2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovations

A System Framework for Efficiently Recognizing Web Crawlers

Weiping Zhu*, Jiangbo Qin*, Ruoshan Kong*, Hai Lin†, and Zongjian He‡
*School of Computer Science, Wuhan University, P. R. China
†School of Cyber Science and Engineering, Wuhan University, P. R. China
‡Center for eResearch, University of Auckland, New Zealand
Email: wpzhu@whu.edu.cn

***Abstract*—In recent years, web crawlers are widely used for collecting data from the Internet. However, they cause many problems including QoS degrading of normal visits, inaccuracy of data analysis, and business concerns about the data in the websites. It is highly demanded that there is a systematical way to recognize the web crawlers. In this paper, we propose a system framework to recognize the web crawlers and take corresponding actions for handle them. The access requests of a website are recorded by the logs, and then a machine learning approach is used to distinguish the web crawlers from normal users based on the logs. Detail components and procedures of the system framework are illustrated. Based on the system framework and approach, we implement an anti-crawler system. A twenty-days experiment show that the system can recognize most of the requests from web crawlers and have few miss detections of accesses from humans as those from web crawlers.**

***Index Terms*—Web Crawler, System Framework, Machine Learning**

I. Introduction

In recent years, data have been playing important role in many fields. How to effectively collect required data has become a hot research topic. Web crawler is an important approach to obtain public data available in the Internet. Typically, the data in social media [1], e-commerce [2], real estate websites [3], stock portals [4] can be collected through web crawlers.

Web crawlers offer convenience for people to collect data, however, cause many new problems. These problems mainly include QoS degrading, inaccuracy of data analysis, and business concerns. First, without proper restrictions, crawling behaviors generate additional work load at the server, therefore affect the QoS of normal visits. An interesting phenomena is the web access difficulties in March every year in China, because many students collect data using web crawlers for their graduate thesises [5]. Second, web crawlers deteriorate the data analysis results based on the visit of websites. Third, some data in the websites are of important commercial usage, therefore prohibited to be collected and used for any other purposes. Therefore, it is highly demanded that a technology can be developed to recognize web crawlers and block their accesses if needed. We call such technology anti-crawler technology.

Compared with rapid development of web crawlers, there are few studies about anti-crawler technology. Existing anti-crawler research mainly focus on solving specific kinds of web crawlers by using different approaches including support vector machine and Bayesian [6], [7], [8]. The solutions are mainly application-specific and difficult to be generalized to other applications. Moreover, the systems build using these approaches are not scalable enough to handle new techniques used by web crawlers, such as the distributed web crawlers, and web crawlers based on proxy servers. Therefore, a proper approach is required to guide the development of web crawler recognizing and processing.

In this paper, we develop a system framework for recognizing the visit of web crawlers at websites. The users' accesses on the websites are record by the logs. The logs are then split into several sessions that indicate the behaviors of different users. A machine learning based approach can be used in the system to distinguish web crawlers and human users. When web crawlers are detected, some countermeasures can be used for handling them. In summary, this paper offers the following contributions.

- We proposed a system framework for recognizing web crawlers. This system framework consists important functions for solving this problem. Both online processing and off-line processing are supported in the system framework.
- We proposed a machine learning approach for web crawler recognizing. The detailed procedures and important technique consideration of the approach are illustrated.
- We implemented an anti-crawler system based on the proposed system framework. This approach achieves desirable performance when recognizing web crawlers.

The rest of the paper is organized as follows: Section II reviews the related works. Section III describes an overview of the system framework, and in and Section IV we describe the machine leaning based recognition approach. The experiment results are reported in Section V. And finally Section VI concludes the paper.

978-1-5386-9380-3/18/$31.00 ©2018 IEEE
DOI 10.1109/SmartWorld.2018.00196

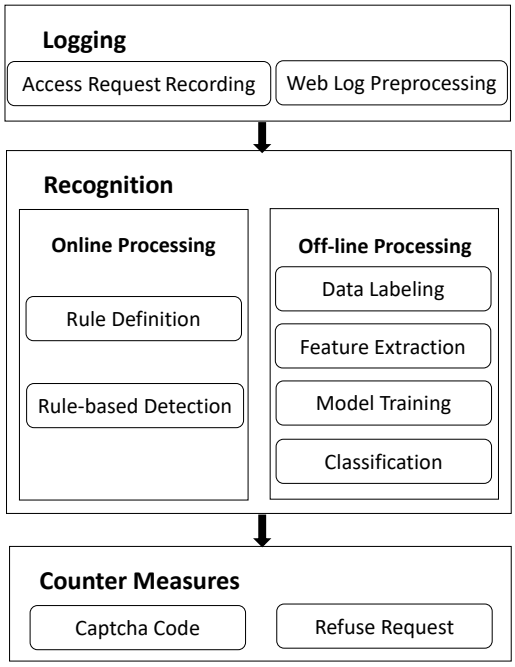

Fig. 1. System framework of an anti-crawler system

II. Related Works

Significant progress has been made on anti-crawler technology. For the machine learning based approaches, G. Jacob et al. firstly proposed the solutions to detect distributed crawlers and used large scale real dataset to evaluate their methods [6]. By extracting the features of website traffic with machine learning technology, they distinguished the actions of human users from those from web crawlers. S. Wan used heuristic rules for trainning samples in the machine learning to detect web crawlers, by using support vector machine [7]. They detected the web crawlers on the early stage and then restrained the crawlers. G. Suchacka et al. applied Bayesian methods to web crawler detection based on the features of user sessions [8]. Their experimental results showed that the classification model based on cluster analysis achieved high accurate results.

III. System Overview

We design a system framework for building the anti-crawler systems, which is shown in Figure 1. The system consists of three major modules: logging, recognition, and countermeasures. Logging is used to record the access requests using logs, recognition is used to recognize the web crawlers from normal accesses, and countermeasures are the actions taken if web crawlers are detected. We will illustrate them one by one as follows.

First, all requests accessing the web server are recorded to web logs. The logs are preprocessed including data format conversion and unrelated data filtering. Then the logs are fed into the recognition module for procession. The system mainly consists of two kinds of recognition modules, online processing module and off-line processing module. Online processing module performs real-time recognition of the users' access requests, while off-line processing module performs analysis of user behaviors as a background task.

Online processing module can be implemented by rule-based recognition. Multiple rules are defined to describe the characteristics of the access requests of web crawlers, and then the recognition can be based on the execution of the rules. Therefore, online processing module includes the rule definition and rule-based detection sub modules. The off-line processing module can be implemented by machine learning approaches. The basic idea is to generate a classier based on the training data, and then applied to the access request to be checked. Since there are many features of the access requests, a proper sub set of them are required to be extracted as the represents of the access requests. The typical machine learning approaches include the sub modules of data labeling, feature extraction, model training, and classification. The system can use online processing or off-line processing, according to the requirements of different applications. The online processing can achieve real-time detection, while off-line processing can achieve high accuracy.

After the web crawlers are recognized, several countermeasures can be taken. The typical actions include executing CAPTCHA and refusing the requests. A CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) is a type of challenge-response test used in computing to determine whether or not the user is a human [9]. If the access requests are suspected from web crawlers, such a test can be requested to be complete by the users. The other action after the web crawler's visit is detected is simply to refuse the access request, by returning an exception code or redirecting it to other websites.

Since machine learning approach is commonly used for classification, we illustrate its implementation in web crawler recognition in the following section.

IV. Machine Learning Based Recognition

We use machine learning technology to recognize the web crawlers based on the users' access logs. The workflow of processing can be seen in Fig. 2. We illustrate the approach in details as follows.

The whole processing includes two stages, the training stage and the classifications stage.In the training stage, the first step of the processing is session segmentation. A session is a unit of measurement of a user's actions taken within a period of time or with regard to completion of a task. Session segmentation is to split an access log of a web site into proper sessions [10]. Existing works of session segmentation split each of them based on time-oriented criterions such as fixed time interval between requests [11], [12], [13], adaptive time interval between requests [14], or duration of a session [15],

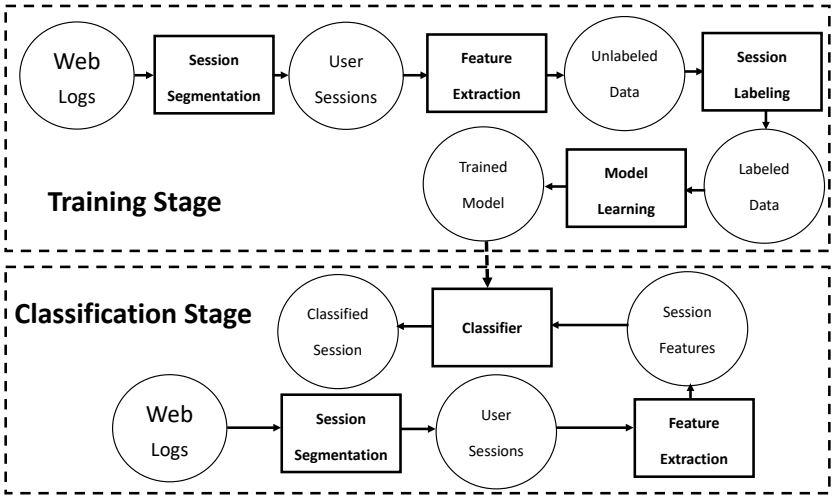

Fig. 2. Overview of the machine learning approach for recognizing web crawlers

[16]. They are applied to different scenarios and can be used in this step. The output of the session segmentation are a collection of user sessions.

Based on the user sessions, several features are required to be extracted for characterizing the access requests. There are different approaches to complete this work. We classify them as three types. The first type is using the identities information of the user sessions, including the source IP address, source physical address, source port, browser type, access time, network protocol, and others. The second type is using sthe statistic information of the logs, including percentage of requests with error codes, percentage of requests for image files, and others. The third type is the exception information in the logs.

The features are extracted and called unlabeled data in the training stage. After the session labeling process, these data will become labeled data. The session labeling process is to mark the data to an access request from humans or web crawlers. This data are usually called training data. There are several ways to perform the labeling. The users can perform it manually, or design an algorithm to automatically perform it. It it noted that the manual labeling is more accurate but quite time consuming, while automatic labeling is fast but relatively inaccurate. The balance of them is required in this step. The labeled data are fed into a machine learning algorithm to generate the trained model for later classification.

In the classification stage, an access request to be checked is performed the session segmentation and feature extraction, taking similar procedures with those in the training stage. And the resulting session features are fed into the classier for determining whether it is from web crawlers or humans.

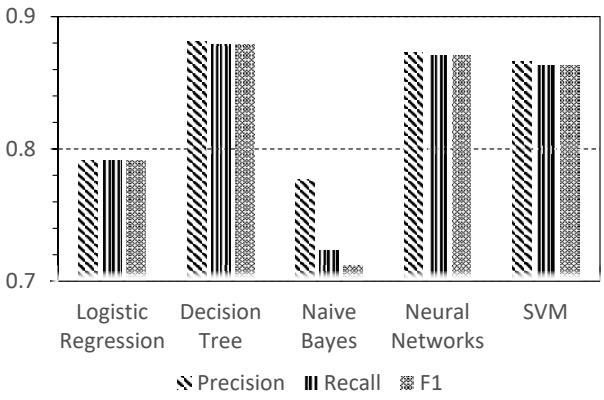

Fig. 3. Precision and recall of different machine learning approaches

V. Performance Evaluation

We follow the system framework described in section III to build an anti-crawler system. The machine learning approach is used to generate the classifier for web crawler detection. We use the access logs of the website of America National Philosophical Counseling Association [17] as the training data. In our implementation, we perform session segmentation based on session duration. Through the labeling, there are 50878 instances from humans, and 47189 instances from web crawlers, which is a roughly balanced dataset.

There are several machine learning approaches that can be used for our purpose, including logistic regression, decision tree, neural networks, naive Bayes, and SVM (Support Vector Machine). In this paper, we use Weka [18], a java machine learning library for model training, to implement these approaches. For choosing a proper approach for our problem, we

TABLE I
CONFUSION MATRIX OF WEB CRAWLER RECOGNITION

	Predicted: Web Crawler	**Predicted:Human**
Actual:Web Crawler	197	39
Actual:Human	4	187

TABLE II
RESULTS OF WEB CRAWLER RECOGNITION

	Our System
Accuracy	90.0%
Precision	98.0%
Recall	83.5%
F_1	90.2%

compare the performance of these approaches with the labeled data by conducting five groups of experiments. Ten-fold cross-validation method is used on this data to evaluate the performance. Figure 3 shows the precision and recall of each machine learning approach. According to the figure, decision tree has the largest precision and recall (0.881 and 0.879, respectively), and naive Bayes has the smallest precision and recall (0.777 and 0.723, respectively). Therefore, we adopt decision tree in our system. Another reason for choosing decision tree is that it is more interpretive than other models.

We build a website for recording users' behaviors. We use the access logs of it to evaluate our proposed approach. The time spans from March 26, 2018 to April 15, 2018. A total of 13761 access requests and 427 independent IP addresses are recorded.

The performance of our system is shown in Table I and II. Table I shows the confusion matrix, and Table II shows the results in terms of precision, recall, and F_1 measure. It can be seen that the precision of our system is 98.0%, i.e., 197 over 201 access requests classified as web crawlers are correctly, and the recall is 83.5%, i.e.,197 over 236 real web crawlers are correctly detected. We argue that this is a proper balance of them, since a relatively high precision can avoid the miss detections of humans as web crawlers and affect the normal user operations. Overall, the F1 of our system is 90.2% and the accuracy is 90.0%, which are quite desirable for web crawler recognition.

VI. CONCLUSION

In this paper, we propose a system framework for web crawler recognition. The access requests are firstly recorded by web logs and then using online or off-line processing approaches to determine whether it is from web crawlers or humans. A common implementation of the online processing is the rule based processing, and a common implementation of the offline processing is machine learning based processing. The detail procedures of the machine learning approaches for web crawlers are described. Following the proposed system framework and approach, we implemented an anti-crawler system. The results show that the system can recognize most of web crawlers and have few miss detections of the request from humans as those from web crawlers.

ACKNOWLEDGMENT

This research is supported in part by National Natural Science Foundation of China No. 61502351, Natural Science Foundation of HuBei Province, China No. 2016CFB453, Luojia Young Scholar Funds of Wuhan University No. 1503/600400001, and Chutian Scholars Program of Hubei, China.

REFERENCES

[1] X. G. Chen, S. Duan, and L. D. Wang, "Research on clustering analysis of internet public opinion," *Cluster Computing*, no. 6, pp. 1–11, 2018.
[2] J. G. Tuladhar, A. Gupta, S. Shrestha, U. M. Bania, and K. Bhargavi, "Predictive analysis of e-commerce products," in *Proceedings of Intelligent Computing and Information and Communication*. Springer Singapore, 2018, pp. 279–289.
[3] H. Horino, H. Nonaka, E. C. A. Carren, and T. Hiraoka, "Development of an entropy-based feature selection method and analysis of online reviews on real estate," in *Proceedings of 2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, Dec 2017, pp. 2351–2355.
[4] S. Sultornsanee, S. Radhakrishnan, D. Falco, A. Zeid, and S. Kamarthi, "Phase synchronization approach to construction and analysis of stock correlation network," *Procedia Computer Science*, vol. 6, no. 1, pp. 52–56, 2011.
[5] CtripTech. (2016, 06) This is enough for anti-crawler technology. [Online]. Available: https://segmentfault.com/a/1190000005840672
[6] G. Jacob, E. Kirda, C. Kruegel, and G. Vigna, "Pubcrawl: protecting users and businesses from crawlers," in *Proceedings of Usenix Conference on Security Symposium*, 2013, pp. 25–25.
[7] S. Wan, Y. Li, and K. Sun, "Protecting web contents against persistent distributed crawlers," in *Proceedings of IEEE International Conference on Communications*, 2017, pp. 1–6.
[8] G. Suchacka and M. Sobkw, "Detection of internet robots using a bayesian approach," in *Proceedings of IEEE International Conference on Cybernetics*, 2015, pp. 365–370.
[9] wikipedia. (2018, 08). [Online]. Available: https://en.wikipedia.org/wiki/CAPTCHA
[10] A. Stassopoulou and M. D. Dikaiakos, "Web robot detection: A probabilistic reasoning approach," *Computer Networks*, vol. 53, no. 3, pp. 265–278, 2009.
[11] A. S. Lalani, "Data mining of web access logs," in *Hybrid Intelligent Systems*, 2003.
[12] P. N. Tan and V. Kumar, "Discovery of web robot sessions based on their navigational patterns," *Data Mining & Knowledge Discovery*, vol. 6, no. 1, pp. 9–35, 2002.
[13] J. Srivastava, R. Cooley, M. Deshpande, and P. N. Tan, "Web usage mining: discovery and applications of usage patterns from web data," *ACM SIGKDD Explorations Newsletter*, vol. 1, no. 2, pp. 12–23, 2000.
[14] L. Zhuang, Z. Kou, and C. Zhang, "Session identification based on time interval in web log mining," *Journal of Tsinghua University*, vol. 163, pp. 389–396, 2004.
[15] M. Spiliopoulou, B. Mobasher, B. Berendt, and M. Nakagawa, "A framework for the evaluation of session reconstruction heuristics in web-usage analysis," *Informs Journal on Computing*, vol. 15, no. 2, pp. 171–190, 2003.
[16] L. D. Catledge and J. E. Pitkow, "Characterizing browsing strategies in the world-wide web," in *Proceedings of the Third International World-Wide Web Conference on Technology, Tools and Applications*. New York, NY, USA: Elsevier North-Holland, Inc., 1995, pp. 1065–1073.
[17] npcassoc.org. (2011, 07). [Online]. Available: http://npcassoc.org/log/access.log
[18] Machine Learning Group at the University of Waikato. (2018, 08). [Online]. Available: https://www.cs.waikato.ac.nz/ml/weka/