Detecting Gambling Sites From Post Behaviors*

Shensi Tong*, Hanlong Zhang*, Beijun Shen*, Hao Zhong*, Yongjian Wang† and Bo Jin†
*School of Electrical Information and Electrical Engineering
Shanghai Jiao Tong University, Shanghai, China
†Key Lab of Information Network Security of Ministry of Public Security
The Third Research Institute of Ministry of Public Security, Shanghai, China
Email: wangyongjian@stars.org.cn

***Abstract*—With the rapid development of the Web, Internet gambling has become a global problem, which causes nontrivial social impacts. Despite of its prosperity, in the major countries such as United States, Russia, and mainland China, Internet gambling is explicitly prohibited, and in the most remaining countries, Internet gambling is under strict regulations. However, there are so many websites that it is rather difficult to regulate Internet gambling and rather challenging to identify them. It may introduce many false positives or false negatives, if we simply grep contents of websites with keywords. In this paper, we find that the behavior of HTTP POST is a strong indicator to detect gambling sites. Based on the finding, we propose a novel approach that detects gambling sites with mined behavior models of such sites. Furthermore, we introduce graph analysis to improve our approach. Our evaluation shows that our approach achieves high precision and recall, when it detects online gambling sites from a large number of websites.**

I. Introduction

Internet gambling has become one of the most popular and lucrative business on the Internet. Global Betting and Gaming Consultancy [1] reports that the gross market of Internet gambling is expected to reach 43 billion US dollars by 2015. The prosperity of Internet gambling draws much attention from both academies and governments [2]. A recent study [3] shows that Internet gambling is even more addictive than traditional gambling, and many researchers believe that Internet gambling leads to crime [4], poverty [5], and mental problems [6]. Although a few researchers (*e.g.*, [7]) believe that it is useless to prohibit Internet gambling, the major countries such as Unities States, Russia, and mainland China explicitly prohibit Internet gambling, and most remaining countries regulate Internet gambling strictly [8]. To regulate gambling sites effectively, a gambling-site-detection technique is urgently needed.

Researchers have proposed various approaches that detect malicious websites (*e.g.*, [9]) which can cause security problems (*e.g.*, automated downloads and executions of malware [10]). But gambling sites are not malicious websites, since they are not designed to cause security problems. Researchers also have proposed various approaches that detect pornographic sites based on their contents (*e.g.*, [11]). However, unlike pornographic sites, it is not reliable to determine gambling sites based on their contents. For example, a wiki page may mention many relevant keywords, when it introduces gambling, but it is not a gambling site. It is not the content, but the functionity that decides whether a website is a gambling site or not. To the best of our knowledge, no previous work was proposed to detect gambling sites automatically, and many research questions are still open. For example, which is the best feature to detect gambling sites? How to detect such sites effectively?

To address the above questions, we analyze various features and models. To deal with real scale detection, we use Map-Reduce framework[1] to accelerate the similarity computation. This paper makes the following major contributions:

- The first approach that mines behavior models for gambling sites and detects previously unknown gambling sites with mined models.
- A tool and two evaluations on 1TB dataset. The results show that our tool detects gambling sites effectively (with more than 90% F-scores). The results also reveal that the POST behavior of a website is the best feature to determine whether it is a gambling site or not.
- An addition evaluation on applying graph analysis to improve our approach. The results are valuable to further optimize our approach.

The paper is organized as follows. Section II shows related work. Section III describes our approach. Section IV presents our evaluations. Section V analyzes the optimization of our approach. Section VI concludes.

II. Related Work

Detecting malicious websites. Researchers have proposed various approaches that detect malicious websites. Ma *et al.* [9] propose an approach that classifies URLs, and detects the tell-tale lexical and host-based properties of malicious website URLs. Canali *et al.* [12] propose a fast filtering technique called Prophiler to detect malicious websites. Eshete *et al.* [13] propose a holistic approach that leverages static analysis, dynamic analysis, machine learning, and evolutionary searching and optimization to effectively analyze and detect malicious web pages. Sushma *et al.* [14] explore how to train classifiers that automatically identify malicious Web pages based on clues from their textual content, structural tags, page links, visual appearance and URLs. Bastian *et al.* [15] utilize modern JavaScript API's to build PhishSafe,

* Corresponding author: Yongjian Wang (wangyongjian@stars.org.cn)

[1] http://hadoop.apache.org/

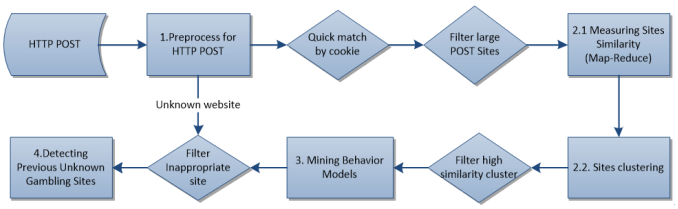


Fig. 1. The overview of our approach

a robust authentication scheme for detecting malicious sites. Ly and Bigdeli [16] propose an extendable firewall, allowing to block malicious websites when their behaviors are detected. Gambling sites are not malicious websites, since they are not designed to steal confidential information or pose other security threats. Our approach addresses a different research question from the preceding approaches.

Identifying web contents. Baykan *et al.* [17] detect languages for websites. Hu *et al.* [11] detect pornographic web pages based on their texts and images. Tsekouras and Gavalas [18] identify cultural contents from crawled web pages. Our work shows that the content alone is insufficient to detect gambling sites, and the POST behavior of a website is the best feature to detect gambling sites.

Web usage mining. It has been a hot research topic to mine usage patterns for websites [19]. The mined patterns are useful to understand user behaviours [20], to discover market opportunities [21], to improve performance of web server [22], to extract topics [23], to segment text contents [24], and to improve the design and implementation of websites [25]. Our approach mines patterns to detect gambling sites, complementing the previous approaches.

III. Approach

Fig. 1 shows the overview of our approach. It takes a large number of HTTP POSTs as the input to mine behavior models for gambling sites. Based on the mined behavior models, it further determines whether an unknown site is a gambling site or not. It consists of four steps: preprocessing HTTP POSTs, clustering sites, mining behavior models and identifying gambling sites.

A. Preprocessing HTTP POSTs

In the HTTP protocol [26], GET and POST are two common request methods of transferring the contents. In particular, GET is designed to retrieve information from a server, while POST is designed to request that a server accepts the data enclosed in the message body of the request. Since each POST causes a change in server states, POSTs offer more valuable hints on the behaviors of a website than GETs do. Typically, a POST request message consists of the following parts:

1) **Request Line.** A typical request line is "POST /a/.../script?K_1=V_1&...&K_n =V_n HTTP/1.1", where "POST" is the request method; "HTTP/1.1" is the protocol version; "/a/.../" is the request URL; "script" is the server program to be executed; and "K_n=V_n" indicates the list of parameters.
2) **Cookie in Request Header.** In a request header, a typical cookie is "K_1=V_1; ...;K_n=V_n", where "K" values are predefined. An example is "JSESSIONID=064185D5B6; NETEASE_SSN=shanghai". In some cases, JSESSIONID and NETEASE_SSN have particular means.
3) **Request Body.** A typical request body is "K_1=V_1&...&K_n=V_n". An example request body of a forum POST is "subject=Test&message=test&formhash =bbb14e19&usesig =1&posttime=138672".

The attachments may contain pictures, videos, malicious programs, but since they contain limited knowledge to determine gambling sites, we ignore them in our approach. For each POST, after our approach extracts the above three parts, it uses the following formula to calculate its hash value.

$$Hash_{post} = MD5(Script\&Keys(RequestLine) \\ \&Keys(RequestBody)) \quad (1)$$

where & denotes the delimiter, and $Keys$ is the function for extracting the variable name. Each part is shown as follows:

- Script: script in the Request Line;
- Keys(Request Line): K1, K2,...Kn in the Request Line;
- Keys(Request Body): K1, K2,...Kn in the Request Body;

To protect the privacy of users, we select only key values instead of concrete values in Eq. (1). Based on Eq. (1), our approach transforms a HTTP POST request into a fixed length of 16 bytes hash value, and stores POST requests and their hash values in a database. The benefit of using hash values is to reduce the size of POSTs, and to increase the follow-up computation speed.

B. Clustering Sites

Kumar *et al.* [27] show that website requests follow the Pareto principle, *i.e.*, on a website, 20% of functionalities are used by 80% of visitors. As a result, a few popular websites have many duplicated POSTs, while a few unpopular websites have only several POSTs. For popular websites, it is necessary to filter duplicate POSTs, and it is necessary to filter unpopular websites, since their POSTs are only several. The final equation is as follows:

$$S = \{Hash_{post1}, Hash_{post2}, ..., Hash_{postm} | m \geq \alpha_1\} \quad (2)$$

where S denotes a site; m is the number of unique $Hash_{post}$; and α_1 represents the minimum value of $Hash_{post}$. In this paper, we define the value as five. After filtering, our approach computes the Jaccard coefficient between two websites, S_i and S_j, as follows:

$$Sim(S_i, S_j) = \frac{(|S_i \cap S_j|)}{(|S_i \cup S_j|)} \quad (3)$$

where $S_i \cap S_j$ is the intersection of two websites' $Hash_{post}$ values, and $S_i \cup S_j$ is their union. In clustering, S and S' are put into the same cluster, if and only if their similarity value is higher than a predefined threshold β_1. The breadth-first algorithm is applied during the whole clustering process.

The time complexity of computing similarities between all paired websites is O(N^2). When websites are typically large in number, the calculation is quite time consuming. To address this issue, we use Hadoop to accelerate the computation. The distribution nature of clusters allows us to scale linearly.

C. Mining Behavior Models

After clustering is completed, we manually pick out gambling site clusters. As clusters are much fewer than websites, it significantly reduces the effort to identify gambling sites. Furthermore, our approach mines a behavior model for each cluster. In information retrieval, Term Frequency-Inverse Document Frequency (TF-IDF) is an established technique to measure the importance of a word to a document in a corpus [28]. We borrow the idea from information retrieval, and define the POST TF-IDF value as following:

$$TF = \frac{\text{Number of times that POST t appears in a cluster}}{\text{Total number of POSTs in the cluster}} \quad (4)$$

$$IDF = \log(\frac{\text{Total number of clusters}}{\text{Number of clusters with POST t in it}}) \quad (5)$$

$$TF - IDF = TF \times IDF \quad (6)$$

The TF-IDF value increases proportionally with the increasing frequency of a POST in a cluster, and is offset by the frequency of the POST in all clusters. After that, all the POSTs are sorted in a descending order according to their TF-IDF values. To focus on critical POSTs, our approach set the number of POSTs as the average of POSTs in each gambling cluster.

D. Detecting Previous Unknown Gambling Sites

Our approach detects previously unknown gambling sites based on the similarity between unknown gambling sites and our extracted models. If $Hash_{post}$ is shared between mined models and an unknown site, we calculate the similarity between them by Eq. (3). If the similarity value is higher than the identification threshold β_2, the unknown site is marked as a gambling site. When there exist multiple gambling models, our approach prefers the highest similarity value.

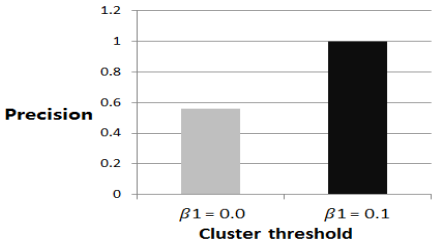
Since the corpus of known gambling sites are limited, some gambling sites may not follow any mined models. When this happens, we re-run our approach on these gambling sites to train new models that capture their POST behaviors.

IV. Evaluations

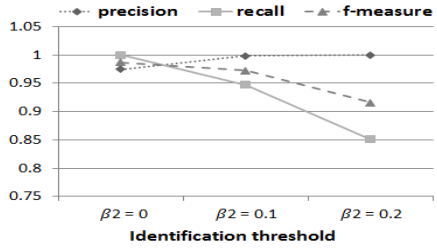
We have implemented a tool for our approach, and conducted two evaluations to address the following research questions.

- How effective is our approach to detect gambling sites (Section IV-A)?
- Which is the best feature to detect gambling sites (Section IV-B)?

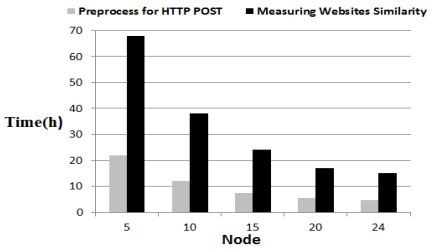
We evaluate our approach on our collected dataset which contains 4,000,000,000 HTTP POSTs of 750,000 sites. On average, each site has 20 unique HTTP POSTs. The dataset is 1TB. After preprocessing, it is reduced to 330MB. Two researchers spent a month identifying gambling sites from all



(a) Clustering



(b) Identification



(c) Performance

Fig. 2. (a) Cluster precision with different threshold β_1=0.0, 0.1. (b) Identification precision, recall and F-measure with different threshold β_2=0.0, 0.1, 0.2. (c) Performance under the clusters with different number of nodes.

the 750,000 sites manually. We calculate precision, recall, and F-measure as follows:

$$Precision = \frac{\text{Detected gambling clusters/sites}}{\text{Detected clusters/sites}} \quad (7)$$

$$Recall = \frac{\text{Detected gambling clusters/sites}}{\text{Gambling clusters/sites}} \quad (8)$$

$$F - measure = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision + Recall}} \quad (9)$$

A. Gambling Site Detection

Tuning the thresholds of our approach. Our approach has a clustering threshold (β_1) and a detection threshold (β_2). We first tried different values for β_1, and Fig. 2. (a) shows the results. The results show that when β_1 is 0.1, the precision of gambling site clusters already reaches 1. Since the accuracy of clustering seriously affects mined models, we believe that precision is more important than recall in this step, and 0.1 is already sufficient for latter steps. When β_1 is 0.1, our approach produces 1,578 clusters in total. We manually checked all the clusters, and found three gambling site clusters. From the three clusters, our approach further mines three behavior models. Tables I, II and III show the three models.

To evaluate the detection effectiveness of our approach, we introduce ten-fold cross-validation. In particular, in each iteration, we use 9/10 of the corpus to mine behavior models

for gambling sites, and use mined models to predict remaining sites. During the process, we tried different values for β_2, and Fig. 2. (b) shows the results. We find that the precision increases with the increasing of β_2, while the recall decreases with the increasing of β_2. As the recall decreases rapidly, in practice, we believe that 0.1 is the best threshold for β_2.

Tuning the performance of our approach. Our implemented tool leverages Hadoop for concurrent processing. We tried different computation nodes, and Fig. 2. (c) shows the results. We find that both the preprocessing time and the detecting time decrease with the increasing of computation nodes. However, the trend becomes smooth, when the node is more than 24. With 24 nodes, it still takes hours to process all the data. But it is acceptable since the whole process can be done offline,although there is some space for improvement.

In summary, our approach achieves high precision and recall within acceptable time limits, when it detects gambling sites from our corpus. The recall is relatively low, and we propose an optimization technique in Section V.

TABLE I
GAMBLING TEMPLATE 1

Script	Function	Frequency
gateway.php	third-party payment platform	84
chk_rule.php	check uid	73
main.php	personal homepage	68
login.php	user login	44
index.php	home page	39
order_action.php	wager page	36
today_wagers2.php	today wager amount	32
Mem2Bank2.php	bank list	26

TABLE II
GAMBLING TEMPLATE 2

Script	Function	Frequency
Mem2Bank2.php	desposit money	76
mem_cash.php	register	46
get_money2.php	withdraw money	45
pay_money_company2.php	select bank	43
mem_drawing_data2.php	add bank	16

TABLE III
GAMBLING TEMPLATE 3

Script	Function	Frequency
play.php	buy lottery	87
index.php	home page	46
registeraccount.php	register	45
shunfengdh.php	desposit money	43
transfer.php	transfer money	28

B. Feature Comparison

We compare HTTP POST with the other three features: URL, HTML, and semantic. Their definitions are as follows:
1. URL. This feature consists of lexical information and host information. In particular, lexical information includes textual properties of a given URL, while host information includes the location, the owner, and the management information of a given URL. For each URL, we focus on the length of the hostname, binary feature for content of URL (if contains numbers), the number of dots in the URL, WHOIS properties, and geographic properties.

2. HTML. This feature is extracted from HTML tags that appear in HTML code of Web pages. In particular, we focus on the frequency of each HTML tag, scripts, images, flashes, and iframes. Typically, Web pages of gambling sites contain lots of form tags where gamblers are expected to pay their bills.
3. Semantic. This feature captures textual information that is visible on Web pages. We calculate TF-IDF values for words in web pages, and use the keywords such as live dealer, lottery and casino to detect gambling site.

TABLE IV
PRECISON, RECALL, F-MEASURE WITH DIFFERENT FEATURES

Feature	Classifier	Precision	Recall	F-Measure
URL	Navie Bayes	0.656	0.615	0.635
	Random Forest	0.689	0.477	0.564
HTML	Navie Bayes	0.249	0.788	0.378
	Random Forest	0.813	0.788	0.8
Semantic	Navie Bayes	0.706	0.667	0.686
	Random Forest	0.737	0.778	0.757
URL+HTML+Semantic	Navie Bayes	0.373	0.926	0.532
	Random Forest	1	0.778	0.875
HTTP POST	Navie Bayes	0.971	1	0.985
	Random Forest	0.983	0.894	0.937

We use the above features to train classifiers with different machine learning techniques. For each technique, we use ten-fold validation to obtain their precision, recall, and F-measure, and Table IV shows the results. To save space, in Table IV, we only show the results of Navie Bayes and Random Forest. The other classifiers also produce qualitatively similar results. The results show that HTTP POST are more effective than all the other features. The URL feature works well only when gambling URLs are different from other URLs. The HTML feature works well only when other sites use different tags from gambling sites. The semantic feature introduces false negative, when non-gambling sites introduce gambling. In contrast, HTTP POST is more robust, since it reflects the functionality of gambling sites.

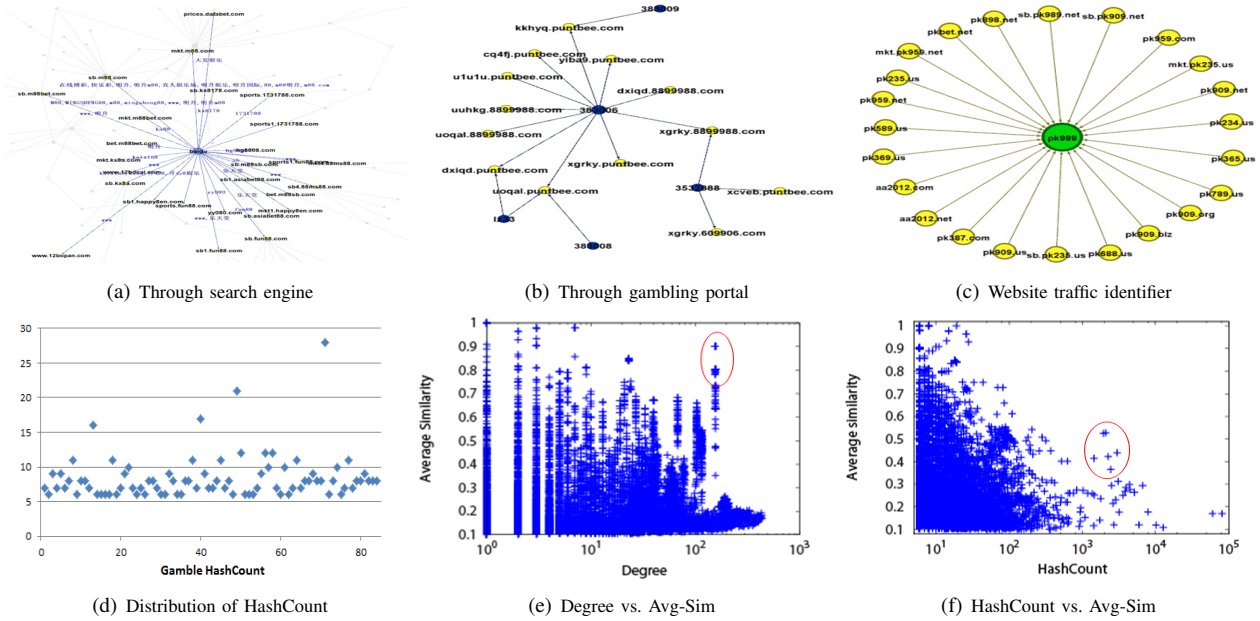
V. OPTIMIZATION VIA GRAPH ANALYSIS

We next leverage graph analysis [29] to discover the laws or anomalies in the clusters. Based on the observations, we optimize our approach with the support of Giraph [30]. Giraph is an interactive visualization and exploration platform that is designed for the analysis of dynamic and hierarchical graphs.

A. Feature

For each cluster, we use an undirected graph to denote its internal structure, where nodes denotes websites and an edge between two nodes denotes that their similarity is more than β_1. Although most features (*e.g.*, number of nodes, number of triangles, effective radius of a central node, number of neighbors, and edge weights) do not lead to any laws, the following trimmed-down features are quite useful:

- **Degree.** For a website, this feature indicates the number of its neighbors.
- **Similarity.** This feature indicates the similarity between two websites.



(a) Through search engine (b) Through gambling portal (c) Website traffic identifier

(d) Distribution of HashCount (e) Degree vs. Avg-Sim (f) HashCount vs. Avg-Sim

Fig. 3. (a),(b),(c) the distribution of utmcsr, utmctr and utmv. (d) HashCount distribution of 84 gambling sites in one cluster. (e), (f) Outliers marked with a red circle.

- **HashCount.** For a website, this feature indicates unique $Hash_{post}$.
- **Utmcsr.** This feature indicates campaign source, *i.e.*, the source that was used to enter the website.
- **Utmctr.** This feature indicates campaign terms, *i.e.*, the keywords that a visitor last used to enter the website by a search engine.
- **Utmv:** This feature indicates user-defined variables that are used to identify a site for traffic statistics.

In the above features, utmcsr, utmctr, and utmv are extracted from HTTP cookies with the support of Google analytics[2].

B. Observations

We carefully group features into pairs, and our observations are as follows:

Observation 1 - Like attracts like. According to the values of utmcsr, utmctr and utmv, we observed that like attracts like, *i.e.*, if a site is identified as a gambling site, its connected sites are likely to be gambling sites. For example, in Fig. 3. (a), the background graph denotes a part of a gambling cluster, and the front graph denotes the results from a search engine. In the front graph, the center blue point denotes the Baidu search engine, and its connected nodes denote gambling sites when we use the keyword, "ming sheng casino or online gambling" to query Baidu. From the background graph, we observe that all the sites are connected. We further investigate connected gambling sites, and we find two types of connections. First, a gambling portal is connected to many gambling sites. For example, in Fig. 3. (b), the yellow nodes denote gambling sites, and the blue nodes gambling portals. Second, a real gambling site may be connected with many fake sites. For example, Fig. 3. (c) shows that several fake gambling sites are jumped to the central real gambling site. As a result, they share the same traffic statistics identifier, "pk989". Based on the observation, we can identify gambling sites, after a gambling site is already detected.

Observation 2 - Concentration. We extracted HashCounts for one of gambling clusters. As described in Fig. 3 (d), the HashCounts of most gambling sites are around 6 to 12, instead of evenly dispersing in a certain range. The other gambling clusters' distributions also follow the law of "concentration". The observation allows us to detect gambling sites based on HashCount.

Observation 3 - Anomaly. First, we observe that in Fig. 3 (e), the average similarities of the points in the red circle are quite high. This result indicates that the contents, styles, URLs, and actions of these websites are all quite similar. We have inspected these sites, and we find that most of them are portals for different locations. For example, bj.58.com and sh.58.com belong to the same company and are both quite similar. Second, we observe that in Fig. 3 (f), HashCount decreases with the increasing of similarity. This trend is determined by Eq. (3). Nonetheless, some points in the red circle do not follow the trend. After inspection, we find that these sites are media content servers or browser synchronization servers. It can improve our approach if we filter these outliers.

C. Optimization result

We apply the observations to optimizing our approach.

- **Matching values in cookies.** According the first observation, when gambling keywords such as online gambling and casino appear in utmctr, or utmv points to illegal identity such as pk989, our approach identifies the corresponding site as a gambling site directly.

[2]http://www.google.com/analytics

- **Filtering large POST sites.** According to the third observation, our approach filters sites whose POSTs are in a great number, during our preprocess. It reduces the follow-up computation time.
- **Filtering outliers from clusters.** According to the third observation, we filter clusters whose similarities are extremely high, since these clusters are unlikely to be related to gambling clusters.
- **Filter outliers from sites.** According to the second observation, we filter sites whose HashCounts are deviation from the average of a gambling cluster.

Fig. 4 shows the results before and after our optimization. The result shows that our optimization improves 20% performance from 20 hours to 15 hours, and improves the recall from 0.95 into 0.99 without decreasing the precision.

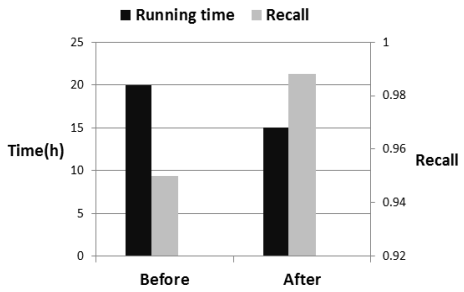


Fig. 4. Optimization result

VI. Conclusion

In this paper, we propose a novel approach that detects gambling sites based on their POST behaviors. We evaluate our approach on a large corpus, and our results show that our approach achieves both high precision and recall, and POST performs better than other features such as URL, HTML and semantic. Moreover, we leverage graph analysis to improve performance and recall. Our results show that the optimization further improves our approach.

In the future, in order to extract more detailed information about crimes, data mining and ontology technology will be taken into consideration to mine the POST behaviors and annotate fields with semantics.

Acknowledgment

This research is supported by the Opening Project (No. C14609) of Key Lab of Information Network Security of Ministry of Public Security (The Third Research Institute of Ministry of Public Security) and National Natural Science Foundation of China (Grant No. 61472242).

References

[1] G. Betting and G. Consultants, "Global gaming report. Castletown, Isle of Man," *British Isles: Author*, 2012.
[2] M. Griffiths, "Internet gambling: Issues, concerns, and recommendations," *CyberPsychology & Behavior*, vol. 6, no. 6, pp. 557–568, 2003.
[3] I. L. K. Wong and E. M. T. So, "Internet gambling among high school students in Hong Kong," *Journal of Gambling Studies*, vol. 30, no. 3, pp. 565–576, 2014.
[4] J. L. McMullan and A. Rege, "Online crime and internet gambling," *Journal of Gambling Issues*, pp. 54–85, 2010.
[5] C. Woodruff and S. R. Gregory, "Profile of Internet gamblers: Betting on the future," *UNLV Gaming Research & Review Journal*, vol. 9, no. 1, p. 1, 2012.
[6] N. M. Petry and J. Weinstock, "Internet gambling is common in college students and associated with poor mental health," *The American Journal on Addictions*, vol. 16, no. 5, pp. 325–330, 2007.
[7] A. Parke and M. Griffiths, "Why Internet gambling prohibition will ultimately fail," *Gaming Law Review*, vol. 8, no. 5, pp. 295–299, 2004.
[8] R. T. Wood, "Internet gambling: Prevalence, patterns, problems, and policy options," Ph.D. dissertation, University of Lethbridge, 2009.
[9] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: learning to detect malicious web sites from suspicious URLs," in *Proc. 15th SIGKDD*, 2009, pp. 1245–1254.
[10] N. P. P. Mavrommatis and M. A. R. F. Monrose, "All your iframes point to us," in *USENIX Security Symposium*, 2008, pp. 1–16.
[11] W. Hu, O. Wu, Z. Chen, Z. Fu, and S. Maybank, "Recognition of pornographic web pages by classifying texts and images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1019–1034, 2007.
[12] D. Canali, M. Cova, G. Vigna, and C. Kruegel, "Prophiler: a fast filter for the large-scale detection of malicious web pages," in *Proc. 20th WWW*, 2011, pp. 197–206.
[13] B. Eshete, A. Villafiorita, and K. Weldemariam, "BINSPECT: holistic analysis and detection of malicious web pages," in *Proc. 8th SecureComm*, 2012, pp. 149–166.
[14] B. Eshete, A. Villafiorita, K. Weldemariam, and M. Zulkernine, "EINSPECT: evolution-guided analysis and detection of malicious web pages," in *Proc. 37th COMPSAC*, 2013, pp. 375–380.
[15] B. Braun, M. Johns, J. Koestler, and J. Posegga, "PhishSafe: leveraging modern JavaScript API's for transparent and robust protection," in *Proc. 4th DBSec*, 2014, pp. 61–72.
[16] S. Ly and A. Bigdeli, "Extendable and dynamically reconfigurable multiprotocol firewall," *International Journal of Software Engineering and Knowledge Engineering*, vol. 15, no. 02, pp. 363–371, 2005.
[17] E. Baykan, M. Henzinger, and I. Weber, "A comprehensive study of techniques for URL-based Web page language classification," *ACM Transactions on the Web*, vol. 7, no. 1, p. 3, 2013.
[18] G. E. Tsekouras and D. Gavalas, "An effective fuzzy clustering algorithm for web document classification: A case study in cultural content mining," *International Journal of Software Engineering and Knowledge Engineering*, vol. 23, no. 06, pp. 869–886, 2013.
[19] B. Mobasher, "Web usage mining," *Web data mining: Exploring hyperlinks, contents and usage data*, vol. 12, 2006.
[20] B. Mobasher, R. Cooley, and J. Srivastava, "Automatic personalization based on web usage mining," *Communications of the ACM*, vol. 43, no. 8, pp. 142–151, 2000.
[21] A. G. Büchner and M. D. Mulvenna, "Discovering internet marketing intelligence through online analytical web usage mining," *ACM Sigmod Record*, vol. 27, no. 4, pp. 54–61, 1998.
[22] Y.-H. Wu and A. L. Chen, "Prediction of web page accesses by proxy server log," *World Wide Web*, vol. 5, no. 1, pp. 67–88, 2002.
[23] M. Xie, "Multi-granularity knowledge mining on the web," *International Journal of Software Engineering and Knowledge Engineering*, vol. 22, no. 01, pp. 1–16, 2012.
[24] P. Fragkou, "Information extraction versus text segmentation for web content mining," *International Journal of Software Engineering and Knowledge Engineering*, vol. 23, no. 08, pp. 1109–1137, 2013.
[25] B. Berendt, "Using site semantics to analyze, visualize, and support navigation," *Data Mining and Knowledge Discovery*, vol. 6, no. 1, pp. 37–59, 2002.
[26] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext transfer protocol–HTTP/1.1," 1999.
[27] C. Kumar, J. B. Norris, and Y. Sun, "Location and time do matter: A long tail study of website requests," *Decision Support Systems*, vol. 47, no. 4, pp. 500–507, 2009.
[28] A. Rajaraman and J. D. Ullman, *Mining of massive datasets*. Cambridge University Press, 2011.
[29] B. Bollobás, *Modern graph theory*. Springer Science & Business Media, 1998, vol. 184.
[30] G. Malewicz, M. H. Austern, A. J. Bik, J. C. Dehnert, I. Horn, N. Leiser, and G. Czajkowski, "Pregel: a system for large-scale graph processing," in *Proc. SIGMOD*, 2010, pp. 135–146.