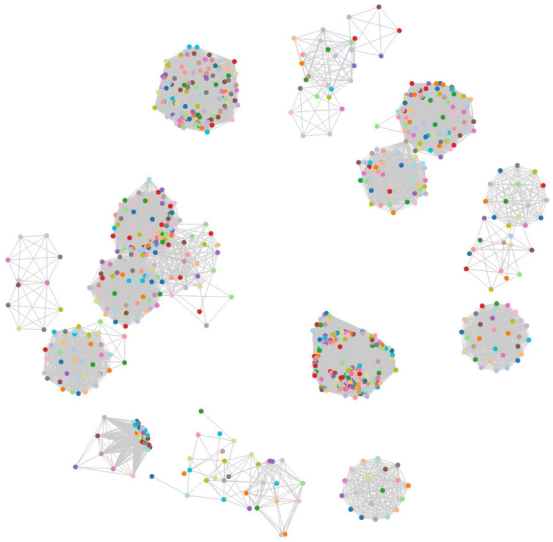

Fig.2. Fraudulent keyword graph: each coloured node represents a keyword.

The clustering coefficient of the graph is as high as 0.91. In comparison, a random graph with the same number of nodes and edges is only 0.09. This clearly indicates the well-connected community structure of fraudulent keywords, which inspires us to detect fraudulent keywords by mining the temporal correlation between candidates and some given seed fraudulent keywords.

4.3 A Strawman Solution for Fraud Detection

A simple approach to exploiting the community structure of fraudulent keywords for fraud detection is as follows. For each appearance of a seed fraudulent keyword in a user's search stream, we set a time window and obtain a seed-centered segment. The keywords in that segment are deemed to be close enough to the seed. If the times that a keyword appears in a seed's time segments is larger than a threshold, the keyword can be marked as a fraudulent one. However, this strawman solution is not practical for the following two reasons. First, some non-fraudulent keywords may also co-occur many times with seed fraudulent keywords. For instance, crowd workers may search some popular keywords along with the fraudulent keywords. Second, completely relying on a time window to capture the temporal correlation is too coarse to obtain

accurate detection. Our evaluation in Section 6 confirms the low accuracy of such a strawman solution. In Section 5, we will use several refinements to address the issues of the strawman solution.

5 Fraud Detection System DFW

This section details the design of DFW (detection of fraudulent keywords) that leverages the community structure of fraudulent keywords. We begin with an overview of the design and then present the details.

5.1 System Overview

Fig.3 gives an overview of the approach. DFW starts with the keywords co-occurring with seeds as fraudulent keyword candidates, and then filters out non-fraudulent keywords step by step to refine the detection results. The input of DFW consists of user search logs and a list of seed fraudulent keywords that reflect the search engines' intents on the fraud they want to uncover. DFW outputs a list of newly detected fraudulent keywords.

DFW first filters out the keywords that rarely co-occur with seeds, and keeps the remaining keywords as candidate fraudulent keywords (Subsection 5.2). This coarse detection allows DFW to only focus on the subsets of keywords that are more likely to be fraudulent according to the given seeds. Since the majority of keywords are expected not to be fraudulent, this step will significantly improve the efficiency of the identification.

Next we develop a refinement algorithm to finely exploit the community structure of fraudulent keywords (Subsection 5.3). The fraudulent keywords for the same purpose of a seed will manifest similar patterns of co-occurrence with the seed. We transform the candidates' occurrence patterns relative to individual seeds as vectors, and apply unsupervised clustering on the vectors to find those showing the similar patterns. The candidate fraudulent keywords that occasionally co-occur with seeds will then be filtered out.

Finally, DFW removes the popular non-fraudulent keywords that happen to show the similar search patterns with the fraudulent keywords (Subsection 5.4). This will happen for two reasons. First, a user ID that is constituted by the combination of IP and user agent filled in the HTTP header may correspond to several users. The searches of popular keywords (e.g., "weather") by normal users may mix with those by fraudsters. Second, a crowd fraudulent user may run automatic fraudulent search generation in background, and meanwhile search popular keywords in browsers. To accurately filter out the non-fraudulent popular keywords, DFW leverages the observation that they co-occur with fraudulent keywords only occasionally. This observation is illustrated in Fig.4, where a node represents a candidate fraudulent keyword and an edge between two nodes exists if they co-occur in a user's search stream within a short time period. It is notable that the popular non-fraudulent keywords connect with the fraudulent keywords loosely in graph, while the fraudulent ones are tightly connected. The remaining keywords after the above three steps of refinements will finally be identified as fraudulent ones.

It is worth noting that while the above refinements are inspired by some well-known techniques, DFW com-

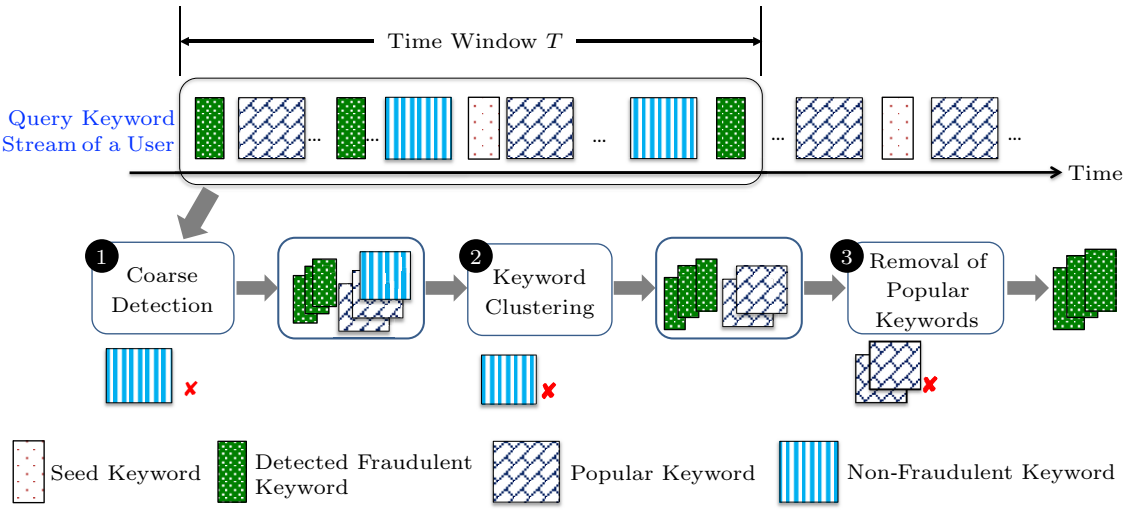

Fig.3. Overview of the fraudulent keyword identification.

bines them in such a way that effectively solves the web search fraud identification problem. Therefore, the novelty of DFW lies in the exploitation of the community structure of fraudulent keywords with a combination of these techniques for the effective detection of fraud.

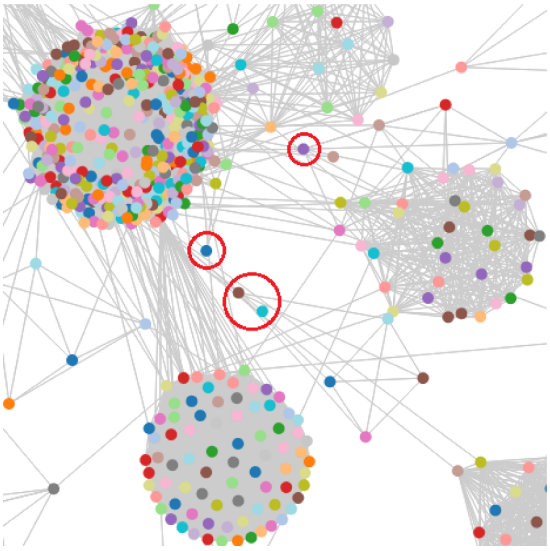

Fig.4. Illustration of the popular non-fraudulent keywords (circled in red) connecting loosely with the fraudulent ones in communities.

5.2 Coarse Detection

The first step is to filter out the keywords that rarely co-occur with seeds in individual users' search stream. To this end, we generate for each seed appearance a seed-centered time frame segment of length T, which consists of the keywords whose timestamps of corresponding queries fall within the range $[t_{\text{seed}} - T/2, t_{\text{seed}} + T/2]$, where t_{seed} is the timestamp of the seed keyword, and T is the time window size we set. Let S denote the set of all time frame segments.

We use the TF metric [35] to filter out candidate keywords that are less correlated to the seeds. Given a keyword k and a time frame segment $s \in S$, the term frequency is

$$tf(k,s) = \frac{n(k,s)}{\sum\limits_{k' \in s} n(k',s)},$$

where $n(k,s)$ refers to the times k appears in s.

We calculate $tf(k,s)$ for each query keyword k in each segment s, i.e., each keyword occurrence with seeds, and set a threshold on TF to filter out non-fraudulent keywords. Fig.5 shows the cumulative distribution function (CDF) of TF values of all keywords in all segments. The low TF values indicate that most of the keywords rarely co-appear with the seeds. Nevertheless, some are frequently seen along with the seeds in short periods of time. We observe that about half of the keyword occurrences with seeds have a TF value less than 0.02, and about 20% of the keyword occurrences will be filtered when the TF threshold is set to 0.01. Based on the distribution and considering that this is the first step in the system processing flow, we conservatively set the TF threshold as 0.01: the keyword occurrences with TF values below the threshold are filtered out.

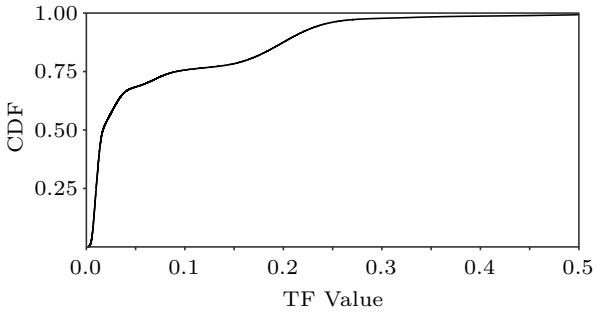

Fig.5. Distribution of TF values that measure the co-occurrences of keywords with the seeds.

5.3 Keyword Clustering

We then take into account the pattern of co-occurrence with seeds by using a vector to represent each candidate keyword and cluster the obtained vectors using the X-means clustering algorithm [36] as it can determine the number of clusters automatically.

A user's query stream is divided into segments with the time frame of a segment being T. For a query keyword q in a segment s associated with a seed keyword q_{seed}, a d-dimensional vector $\boldsymbol{V} = (v_1, v_2, ..., v_d)$ is generated, where $d = n(q_{\text{seed}}, s)$ is the times q_{seed} appears in s. We call d the dimension of segment s. Let $t_1 < t_2 < ... < t_{n(q,s)}$ be the timestamps of each occurrence of q in s, and $t_{s1} < t_{s2} < ... < t_{sd}$ denote that of q_{seed} in s. We assign $v_i = 1$ if q appears close to the i-th occurrence of q_{seed}, i.e., there exists a t_j subject to $|t_j - t_{si}| < t_w$, where t_w is a threshold value we set. Otherwise, v_i is assigned 0. t_w is a design parameter that reflects the closeness of a keyword to the seed, and it is set to 2 minutes in our current design.

Fig.6 gives an example of converting query keywords to vectors, where the centered seed keyword is s. Since the seed keyword s appears 5 times in the time window, each keyword is converted into a 5-dimensional vector. The query keyword a appears 4 times closely

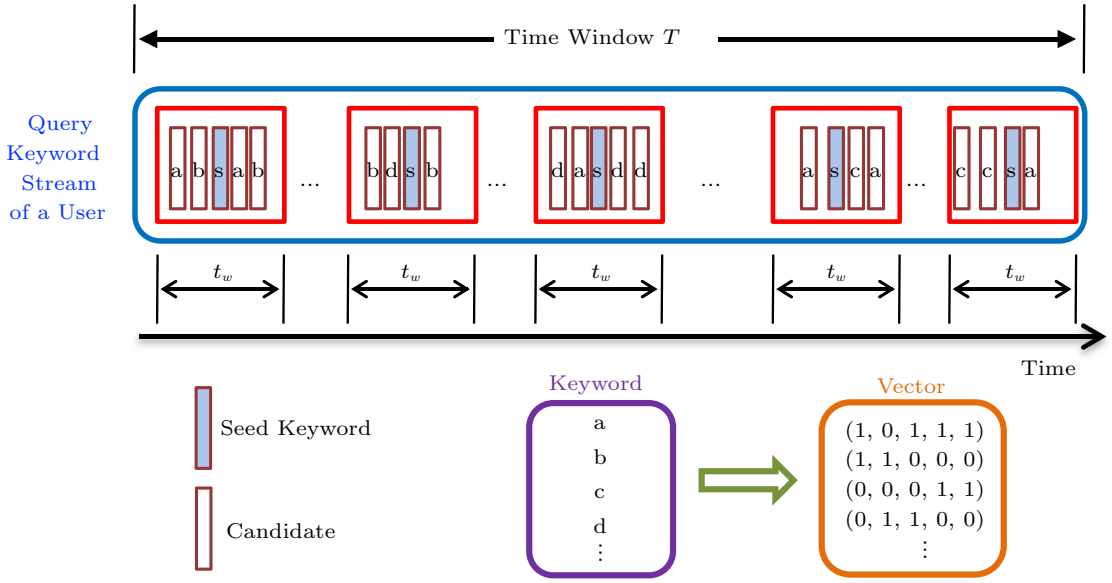

Fig.6. Example of how to convert query keywords to vectors.

with the seed except for the second appearance of the seed, thereby the vector representing a is $(1, 0, 1, 1, 1)$.

The time window of a segment T affects the dimensions of keyword vectors. Intuitively, a larger time window is preferred in order to obtain larger size vectors, which may improve the accuracy of clustering. Nevertheless, a larger time window will increase the processing time of clustering in practice. We evaluate the number of dimensions of the keyword vectors by varying the time window T (5 min, 10 min and 15 min) in Fig.7. As expected, a larger T increases the number of dimensions. We take T as 10 minutes to balance between the processing time and the accuracy. Indeed, increasing the identified fraudulent keywords with $T = 15$ min does not improve the detection accuracy significantly, because the remaining keywords after keyword clustering are almost the same as those obtained with $T = 10$ min, and the last step (removal of popular query keywords) is independent of T.

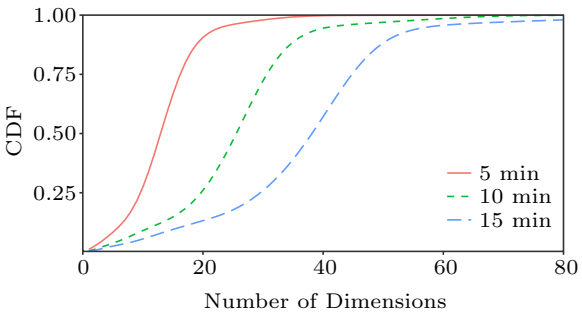

Fig.7. CDF of the number of dimensions of keyword vectors when varying time window T.

We then use X-means [36] to cluster query keywords in each segment. X-means is an extension of K-means which can determine the number of clusters automatically. We run the X-means clustering on each seed appearing in segment s. The clustering as such can be run in parallel. Keywords in the same clusters have similar patterns of co-occurrence with a seed keyword, and can be reasonably considered from the same bag of fraudulent keywords.

We plot the CDF of the number of keywords in all clusters we obtained from the dataset in Fig.8. While some clusters contain over 20 keywords, most are small because they contain non-fraudulent keywords accidentally searched by a user along with the seeds. The median cluster size is 3, and thus only the clusters having more than three keywords are kept for further refinement.

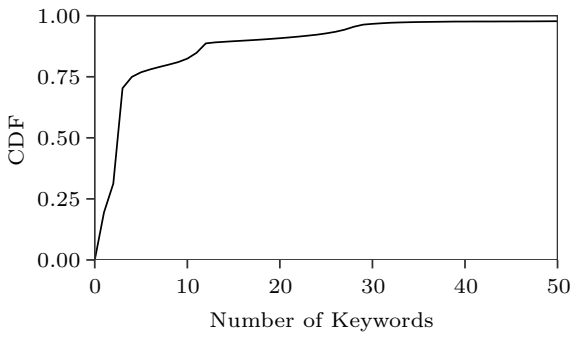

Fig.8. CDF of the number of unique keywords in individual clusters.

5.4 Removal of Popular Query Keywords

Query keywords remaining up to now may be composed of popular non-fraudulent keywords and the fraudulent keywords that we intend to identify. We use the idea of tie strengths [37] to filter out popular keywords given that fraudulent keywords form communities with dense connectivity, while popular keywords are loosely connected with the fraudulent keywords (see Fig.4).

In detail, we build a keyword graph where the nodes are the candidate fraudulent keywords, and there is an edge between two keywords if they were searched within 2 minutes by any user. Fig.4 is a snapshot of such a graph. Edges associated with a popular keyword are probably weak ties, while fraudulent keywords are likely to form strong ties. For q_i and q_j in the query keyword graph, we use the metric overlap [37]:

$$O_{ij} = n_{ij}/((k_i - 1) + (k_j - 1) - n_{ij}),$$

to measure the tie strength of edge (q_i, q_j), where n_{ij} is the number of common neighbors of q_i and q_j, and $k_i(k_j)$ denotes the degree of query keyword $q_i(q_j)$. O_{ij} represents the proportion of q_i and q_j's common neighborhood. Consider an extreme case, if q_i and q_j are in the same clique, then $O_{ij} = 1$. On the contrary, if q_i and q_j have no common neighbors, then $O_{ij} = 0$.

We take the average of the overlap value of edges associated with q_i as the overlap value of node q_i. Fig.9 shows the CDF of the overlap over all remaining keywords. We observe a bimodal distribution that about half of the keywords have overlap values less than 0.4. These keywords indeed do not belong to densely connected communities formed by the fraudulent keywords, and should be filtered out. The rest of candidate keywords are finally taken as the identified fraudulent ones.

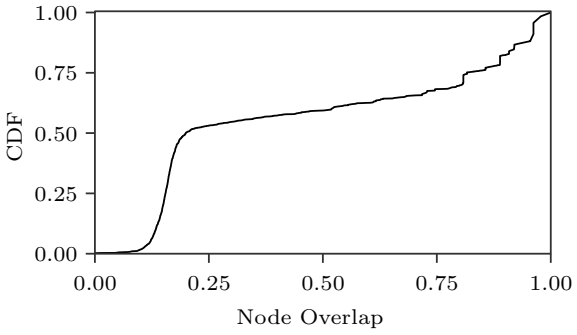

Fig.9. CDF of the node overlap.

6 Results and Analysis

In this section, we first evaluate the accuracy of DFW, and then examine the patterns of the identified fraudulent keywords and the characteristics of fraudsters, including both bots and crowd workers.

6.1 Accuracy Evaluation

We apply the proposed fraud detection approach DFW on our dataset with 57 seed fraudulent keywords provided by the examined search engine, and 627 new keywords are identified as fraudulent ones.

For the sake of evaluating the accuracy of our approach, the top 500 popular keywords are graded manually by experts from the examined search engine and the scores are provided to us as the ground truth about whether a query keyword is fraudulent or not. The top 500 popular keywords account for over 40% of search traffic as a result of heavy-tailed distribution of keyword search volumes. No more keywords are considered because manual grading requires extensive time and expensive resources. Each keyword is assigned an integer ranging from 1 to 5, reflecting the likelihood of being a fraudulent keyword (see Table 2). Generally, if the top search results of a keyword are informative and relevant to the keyword, they are more likely benign searches. Otherwise, they tend to be fraudulent. When grading the top 500 keywords, the experts take into account several factors, including the search volume of a keyword, its rank and the search results in the examined search engine. For instance, after submitting each keyword to the search engine, the web pages on the search engine results page are investigated to check if they are spam web pages. Indeed, the manual grading of keywords is actually what the examined search engine do in practice, and the search engine's intents on the fraud that they want to detect are reflected by the provided seed query keywords. We take a Chinese keyword 上海龙凤 as an example. Literally this keyword means "Shanghai Dragon Phoenix", which may refer to "twins of mixed sex in Shanghai" or the name of a Cheongsam manufacturer. However, when this keyword is graded, the top search results are pornographic forums that appear at the top of the results page by conducting fraud. Therefore, this keyword is graded 5. Finally, the top 500 keywords are carefully graded and used as the ground truth.

Table 2. Meaning of Each Fraud Score

Score	Meaning
1	Probably not a fraudulent keyword
2	Possibly not a fraudulent keyword
3	Uncertain
4	Possibly a fraudulent keyword
5	Probably a fraudulent keyword

Each of the top 500 popular keywords is evaluated by all the experts, and the average over them is taken as the final score of the keyword. Fig.10 plots the distribution of the fraud score for the top 500 popular keywords. We conservatively label those scoring above 4 as fraudulent keywords.

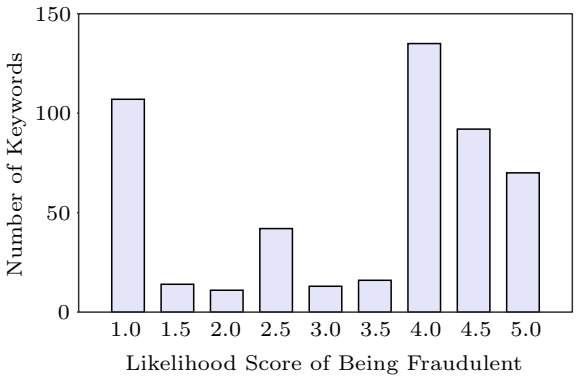

Fig.10. Fraud score histogram of the top 500 popular keywords.

We compare DFW with other three baselines that are used by the search engine routinely for fraud detection.

• *Keyword Association.* For each seed s, we set a time frame for each of its appearance, where the seed is seated in the middle. To achieve a fair comparison, the time frame is of the same size of that set in the first step of DFW (see Subsection 5.2). For keyword k that appears in the time window of s, we calculate support and confidence for pair $\{s, k\}$, where the support is defined as the ratio of the number of time frames that both s and k appear to the number of all time frames, and the confidence is the ratio of the number of time frames that both s and k appear to the number of time frames that s appears. If both support and confidence of a pair $\{s, k\}$ exceed predefined threshold values, k is detected as a fraudulent keyword.

• *Strawman Solution.* This solution has been detailed in Subsection 4.3. Specifically, we count for each candidate keyword the times it co-occurs with seeds, and label those with co-occurrence frequency exceeding a predefined threshold as fraudulent ones.

• *Label Propagation Algorithm.* This algorithm is widely used in related work [24, 27, 38]. Specifically, we compare our approach with bipartite graph propagation algorithms [24]. We construct two bipartite graphs: the user-keyword bipartite graph (U-K graph for short) and the session-keyword bipartite graph (S-K graph for short). The relationship between users and query keywords that they search forms the user-keyword bipartite graph. The user-keyword bipartite graph propagation algorithm is based on the assumption that if a user searches fraudulent keywords, the other keywords searched by this user are likely to be fraudulent. Similarly, an edge exists in the session-keyword bipartite graph if the associated keyword appeared in the associated user session, and the session-keyword bipartite graph propagation algorithm is based on the assumption that if a fraudulent keyword appears in a user session, the other keywords in this session are likely to be fraudulent. At the beginning of the algorithm, we assign a score for each query keyword. Specifically, if the keyword is a seed query keyword, we assign an initial score of 1. All the other keywords are assigned a score of 0. The algorithm is completed by multi-step iteration. An iteration contains two steps. The first step is to update the score for each user (session) with the average of the scores of its adjacent keywords. Then the second step is to update the score for each keyword with the average of the scores of its adjacent users (sessions). Specially, for a seed keyword, the score is always 1. We repeat the two steps until the scores of the keywords between two iterations change little. Finally we obtain the scores of the keywords, and label a keyword as detected fraudulent keyword if its score exceeds a threshold. Fig.11 illustrates the CDF of the scores of the query keywords after the algorithm iteration stops. We set the threshold to 0.1 according to Fig.11.

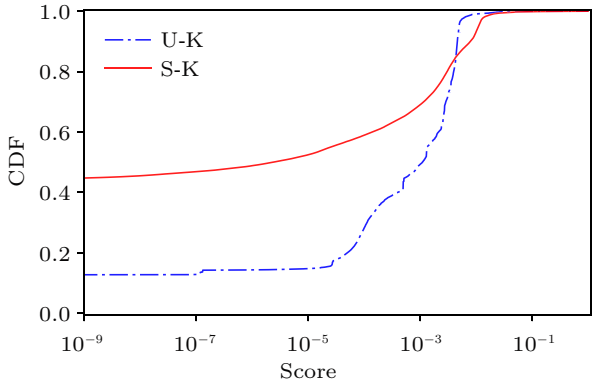

Fig.11. CDF of the scores of the query keywords after the bipartite graph propagation algorithm stops.

We use the same set of seeds for the four approaches under examination. Table 3 compares them from the perspectives of precision, recall, accuracy and $F1$-score, where precision is the fraction of true fraudulent keywords among the identified ones, recall is the fraction of identified true fraudulent keywords over the total number of true fraudulent ones, accuracy is of correctly identified keywords to the total number of keywords under consideration (i.e., 500), and $F1$-score = (2×precision×recall)/(precision+recall).

Table 3. Comparison of Five Approaches

Approach	Precision (%)	Recall (%)	Accuracy (%)	$F1$-Score (%)
K.A.	85.25	88.14	84.00	86.67
Strawman	84.52	88.81	83.80	86.61
U-K [24]	92.11	87.12	88.00	89.55
S-K [24]	84.48	99.66	89.00	91.44
DFW	99.24	88.81	93.00	93.74

Note: K.A. is short for keyword association, and U-K and S-K are the abbreviations of U-K graph propagation and S-K graph propagation, respectively.

We can see that DFW outperforms the other baseline approaches, especially in terms of precision and accuracy. DFW achieves a high precision because the fine-grained refinements we developed can mine the community structure of fraudulent keywords precisely. The recalls of the examined approaches are close except for the S-K bipartite graph propagation algorithm. The high recall of S-K bipartite graph propagation algorithm confirms the temporal correlation of fraudulent keywords that they usually co-occur with each other in a short period of time. However, the precision of S-K bipartite graph propagation algorithm is relatively low, probably because popular query keywords also co-occur in the sessions containing fraudulent keywords. F-score is a weighted average of the precision and recall, where the relative contribution of precision and recall to $F1$-score is equal. We can see that DFW achieves the highest $F1$-score.

Specifically, as for DFW, we obtain 262 true positives with 2 false positives, 33 false negatives and 203 true negatives. It is surprising to see that over half of the top 500 popular keywords are fraudulent, which illustrates the rampant fraud in search engines. A manual examination on the missed 33 fraudulent keywords revealed that they were used for a fraudulent task that was not relevant to any of the seeds that we used as input. As such, they did not co-occur frequently with any seed.

6.2 Analysis of Fraudulent Keywords

We compute each 10 minutes the search volumes of non-fraudulent and fraudulent query keywords during 30 days as shown in Fig.12, where the search volumes are normalized by the maximum of non-fraudulent search volume. We observe that fraudulent keywords are responsible for quite a portion of the total search volume. The peak of fraudulent keyword search volume reaches a half of the normal search volume. The non-fraudulent search volume reveals a periodicity of one week. However, fraudulent search volume does not show an explicit periodicity. Another interesting observation is that the fraudulent search traffic may surpass the non-fraudulent traffic at midnight. The reason is that humans need sleep, but bots do not. We also observe some bursts of the fraudulent search traffic caused by the emergence of new bags of fraudulent keywords.

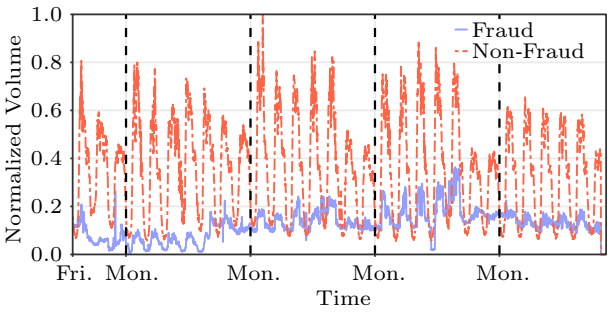

Fig.12. Search volume over 30 days.

We then investigate the temporal patterns of individual fraudulent keywords. To this end, we cluster the keywords based on the daily search volumes of each keyword using hierarchical clustering as in [39]. Specifically, each keyword is represented by a 30-dimension vector, where the i-th element is the search volume at the i-th day. Here, we consider only the 262 fraudulent keywords in the top 500 popular keyword list as they generate adequate search volumes per day. We normalize these vectors before clustering to eliminate the influence of their amplitude difference. We run the z-score normalization on each vector, and perform the agglomerative hierarchical clustering algorithm [40] on the normalized vectors. We determine the optimum number of clusters by virtue of the Davies-Bouldin index (DBI) [41, 42], which depends on neither the number of clusters nor the method of clustering.

The algorithm yields 25 clusters. Fig.13 shows the number of fraudulent keywords in each cluster. Most clusters are small while there exist four major clusters that contain most of the fraudulent keywords. To depict

different patterns of the four major clusters, we pick a typical keyword from each of them and plot the temporal variation of their search volumes in Fig.14. We see very different temporal patterns: the first cluster, targeting medical treatment, shows continuous fraudulent traffic which may be generated by bots; the second and the third clusters, targeting house decoration and car washer respectively, manifest kinds of diurnal fraud behaviors; and the fourth cluster, targeting hospital ads, shows a burst feature. Table 4 lists some examples of keywords in the Car Washer cluster.

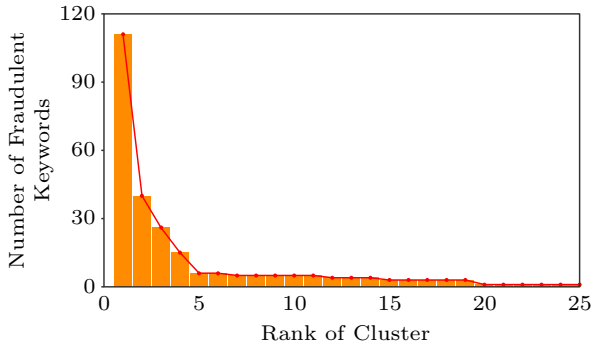

Fig.13. Number of fraudulent keywords in each cluster.

6.3 Analysis of Fraudsters

We then examine users conducting fraud in web search, i.e., fraudsters. We use the detected fraudulent keywords to compute the fraudulent search volume of each user, and then calculate the ratio of the fraudulent search volume to his/her total search volume. We group users into six bins according to their search volumes, and compute the 25th, 50th, 75th, 90th and 99th percentile ratio of each group as shown in Table 5. We also present the number of users in each group.

It is expected that most of users generate a small amount of searches and they rarely perform fraud. Nevertheless, we see some users with moderately high search volumes, which should not be bots, are indeed involved in fraud, as their fraudulent search ratios are unreasonably high. On the other hand, groups with large search volumes, which are probably bots generating automated search queries, consist of users with low fraudulent search ratios. They are either crawlers of the search engine or performing fraud with the keywords that we did not identify because the relevant seeds were missing.

We then perform breakdown analysis of the fraud-

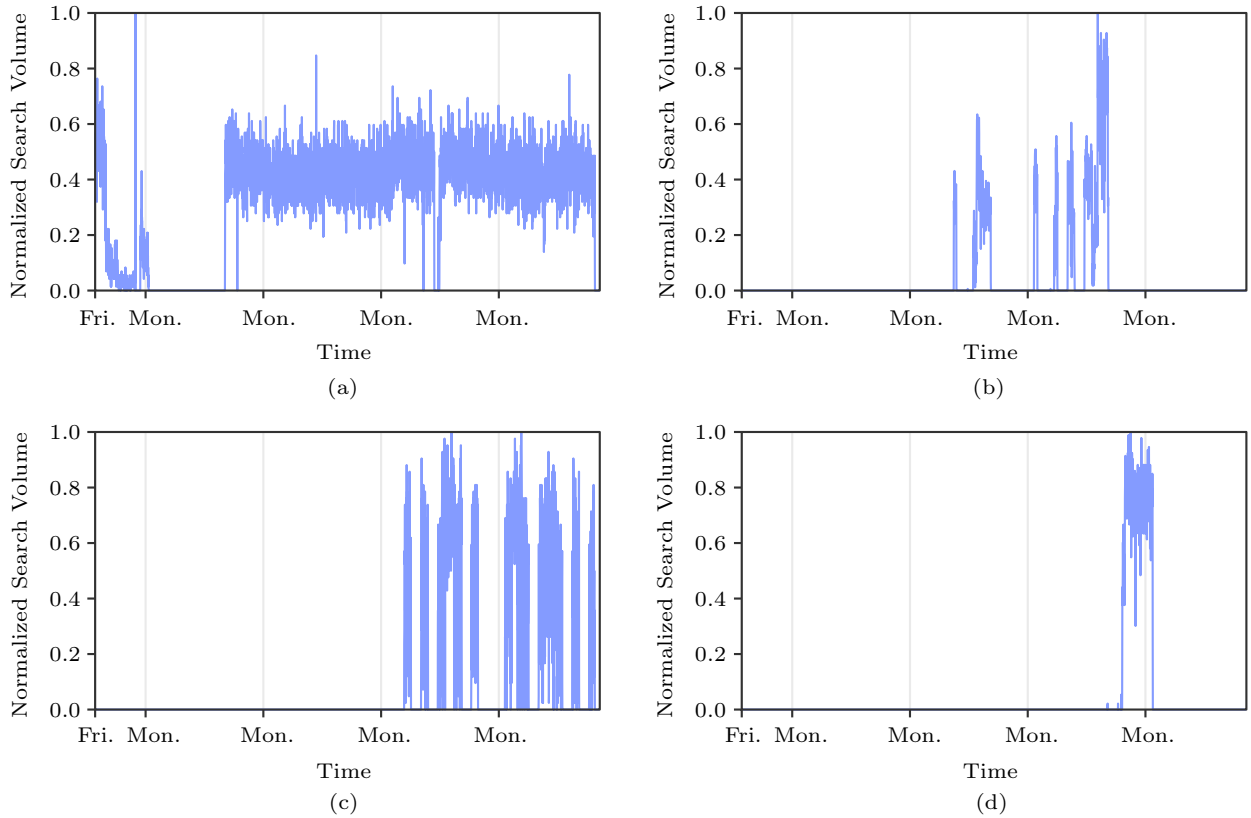

Fig.14. Four typical temporal evolution patterns of fraudulent keywords. (a) Rank 1 (Medical Treatment). (b) Rank 2 (House Decoration). (c) Rank 3 (Car Washer). (d) Rank 4 (Hospital). The corresponding targets of fraud are shown in the parentheses.

sters to gain further insights of their mixture patterns. To this end, we take into account not only the total search volumes, but also the number of unique query keywords and the fraudulent keyword ratios (i.e., the ratio of the number of fraudulent keywords to the number of total keywords). We fit the joint distribution of these three attributes of individual users to a Gaussian mixture distribution, and use AIC to guide the selection of the number of mixture components. Finally, we obtain five components, which correspond to five clusters of fraudsters as shown in Table 6.

Table 4. Examples of Keywords in the Car Washer Cluster with English Translations Listed in the Second Column

Keyword	English Translation
洗车设备价格	Car wash equipment price
免擦洗车机	No-scrub car washer
自动洗车机报价	Automatic car washer quote
龙门洗车机	Reciprocating car washer
全自动洗车机	Fully automatic car washer
无接触洗车机	Touchless car washer
洗车机报价	Car washer quote
洗车机厂家哪家好	Which car washer manufacturer is the best
无刷洗车机	Brushless car washer
洗车机品牌哪家好	Which car washer brand is the best
洗车机十大品牌	Top 10 car washer brands

Table 5. Users' Search Volumes and Fraudulent Search Ratios

Search Vol. (log10)	#Users	Fraudulent Search Ratio (Percentile)				
		25th	50th	75th	90th	99th
[0,1)	8.23 M	0.00	0.00	0.00	0.00	0.00
[1,2)	0.35 M	0.00	0.00	0.00	0.00	0.00
[2,3)	21.90 k	0.00	0.00	0.00	0.00	0.02
[3,4)	1.51 k	0.00	0.00	0.00	0.00	0.98
[4,5)	263	0.00	0.00	0.60	0.93	1.00
$\geqslant 5$	71	0.50	0.93	0.99	1.00	1.00

Note: #: number of.

Table 6. Characteristics of the 5 Clusters of Users: Number of Keywords, Frandulent Keyword Ratio and Search Volume

Cluster	#Keywords	Ratio	Volume	Label
1	16	0.10	18	Light normal users
2	70	0.03	80	Heavy normal users
3	486	0.01	488	Light crawlers
4	5 905	0.01	4 963	Heavy crawlers
5	24	0.78	5 517	Fraudsters

The five clusters of users exhibit much different patterns. The 1st and the 2nd clusters show a normal number of unique keywords with low fraudulent keyword ratios and low total search volumes. It seems that the users in these two clusters are normal users who have occasionally searched fraudulent keywords. Users in the 3rd and the 4th clusters, although having low fraudulent ratios, searched a large number of keywords with each keyword being searched once on average. They should be crawlers accidentally involved in fraud because some keywords they submitted are fraudulent ones. The 5th cluster manifests typical characteristics of fraud with a handful of keywords but a high fraudulent ratio and a large search volume.

We further distinguish between fraudsters in the 5th cluster who generated over 500 searches as bots and crowd fraud workers according to the fraudulent search ratio. If a user's fraudulent search ratio exceeds 0.8, we label this user as a bot for fraud; otherwise, if the ratio is between 0.5 and 0.8, we label this user as a human worker for fraud. In total, we obtain 215 bots and 112 workers, and thus $112/(215+112) = 34.25\%$ of fraudsters are crowd workers, which confirms our observation of the emergence of human workers in fraud. Fig.15 further depicts the temporal variations of search volumes of the bots and crowd workers. Indeed, the temporal patterns of bots and workers show significant difference, where bots generate continuously much more fraudulent searches than non-fraudulent ones, while work-

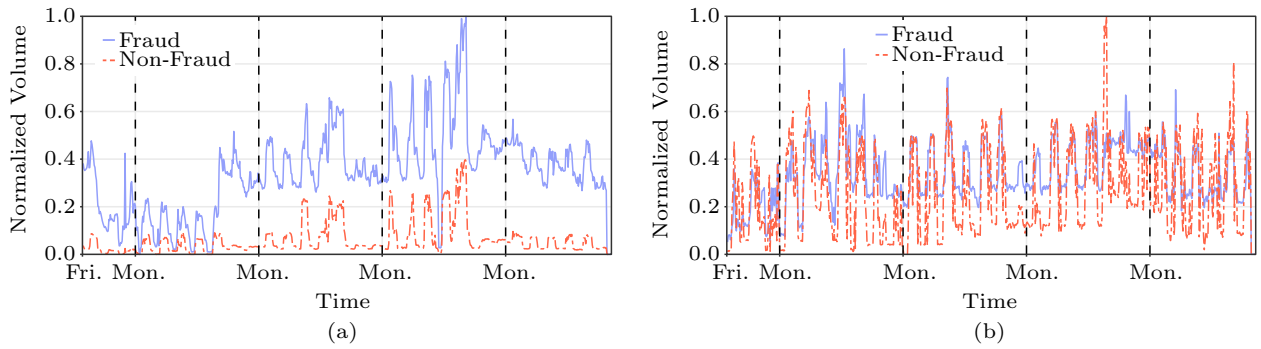

Fig.15. Search volume variations of bots and workers conducting fraud over 30 days. (a) Bots. (b) Crowd workers.

ers reveal a diurnal behavior characteristic with gentle search volumes.

Finally, we return back to examine the 80 bots that we identified in Section 4, to see whether they are identified as bots by our approach or not. The result reveals that 69 of 80 bots are present in the 4th cluster and marked as bots. The other 11 bots are not labeled as fraudsters because their fraudulent search ratios are close to 0. A further investigation reveals that two of these 11 users each submitted only one query keyword; three other users searched a bag of keywords, and the rest six users searched another bag of keywords. We miss them because the keyword that was repeatedly searched by the first two users is not included in our seed list and not relevant to any seed; and the two bags of keywords that the remaining nine users searched are not relevant to any seed in the seed list provided to us. Indeed, our approach is sensitive to the given seed list that may reflect the intents of the search engine on what kind of fraud the search engine wants to identify. We will further discuss on this issue in Section 7.

7 Discussion

Seed Provision. Our approach is sensitive to the given seed fraudulent keywords. The seeds we use are provided by the examined search engine. Indeed, they reflect the search engine's intents on the fraud they want to detect. As far as we know, search engines often hire a group of experts to manually label the fraudulent keywords. Our approach can alleviate them from tediously labeling all fraudulent keywords but just a few of them as seeds.

Design Parameters. There are several design parameters in our approach. Most key parameters can be learned empirically as what we did in Section 5. Other parameters can be tuned based on designers' purpose. For instance, parameter t_w, which controls the closeness between seeds and other keywords, can be set to a larger value in order to include more fraudulent candidates at the cost of increased false positives.

Possible Ways to Avoid Our Detection. There are two ways that can avoid our detection. The first one is to search very few fraudulent keywords repeatedly, (i.e., using a very small bag of fraudulent keywords to achieve the fraud purpose). Even though some of these keywords are in the seed list, others have the possibility to be filtered out by the keyword clustering step (see Subsection 5.3), because of the small number of keywords in the obtained clusters. The high-volume of repeated searches on few keywords is a typical pattern of bots. As such, we can merge the output of our approach with the keywords searched by bots to address this issue.

The second way is to disguise the fraudulent keywords as popular non-fraudulent ones. Suppose that a set of non-fraudulent keywords are mixed into the bags of fraudulent ones, and the set of non-fraudulent keywords change from time to time. In this way, the fraudulent keywords would have many weak ties with the non-fraudulent keywords, which lowers the average overlap of the fraudulent keywords in the fraudulent keyword graph (see Subsection 5.4), and thus they would be mistaken as popular non-fraudulent keywords. Nevertheless, this avoidance is cost-consuming because of the increased searches of huge amount of non-fraudulent keywords. Even if it happens, we can detect them by looking at whether the few high-strength ties connect to some fraudulent keywords. If so, the keywords under examination are of great suspicion.

More Sophisticated Approaches? DFW leverages off several techniques to finely exploit the community structure of fraudulent keywords for fraud detection in web search. It was developed with the engineers of the examined search engine. The first design principle is thus practical and effective. DFW achieves a very high precision, while the recall is dependent on the seeds that reflect the search engine's purpose. A more sophisticated approach would not be able to improve the precision further, but may increase the complexity greatly. Indeed, the examined search engine is about to integrate DFW to their maintenance system to alleviate the manually tedious labeling of fraudulent keywords.

Further Validation. In this paper, we focus on detecting fraud in web search, where the anomalous searches are intentionally generated to increase fake traffic for a specific purpose which is usually money-driven. Since we do not focus on detecting click fraud, our dataset does not contain click data. That is, we are not able to calculate the metrics like CTR (click-through rate). We leave the comparison with the abnormal user detection using click data as our future work. In addition, although we use the labels of the top 500 query keywords to evaluate our approach in Subsection 6.1, our approach can identify popular and unpopular fraudulent query keywords. We will further investigate the performance of our approach in practical use.

8 Conclusions

In this paper, we proposed a simple yet effective approach to detecting fraud in web search from the perspective of fraudulent query keywords. The approach, called DFW, originates from the community structure of fraudulent keywords that serve the same fraudulent target or task. DFW, taking a list of seed fraudulent keywords as input, mines the temporal correlation of query keywords to the seeds, and gradually refines the detection results. Using a dataset consisting of the web search logs for 30 days from a major search engine, we showed that DFW outperforms the three baselines that were routinely used by search engines. Specifically, the detection precision and accuracy of DFW are as high as 99% and 93% respectively. We also analyzed the characteristics of fraudulent keywords and fraudsters. Surprisingly, about 1/3 of the fraudsters exhibit fraudulent behavior other than bots.

We plan to further investigate the performance of our approach in practical use in the future.

References

[1] Jansen B J. Click fraud. *Computer*, 2007, 40(7): 85-86. DOI: 10.1109/MC.2007.232.

[2] Jansen B J, Mullen T. Sponsored search: An overview of the concept, history, and technology. *International Journal of Electronic Business*, 2008, 6(2): 114-131.

[3] Ghose A, Yang S. An empirical analysis of search engine advertising: Sponsored search in electronic markets. *Management Science*, 2009, 55(10): 1605-1622. DOI: 10.1504/IJEB.2008.018068.

[4] Fain D C, Pedersen J O. Sponsored search: A brief history. *Bulletin of the American Society for Information Science and Technology*, 2006, 32(2): 12-13. DOI: 10.1002/bult.1720320206.

[5] Gallaugher J. Information Systems: A Manager's Guide to Harnessing Technology. University of Minnesota Libraries Publishing, 2015.

[6] Shakiba T, Zarifzadeh S, Derhami V. Spam query detection using stream clustering. *World Wide Web*, 2018, 21(2): 557-572. DOI: 10.1007/s11280-017-0471-z.

[7] Stone-Gross B, Stevens R, Zarras A, Kemmerer R, Kruegel C, Vigna G. Understanding fraudulent activities in online ad exchanges. In *Proc. the 2011 ACM SIGCOMM Internet Measurement Conference*, November 2011, pp.279-294. DOI: 10.1145/2068816.2068843.

[8] Pandit S, Chau D H, Wang S, Faloutsos C. Netprobe: A fast and scalable system for fraud detection in online auction networks. In *Proc. the 16th International Conference on World Wide Web*, May 2007, pp.201-210. DOI: 10.1145/1242572.1242600.

[9] Yu F, Xie Y L, Ke Q F. SBotMiner: Large scale search bot detection. In *Proc. the 3rd ACM International Conference on Web Search and Data Mining*, February 2010, pp.421-430. DOI: 10.1145/1718487.1718540.

[10] Zhang J J, Xie Y L, Yu F, Soukal D, Lee W. Intention and origination: An inside look at large-scale bot queries. In *Proc. the 20th Annual Network and Distributed System Security Symposium*, February 2013.

[11] Zhang L F, Guan Y. Detecting click fraud in pay-per-click streams of online advertising networks. In *Proc. the 28th International Conference on Distributed Computing Systems*, June 2008, pp.77-84. DOI: 10.1109/ICDCS.2008.98.

[12] Buehrer G, Stokes J W, Chellapilla K, Platt J. Classification of automated web traffic. In *Weaving Services and People on the World Wide Web*, King I, Baeza-Yates R (eds.), Springer Verlag, 2009, pp.3-26. DOI: 10.1007/978-3-642-00570-1_1.

[13] Sadagopan N, Li J. Characterizing typical and atypical user sessions in clickstreams. In *Proc. the 17th International Conference on World Wide Web*, April 2008, pp.885-894. DOI: 10.1145/1367497.1367617.

[14] Duskin O, Feitelson D G. Distinguishing humans from robots in web search logs: Preliminary results using query rates and intervals. In *Proc. the 2009 Workshop on Web Search Click Data*, February 2009, pp.15-19. DOI: 10.1145/1507509.1507512.

[15] Tian T, Zhu J, Xia F, Zhuang X, Zhang T. Crowd fraud detection in Internet advertising. In *Proc. the 24th International Conference on World Wide Web*, May 2015, pp.1100-1110. DOI: 10.1145/2736277.2741136.

[16] Kang H W, Wang K S, Soukal D, Behr F, Zheng Z J. Large-scale bot detection for search engines. In *Proc. the 19th International Conference on World Wide Web*, April 2010, pp.501-510. DOI: 10.1145/1772690.1772742.

[17] Haidar R, Elbassuoni S. Website navigation behavior analysis for bot detection. In *Proc. the 2017 IEEE International Conference on Data Science and Advanced Analytics*, October 2017, pp.60-68. DOI: 10.1109/DSAA.2017.13.

[18] Guo Y, Shi J, Cao Z, Kang C, Xiong G, Li Z. Machine learning based cloudbot detection using multi-layer traffic statistics. In *Proc. the 21st IEEE International Conference on High Performance Computing and Communications, the 17th IEEE International Conference on Smart City and the 5th IEEE International Conference on Data Science and Systems*, August 2019, pp.2428-2435. DOI: 10.1109/HPCC/SmartCity/DSS.2019.00339.

[19] Toffalini F, Abbà M, Carra D, Balzarotti D. Google dorks: Analysis, creation, and new defenses. In *Proc. the 13th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, July 2016, pp.255-275. DOI: 10.1007/978-3-319-40667-1_13.

[20] Metwally A, Agrawal D, Abbadi A E. Duplicate detection in click streams. In *Proc. the 14th International Conference on World Wide Web*, May 2005, pp.12-21. DOI: 10.1145/1060745.1060753.

[21] Metwally A, Agrawal D, Abbadi A E. Detectives: Detecting coalition hit inflation attacks in advertising networks streams. In *Proc. the 16th International Conference on World Wide Web*, May 2007, pp.241-250. DOI: 10.1145/1242572.1242606.

[22] Immorlica N, Jain K, Mahdian M, Talwar K. Click fraud resistant methods for learning click-through rates. In *Proc. the 1st International Workshop on Internet and Network Economics*, December 2005, pp.34-45. DOI: 10.1007/11600930_5.

[23] Dave V, Guha S, Zhang Y. ViceROI: Catching click-spam in search ad networks. In *Proc. the 2013 ACM SIGSAC Conference on Computer & Communications Security*, November 2013, pp.765-776. DOI: 10.1145/2508859.2516688.

[24] Li X, Zhang M, Liu Y Q, Ma S P, Jin Y J, Ru L Y. Search engine click spam detection based on bipartite graph propagation. In *Proc. the 7th ACM International Conference on Web Search and Data Mining*, February 2014, pp.93-102. DOI: 10.1145/2556195.2556214.

[25] Nagaraja S, Shah R. Clicktok: Click fraud detection using traffic analysis. In *Proc. the 12th Conference on Security and Privacy in Wireless and Mobile Networks*, May 2019, pp.105-116. DOI: 10.1145/3317549.3323407.

[26] DeBlasio J, Guha S, Voelker G M, Snoeren A C. Exploring the dynamics of search advertiser fraud. In *Proc. the 2017 Internet Measurement Conference*, November 2017, pp.157-170. DOI: 10.1145/3131365.3131393.

[27] Wei C, Liu Y Q, Zhang M, Ma S P, Ru L Y, Zhang K. Fighting against web spam: A novel propagation method based on click-through data. In *Proc. the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, August 2012, pp.395-404. DOI: 10.1145/2348283.2348338.

[28] Haider C M R, Iqbal A, Rahman A H, Rahman M S. An ensemble learning based approach for impression fraud detection in mobile advertising. *Journal of Network and Computer Applications*, 2018, 112: 126-141. DOI: 10.1016/j.jnca.2018.02.021.

[29] Dong F, Wang H Y, Li L, Guo Y, Bissyandé T F, Liu T M, Xu G A, Klein J. FraudDroid: Automated ad fraud detection for Android apps. In *Proc. the 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, November 2018, pp.257-268. DOI: 10.1145/3236024.3236045.

[30] Halfaker A, Keyes O, Kluver D, Thebault-Spieker J, Nguyen T, Shores K, Uduwage A, Warncke-Wang M. User session identification based on strong regularities in inter-activity time. In *Proc. the 24th International Conference on World Wide Web*, May 2015, pp.410-418. DOI: 10.1145/2736277.2741117.

[31] Jones R, Klinkner K L. Beyond the session timeout: Automatic hierarchical segmentation of search topics in query logs. In *Proc. the 17th ACM Conference on Information and Knowledge Management*, October 2008, pp.699-708. DOI: 10.1145/1458082.1458176.

[32] Akaike H. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 1974, 19(6): 716-723. DOI: 10.1109/TAC.1974.1100705.

[33] Fruchterman T M J, Reingold E M. Graph drawing by force-directed placement. *Software: Practice and Experience*, 1991, 21(11): 1129-1164. DOI: 10.1002/spe.4380211102.

[34] Saramäki J, Kivelä M, Onnela J P, Kaski K, Kertesz J. Generalizations of the clustering coefficient to weighted complex networks. *Physical Review E*, 2007, 75(2): Article No. 027105. DOI: 10.1103/PhysRevE.75.027105.

[35] Schütze H, Manning C D, Raghavan P. Introduction to Information Retrieval. Cambridge University Press, 2008.

[36] Pelleg D, Moore A. X-means: Extending K-means with efficient estimation of the number of clusters. In *Proc. the 17th International Conference on Machine Learning*, June 29–July 2, 2000, pp.727-734.

[37] Onnela J P, Saramäki J, Hyvönen J, Szabó G, Lazer D, Kaski K, Kertész J, Barabási A L. Structure and tie strengths in mobile communication networks. *Proceedings of the National Academy of Sciences*, 2007, 104(18): 7332-7336. DOI: 10.1073/pnas.0610245104.

[38] Whang J J, Jung Y, Kang S, Yoo D, Dhillon I S. Scalable Anti-TrustRank with qualified site-level seeds for link-based web spam detection. In *Proc. the 2020 Web Conference*, April 2020, pp.593-602. DOI: 10.1145/3366424.3385773.

[39] Wang H D, Xu F L, Li Y, Zhang P Y, Jin D P. Understanding mobile traffic patterns of large scale cellular towers in urban environment. In *Proc. the 2015 Internet Measurement Conference*, October 2015, pp.225-238. DOI: 10.1145/2815675.2815680.

[40] Corpet F. Multiple sequence alignment with hierarchical clustering. *Nucleic Acids Research*, 1988, 16(22): 10881-10890. DOI: 10.1093/nar/16.22.10881.

[41] Davies D L, Bouldin D W. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1979, 1(2): 224-227. DOI: 10.1109/TPAMI.1979.4766909.

[42] Maulik U, Bandyopadhyay S. Performance evaluation of some clustering algorithms and validity indices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002, 24(12): 1650-1654. DOI: 10.1109/TPAMI.2002.1114856.

Dong-Hui Yang is currently a Ph.D. candidate at the Institute of Computing Technology (ICT), Chinese Academy Sciences (CAS), Beijing. He received his B.E. degree in Department of Automation from Tsinghua University, Beijing, in 2016. His research interests include the domain name system (DNS) and Internet measurement.

Zhen-Yu Li received his B.E. degree from Nankai University, Tianjin, in 2003 and his Ph.D. degree in computer architecture from Graduate School of Chinese Academy of Sciences (CAS), Beijing, in 2009. He is a professor at the Institute of Computing Technology, CAS, Beijing. His research interests include Internet measurement and networked Systems.

Xiao-Hui Wang received her M.S. degree in computer technology from Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing, in 2017. She is currently an R&D engineer of Global Energy Interconnection Research Institute Co., Ltd. (GEIRI), Beijing. Her research interests include electric artificial intelligence, data mining and modeling based on electric data and federated learning.

Kavé Salamatian is a full professor with the Université Savoie Mont Blanc, Chambérg. He was previously a reader with Lancaster University, Lancaster, U.K., and an associate professor with the University Pierre et Marie Curie, Paris, France. His main areas of research are Internet measurement and modeling and networking information theory.

Gao-Gang Xie received his Ph.D. degree in computer science from Hunan University, Changsha, in 2002. He is a professor in the Computer Network Information Center, Chinese Academy Sciences (CAS), Beijing. His research interests include Internet architecture, SDN/NFV, and Internet measurement.