Behaviour-Based Web Spambot Detection by Utilising Action Time and Action Frequency

Pedram Hayati, Kevin Chai, Vidyasagar Potdar, Alex Talevski

Anti-Spam Research Lab (ASRL)
Digital Ecosystem and Business Intelligence Institute
Curtin University, Perth, Western Australia
<pedram.hayati, kevin.chai>@postgrad.curtin.edu.au
<v.potdar, a.talevski>@curtin.edu.au

Abstract. Web spam is an escalating problem that wastes valuable resources, misleads people and can manipulate search engines in achieving undeserved search rankings to promote spam content. Spammers have extensively used Web robots to distribute spam content within Web 2.0 platforms. We referred to these web robots as spambots that are capable of performing human tasks such as registering user accounts as well as browsing and posting content. Conventional content-based and link-based techniques are not effective in detecting and preventing web spambots as their focus is on spam content identification rather than spambot detection. We extend our previous research by proposing two action-based features sets known as *action time* and *action frequency* for spambot detection. We evaluate our new framework against a real dataset containing spambots and human users and achieve an average classification accuracy of 94.70%.

Keywords: Web spambot detection, Web 2.0 spam, spam 2.0, user behaviour

1 Introduction

Web spam is a growing problem that wastes resources, misleads people and can trick search engines algorithms to gain unfair search result rankings [1]. As a result, spam can decrease the quality and reliability of the content in the World Wide Web (WWW). As new web technologies emerge, new spamming techniques have also emerged to misuse these technologies [2]. For instance, collaborative Web 2.0 websites have been targeted by *Spam 2.0* techniques. Examples of Spam 2.0 techniques would include creating fake and attractive user profiles in social networking websites, posting promotional content in forums and uploading advertisement comments within blogs.

While similar to traditional spam, Spam 2.0 poses some additional problems. Spam content can be added to legitimate websites and therefore influence the quality of content within the website. A website that contains spam content can lose its

popularity among visitors as well as being blacklisted for hosting unsolicited content if the website providers are unable to effectively manage spam.

A Spam 2.0 technique that has been used extensively is *Web Spambots* or *Spam Web Robots*(which we refer to as *spambots*). Web robots are automated agents that can perform a variety of tasks such as link checking, page indexing and performing vulnerability assessment of targets [3]. However spambots are specifically designed and employed to perform malicious tasks i.e. to spread spam content in Web 2.0 platforms [4]. They are able to perform human-users tasks on the web such as registering user accounts, searching/submitting content and to navigate through websites. In order to counter the Spam 2.0 problem from its source, we focus our research efforts on spambot detection.

Many countermeasures have been used to prevent general web robots from the website [5-7]. However, such solutions are not sophisticated enough to deal with evolving spambots and existing literature lacks specific work dedicated to spambot detections within Web 2.0 platforms.

The study performed in this paper continues from our previous work on spambot detection [4] and presents a new method to detect spambot based on web usage navigation behaviour. The main focuses and contributions of this paper are to:

- Propose a behaviour based detection framework to detect spambots on the Web.
- Present two new feature sets that formulate spambot web usage behaviour.
- Evaluate the performance of our proposed framework with real data.

We extract feature sets from web usage data to formulate web usage behaviour of spambots and use Support Vector Machine (SVM) as a classifier to distinguish between human users and spambots. Our result is promising and shows a 94.70% average accuracy in spambot classification.

2 Spambot Detection

As previously discussed, one area of research to counter the Spam 2.0 problem is spambot detection. The main advantage of such an approach is to stop spam at the source so spambots do not continue to waste resources and mislead users. Additionally, spammers have shown that they use variety of techniques to bypass content-based spam filters (e.g. word-salad [8], Naïve Bayas poisoning [9]) hence spambot detection can be effective solution for the current situation.

The aim for spambot detection is to classify spambot user from human users while they are surfing a website. Some practical solutions such as *Completely Automated Public Turing test to tell Computers and Human Apart(CAPTCHA)[5]*, *HashCode[6]*¸ *Noune[10]*, *Form Variation [10]*, *Flood Control [10]* have been proposed to either prevent or slow down spambots activity within a website. Additionally the increasing amounts of Spam 2.0 and recent works prove that such techniques are not effective enough for spambot detection [11].

Behaviour-based spam detection has more capabilities to detect new spamming patterns as well as early detection and adaptation to legitimate and spam behaviour [12]. In this work we propose behaviour-based spambot detection method based on web usage data.

2.1 Problem Definition

We can formulate spambot detection problem in to a binary classification problem similar to the spam classification problem describe in [13]:

$$D = \{u_1, u_2, ..., u_{|U|}\} \tag{1}$$

where,
D is a dataset of users visiting a website
u_i is the i^{th} user

$$C = \{c_h, c_s\} \tag{2}$$

where,
C refers overall set of users
c_h refers to human user class
c_s refers to spambot user class

Then the decision function is

$$\phi(u_i, c_j) : D \times C \rightarrow \{0,1\} \tag{3}$$

$\phi(u_i, c_j)$ is a binary classification function, where

$$\phi(u_i, c_j) = \begin{cases} 1 & u_i \in c_s \\ 0 & otherwise \end{cases} \tag{4}$$

In spambot detection each u_i belongs to one and only one class so, the classification function can be simplified as $\phi(u_i)_{spam} : D \rightarrow \{0,1\}$.

3 Behaviour-Based Spambot Detection

3.1 Solution Overview

Our fundamental assumption is that spambots behave differently to human users within Web 2.0 applications. Hence by evaluating web usage data of spambots and human users, we believe we can identify the spambots. Web usage data can be implicitly gathered while users and spambots surf though websites . However, web usage data by itself is not effective in distinguishing spambot and human users.

Additional features need to be evaluated with web usage data in Web 2.0 applications for effective spambot detection. Therefore, we investigate two new feature sets called *Action Time* and *Action Frequency* in study spambot behaviour. An *Action* can be defined as a user set of requested web objects in order to perform a certain task or purpose. For instance, a user can navigate to the registration page in an online forum, fill in the required fields and press on submit button in order to register a new user account. This procedure can be formulated as "*Registering a user account*" action.

Actions can be a suitable discriminative feature to model user behaviour within forums but can also be extendible to many other Web 2.0 platforms. For instance, the "*Registering a user account*" action is performed in numerous Web 2.0 platforms, as users often need to create an account in order to read and write content.

In this work we make a use of *action time* and *action frequency* to formulate web usage behaviour. *Action time* is amount of time spend on doing a particular action. For instance, in "*Registering a new user account*" action, *action time* is the amount of time user spends navigating to account registration page, completing the web form and submitting the inputted information. Similarly, *action frequency*, is the frequency of doing one certain action. Additionally, if a user registers two accounts, their "*Registering a new user account*" *action frequency* is two. Section 3.2 provides a formal explanation of *action time* and *action frequency*.

It is possible to classify spambots from human user once *action time* and *action frequency* are extracted from web usage data and feed into the SVM classifier.

3.2 Framework

Our proposed framework consists of 4 main modules, which include *web usage tracking*, *data preparation*, *feature measurement* and *classification*as shown in Figure 1.

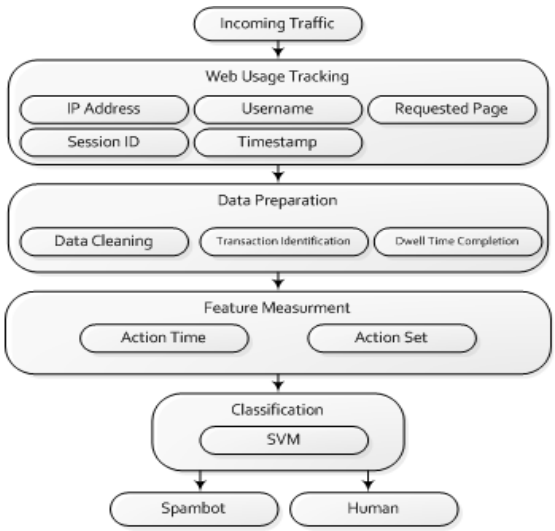

Fig. 1. Behaviour-Based Spambot detection framework

Incoming Traffic

Incoming traffic shows a user entering the website through a web interface such as the homepage of an online forum.

Web Usage Tracking

This module records web usage data including the user's *IP address*, *username*, *requested webpage URL*, *session identity*, and *timestamp*. The username and session ID makes it possible to track each user when he/she visits the system.

Conventionally, web usage navigation tracking is done through web server logs[14]. However, these logs can not specify usernames and sessions of each request. Hence we employ our own web usage tracking system developed in our previous work, *HoneySpam 2.0*[4] in order to collect web usage data.

Data Preparation

This module includes three components, which are *data cleaning*, *transaction identification* and *dwell time completion*.

Data cleaning

This component removes irrelevant web usage data from:

- Researchers who monitor the forum
- Visitors who did not create a user account
- Crawlers and other Web robots that are not spambots

Transaction Identification

This component performs tasks needed to make meaningful clusters of user navigation data[15]. We group web usage data into three levels of abstraction, which include *IP*, *User* and *Session*. The highest level of abstraction is IP and each IP address in web usage data consist multiple users. The middle level is the user level and each user can have multiple browsing sessions. Finally, the lowest level is the session level which contains detailed information of how the user behaved for each website visit.

In our proposed solution we performed spambot detection at the session level for the following reasons:

- The session level can be built and analysed quickly while other levels need more tracking time to get a complete view of user behaviour.
- The session level provides more in-depth information about user behaviour when compared with the other levels of abstraction.

Hence we define a transaction as a set of webpages that a user requests in each browsing session and extract features accordingly.

Dwell time completion

Dwell time is defined as an amount of time user spend on specific webpage in his/her navigation sequence. It can be calculated by looking at each record timestamp. Dwell time is defined as:

$$d_i' = t_{i+1} - t_i \ \ where \ 1 \le i \le |S| \qquad (5)$$

d_i' is dwell time for i[th] requested webpage in session S at time t;

In E.q.5. it is not possible to calculate dwell time for the last visited page in a session. For example, a user navigates to the last page on the website then closes his/her web browser. Hence we consider the average dwell time spent on other webpages in the same session as the dwell time for the last page.

Feature Measurement

This module extracts and measures our proposed feature sets of *action time* and *action frequency* from web usage data.

Definition 1: Set of actions (s_i)

Given a set of webpages $W = \{w_1, w_2, ..., w_{|W|}\}$, A is defined as a set of *Actions*, such that

$$A = \{a_i \mid a_i \subset W\} = \{\{w_l, ..., w_k\}\} \quad 1 \le l, k \le |W| \qquad (4)$$

Respectively s_i is defined as

$$s_i = \{a_j\} \quad 1 \le i \le |T|; \quad 1 \le j \le |A| \qquad (5)$$

s_i refers to a set of actions performed in transaction i and T is total number of transactions.

Definition 2: Action Frequency ($\overrightarrow{aF} = \langle h_1^i, ..., h_{|A|}^i \rangle$)

Action frequency ($\overrightarrow{aF}$) is a vector where h_j^i is the frequency of j th action in s_i. otherwise it is zero.

Definition 3: Action Time ($\overrightarrow{aT} = \langle d_1^i, ..., d_{|A|}^i \rangle$)

We define action time as a vector where

$$d_j^i = \begin{cases} \dfrac{\sum_{k \in a_j} d_k'}{h_j^i} & a_j \in s_i \\ 0 \quad otherwise \end{cases} \qquad (7)$$

d_j^i is a dwell time for action a_j in s_i which is equal to total amount of time spend on each webpage inside a_j. In cases that a_j occurs more than once, we divide d_j^i by the action frequency, h_j^i to calculate the average dwell time.

Classification

We employ Support Vector Machine (SVM) as our machine learning classifier. Support Vector Machine (SVM) is a machine learning algorithm designed to be robust for classification especially binary classification [16]. SVM trains by n data points or features $\{(x_1, y_1), (x_2, y_2), ...(x_n, y_n))$ and each feature comes along with class label (y_i). As mentioned in previous section there are two classes *{human,spambot}* in spambot detection, which we assign numerical value -1 and +1 to each class accordingly. SVM then tries to find an optimum hyperplane to separate two classes and maximising the margin between each class. A decision function on new data point x is define as $\phi(x) = \text{sgn}(w, x + b)$ where *w* is weight vector and *b* is bias term.

3.3 Performance Measurement

We utilised *F-Score* to measure the performance of our classification results [17]. F-Score is defined E.q. 8.

$$F = 2\frac{\text{Re}\,call \times \text{Pr}\,ecision}{\text{Re}\,call + \text{Pr}\,esision} \qquad (8)$$

where

$$\text{Re}\,call = \frac{|Detected\ Spambots|}{|Spambots|}$$

$$\text{Pr}\,ecision = \frac{|Detected\ Spambots|}{|Spambots + Humans|}$$

In the next section we discuss about experimental result of our work

4 Experimental Results

4.1 Data Set

We collected our spambot data from our previous work [4] over a period of a month. We combine this data with human data collected from an online forum with same configuration as spambot host. We removed domain specific information from both datasets. Next we combined these two dataset for experimentation. Table 1 illustrates a summary of our collected data.

Table1. Summary of collected data

Data	Frequency
# of human records	5555
# of spambot records	11039
# of total sessions	4227
# of actions	34

In feature measurement module we come up with 34 individual actions. We extract *action time* and *action frequency* from our dataset and use them separately in our classifier.

4.2 Results

We run 2 experiments on our dataset based on each feature set. We achieved an average accuracy of 94.70%, which ranges from 93.18% for *action time* to 96.23% for *action frequency*. Table 2 and Table 3 summarise the result from each experiment along with ratio of true-positives(TP) (the number of correctly classified spambots) and false-positives (FP) (number of incorrectly classified human users).

Table2. Summary of experimental results on *action time* (aT) feature set

C	TP	FP	Precision	Recall	F
c_h	0.976	0.399	0.948	0.976	0.962
c_s	0.601	0.024	0.774	0.601	0.676
Average	0.932	0.355	0.927	0.932	0.928

Table3. Summary of experimental results on *action frequency* (aF) feature set

C	TP	FP	Precision	Recall	F
c_h	0.998	0.299	0.961	0.998	0.979
c_s	0.701	0.002	0.975	0.701	0.815
Average	0.962	0.264	0.963	0.962	0.960

It is clear that *action frequency* is a slightly better classification feature to classify spambot from human users. Spambots tend to repeat certain tasks more often when compared with humans that perform a larger variety of tasks rather than focusing on specific tasks. The result of our work shows that *action time* and *action frequency* are good feature for spambot detection and therefore Spam 2.0 prevention.

5 Related Works

There has been extensive research focused on spam management and spam filtering. However, there has been little work dedicated to Spam 2.0 and spambot detection.

In the web robot detection, Tan et al. [3] propose a framework to detect unseen and camouflaged web robots. They use navigation pattern, session length and width as well as the depth of webpage coverage to detect web robots. Park et al. [7] present a malicious web robot detection method based on HTTP headers and mouse movement. However none of these works have studied spambots in Web 2.0 applications.

Yiquen et al.[18] and Yu et al. [19] utilise user web access logs to classify web spam from legitimate webpages. However the focus of their work is different from ours as they rely on user web access log as a trusted source for web spam classification. However, in this work we show that web usage logs can be obtained from both humans and spambots and such as distinction should be made.

In our previous work on HoneySpam 2.0 [4], we propose a web tracking system to track spambot data. The dataset collected in HoneySpam 2.0 is used in this work.

6 Conclusion

To the best of our knowledge, our research from [4] and this paper is the first work focused on spambot detection while conventional research has been focused on spam content detection. In this paper, we extended our previous work in spambot detection in Web 2.0 platforms by evaluating two new feature sets known as action time and action frequency. These feature sets offer a new perspective in examining web usage data collected from both spambots and human users. Our proposed framework was validated against an online forum and achieved an average accuracy of 94.70% and evaluated the performance of our framework using F-score. Future work will be focused on evaluating more feature set or combination of feature set, decrease ratio of false-positives as well as extending our work on other web 2.0 platforms to classify spambots from human users.

References

[1] Z. Gyongyi and H. Garcia-Molina, "Web spam taxonomy," in *Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web*, Chiba, Japan, 2005.

[2] P. Hayati and V. Potdar, "Toward Spam 2.0: An Evaluation of Web 2.0 Anti-Spam Methods " in *7th IEEE International Conference on Industrial Informatics* Cardiff, Wales, 2009.

[3] P.-N. Tan and V. Kumar, "Discovery of Web Robot Sessions Based on their Navigational Patterns," *Data Mining and Knowledge Discovery,* vol. 6, pp. 9-35, 2002.

[4] P. Hayati, K. Chai, V. Potdar, and A. Talevski, "HoneySpam 2.0: Profiling Web Spambot Behaviour," in *12th International Conference on Principles of Practise in Multi-Agent Systems*, Nagoya, Japan, 2009, pp. 335-344.

[5] L. von Ahn, M. Blum, N. Hopper, and J. Langford, "CAPTCHA: Using Hard AI Problems for Security," in *Advances in Cryptology — EUROCRYPT 2003*, 2003, pp. 646-646.
[6] D. Mertz, "Charming Python: Beat spam using hashcash," [Accessed: 3 Aug 09] http://www.ibm.com/developerworks/linux/library/l-hashcash.html, 2004.
[7] K. Park, V. S. Pai, K.-W. Lee, and S. Calo, "Securing Web Service by Automatic Robot Detection," *USENIX 2006 Annual Technical Conference Refereed Paper,* 2006.
[8] T. Uemura, D. Ikeda, and H. Arimura, "Unsupervised Spam Detection by Document Complexity Estimation," in *Discovery Science*, 2008, pp. 319-331.
[9] S. Sarafijanovic and J.-Y. Le Boudec, "Artificial Immune System for Collaborative Spam Filtering," in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2007)*, 2008, pp. 39-51.
[10] U. Ogbuji, "Real Web 2.0: Battling Web spam," [Accessed: 3 Aug 09] http://www.ibm.com/developerworks/web/library/wa-realweb10/, 2008.
[11] H. Abram, W. G. Michael, and C. H. Richard, "Reverse Engineering CAPTCHAs," in *Proceedings of the 2008 15th Working Conference on Reverse Engineering - Volume 00*: IEEE Computer Society, 2008.
[12] J. S. Salvatore, H. Shlomo, H. Chia-Wei, L. Wei-Jen, N. Olivier, and W. Ke, "Behavior-based modeling and its application to Email analysis," *ACM Trans. Internet Technol.,* vol. 6, pp. 187-221, 2006.
[13] Z. Le, Z. Jingbo, and Y. Tianshun, "An evaluation of statistical spam filtering techniques," *ACM Transactions on Asian Language Information Processing (TALIP),* vol. 3, pp. 243-269, 2004.
[14] R. Cooley, B. Mobasher, and J. Srivastava, "Data Preparation for Mining World Wide Web Browsing Patterns," *Knowledge and Information Systems,* vol. 1, pp. 5-32, 1999.
[15] R. Cooley, B. Mobasher, and J. Srivastava, "Web mining: information and pattern discovery on the World Wide Web," in *Tools with Artificial Intelligence, 1997. Proceedings., Ninth IEEE International Conference on*, 1997, pp. 558-567.
[16] C. Chang and C. Lin, "LIBSVM: a library for support vector machines," S. a. a. http://www.csie.ntu.edu.tw/~cjlin/libsvm, Ed., 2001.
[17] C. J. V. Rijsbergen, *Information retrieval*: Butterworths, 1979.
[18] L. Yiqun, C. Rongwei, Z. Min, M. Shaoping, and R. Liyun, "Identifying web spam with user behavior analysis," in *Proceedings of the 4th international workshop on Adversarial information retrieval on the web* Beijing, China: ACM, 2008.
[19] H. Yu, Y. Liu, M. Zhang, L. Ru, and S. Ma, "Web Spam Identification with User Browsing Graph," in *Information Retrieval Technology*, 2009, pp. 38-49.