Web Spambot Detection Based on Web Navigation Behaviour

Pedram Hayati, Vidyasagar Potdar, Kevin Chai, 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 robots have been widely used for various beneficial and malicious activities. Web spambots are a type of web robot that spreads spam content throughout the web by typically targeting Web 2.0 applications. They are intelligently designed to replicate human behaviour in order to bypass system checks. Spam content not only wastes valuable resources but can also mislead users to unsolicited websites and award undeserved search engine rankings to spammers' campaign websites. While most of the research in anti-spam filtering focuses on the identification of spam content on the web, only a few have investigated the origin of spam content, hence identification and detection of web spambots still remains an open area of research. In this paper, we describe an automated supervised machine learning solution which utilises web navigation behaviour to detect web spambots. We propose a new feature set (referred to as an action set) as a representation of user behaviour to differentiate web spambots from human users. Our experimental results show that our solution achieves a 96.24% accuracy in classifying web spambots.**

Keywords*— *Web spambot detection, Web 2.0 spam, spam 2.0, user behaviour

I. Introduction

Spammers do not only deploy their own spam webpages (known as *Web spam*) but they spread spam content over Web 2.0 applications such as online communities, wikis, social bookmarking, online discussion boards etc. [1]. Web 2.0 collaboration platforms like online discussion forums, wikis, blogs, etc. are misused by spammers to distribute spam content. This new spamming technique is called *Spam 2.0* [1]. Examples of Spam 2.0 would include spammers posting promotional threads in online discussion boards, manipulating *wiki* pages, creating fake and attractive user profiles in online community websites etc [1].

According to live reports by [2], the amount of comment spam on the Web has doubled within the past year. To date, spammers exploit new tools and techniques to achieve their purposes. An example of such a tool is a *Web spambot* (simply *spambot*), which is a type of web robot designed to spread spam content on behalf of spammers [3]. Spambots are able to crawl the web, create user accounts and contribute in collaboration platforms by spreading spam content [4]. Spambots do not only waste useful resources but also put the legitimate website in danger of being blacklisted, hence identifying and detecting spambots still remain to be an open area of research. It should be noted that Web spambots are different from spambots which are designed to harvest email address from webpages. For the sake of simplicity here we refer to web spambot as spambot.

Current countermeasures which solely focus on detection and prevention of spam content are not suitable enough to be used in a Web 2.0 environment [1]. For example, most of the content-based methods in email spam [5] or web spam techniques such as link-based detection [6], Trustrank [7], etc are not applicable in Web 2.0 environments since unlike web spam, Spam 2.0 content involves spammers contributing into legitimate website [1].

In this paper, we present a novel method to detect spambots inside Web 2.0 platforms (Spam 2.0) based on web usage navigation behaviour. The main contributions of this paper are to:

- Present a framework to detect spambots by analysing web usage data and evaluate its feasibility in combating the distribution of Spam 2.0.
- Propose an action set as a new feature set for spambot detection.
- Evaluate the performance of our proposed framework with real world data.

We make use of web usage navigation behaviour to build up our feature set and train our Support Vector Machine (SVM) classifier. Our preliminary results show that our framework is capable of detecting spambot with 96.24% accuracy.

The rest of paper is structured as follow.

- Section II gives an overview of the problems in spambot detection along with the problem definition.
- Section III presents our proposed framework for spambot detection.
- Data preparation and our experimental results are discussed in Section IV.
- We conclude the paper in Section V along with our future works.

1550-445X/10 $26.00 © 2010 IEEE
DOI 10.1109/AINA.2010.92

II. PROBLEM

Spambots mimic human behaviour in order to spread spam content. To hinder spambots activities, most websites adopt *Completely Automated Public Turing test to tell Computers and Human Apart* (CAPTCHA) which is a popular challenge-response technique to differentiate web robots from humans [8]. However, CAPTCHA is not a suitable solution for stopping spambots and it inconveniences human users. Existing research shows that by making use of machine learning algorithm even CAPTCHA based techniques can be deciphered by programming code [9-11]. Other filtering techniques are content based i.e. focusing on spam content classification rather than spambot detection [1, 3]. The formal definition of the spambot detection problem is discussed in following section.

A. Problem Definition

The problem of spambot detection is a binary classification problem that is similar to the spam classification problem described in [12]. Suppose

$$D = \{u_1, u_2, ..., u_{|U|}\} \quad (1)$$

where,

D is a dataset of users visiting a website

u_i is the i th user

$$C = \{c_h, c_s\} \quad (2)$$

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

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

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

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

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

III. PROPOSED SOLUTION

A. Solution and Overview

While most of the research in anti-spam filtering has a focused on identification of spam content on the web, only a few have investigated the source of the spam problem [1, 4, 13-15]. One way of identifying the source of spam is to study spambot behaviour. In this paper our fundamental assumption is that spambot behaviour is intrinsically different from those of humans. In order to test this assumption, we make use of web usage data from both spambots and humans. Web usage data contains information regarding the way web users navigate websites and can be implicitly collected while a user browses the website. However, it is necessary to convert web usage data in a format that

- is discriminative and reliable feature set that differentiates spambot behaviour from humans
- can be extended to other platforms of Web 2.0 applications such as wikis, blogs, online communities etc.

Hence, we propose a new feature set called an *action set* to formulate web usage data. We define an *Action* as a set of user requested webpages to achieve a certain goal. For example, in an online forum, a user navigates to a specific board then goes to the *New Thread* page to start a new topic. This user navigation can be formulated as *submit new content* action. Table I presents some example of actions for different Web 2.0 platforms. We provide a formal description of action set in Section 3.2. Our results show that the use of action sets is an effective representation of user behaviour and can be successfully used to classify spambot and human users.

TABLE I. EXAMPLES OF USER ACTIONS IN WEB 2.0 PLATFORMS INCLUDING ONLINE DISCUSSION BOARDS (I.E. FORUMS), BLOGS, WIKIS, ONLINE COMMUNITIES

Platform	Action
Online Discussion Boards	Post a topic, Reply to a topic, Contact other users, etc.
Blogs (comment)	Read posts, Read others comment, Post a comment, etc.
Wikis	Start new topic, Edit topic, Search, etc.
Online Communities	Adding new friend, Sending message, Writing comments, etc.

B. Framework

Our proposed framework consists of 3 main modules – Tracker, Extractor, and Classifier as shown in Fig. 1. Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity that you use in an equation.

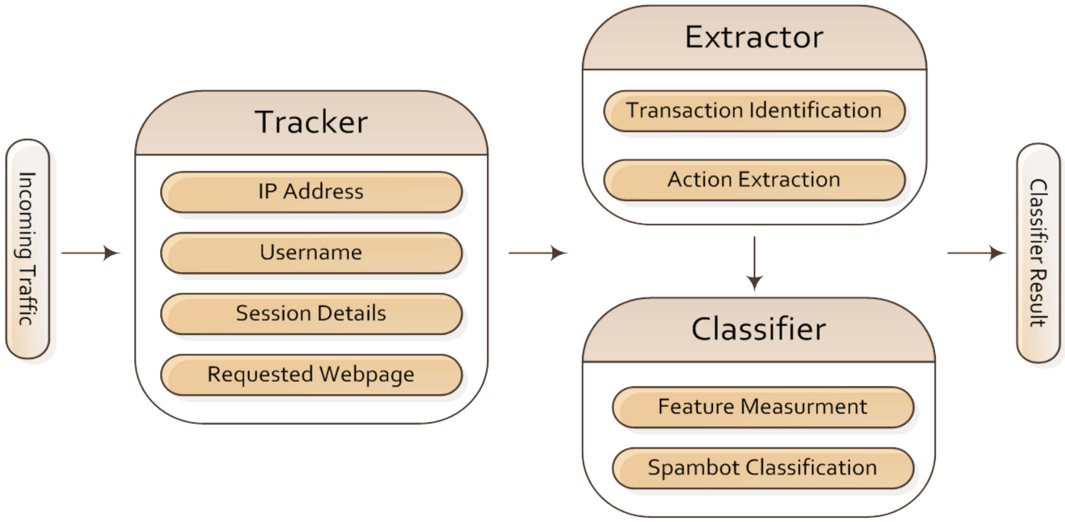


Figure 1. Architecture of Proposed Framework

*1) Incoming Traffic :*Represents a new user who enters the system via a web interface like the index page of a discussion board.

2) Tracker : This is the entry module for all incoming traffic and tracks *IP addresses*, *Usernames*, *Session Details*, and *Requested Webpages*. It starts by reading *HTTP Request* headers and extracts each of above attributes. A unique session is assigned for individual browsing session, so it is possible to track the navigation on each occasion when the user visits the website. The tracker module stores the gathered data along with corresponding username for each record.

3) Extractor : This is the second module of the proposed framework and involves two main activities which are transaction identification and action extraction.

a) Transaction Identification : Transaction identification involves creating a meaningful group of user requests [16]. In our framework, we group user requests based on three levels of abstraction. These levels range from the highest to the lowest level of abstraction (Fig. 2). At the highest level of abstraction, we have *IP address*, followed by *user* and at the lowest level of abstraction we have *session*.

The IP address which is at the highest level can contain more than one user. Inherently, at the user level, each user group can contain more than one session. The session level is the lowest level of abstraction and it contains information about user navigation data for each website visit. The user level of abstraction defines the total active time in a website spent by a user. The IP level can illustrate the behaviour of a specific host during the total amount of time that a host is connected to the website.

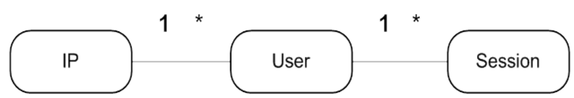


Figure 2. Levels of Web Usage Data Abstraction

In our proposed solution we chose the session abstraction level for the following reasons:

- session level can be built and analysed quickly while other levels need more tracking time to get a complete view of user behaviour.
- 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.

b) Action Extraction : Action extraction is the next activity in the Extractor module. Given a set of webpages $W = \{w_1, w_2, \ldots, w_{|W|}\}$ A is defined as a set of *Actions*, such that

$$A = \{a_i \mid a_i \subset W\} = \{\{w_l, \ldots, w_k\}\} \quad 1 \le l, k \le |W| \quad (5)$$

Respectively s_i is defined as

$$s_i = \{a_j\} \quad 1 \le i \le |U|; \quad 1 \le j \le |A| \quad (6)$$

s_i refers to a set of actions performed by user i.

4) Classifier : Classifier module is the third module in the proposed framework and involves two main activities, which are feature measurement and spambot classification.

a) Feature Measurement : Our framework extracts and measures features to build an input vector for spambot classification. In order to build the input vector, we consider each action (a_i) as a feature in our system. Suppose, there are n different actions, we represent input vector, $\vec{s'_i}$, as a bit vector (Eq. 7).

$$\vec{s'_i} = \{a_1^i, a_2^i, ..., a_n^i\} \tag{7}$$

where,

$$a_j^i = \begin{cases} 1 & a_j \in s_i \\ 0 & otherwise \end{cases} \tag{8}$$

b) Spambot Classification : In order to perform spambot classification, we used SVM on our extracted and measured features. SVM is selected as it has solid theoretical foundation, can handle high dimensional data better than most classifiers, supports non-linearity and can perform classification more accurately than other classifiers in applications involving relatively high dimensional data as in our experiment [17]. We utilise popular SVM tool known as LibSVM [18] for our spambot classification experiment.

5) Classifier Result : The result of the SVM classifier is the classification of user sessions into two classes i.e. spambots or humans.

The following section provides a detailed view of our algorithm along with the pseudo code.

C. Algorithm

Table II provides steps of our proposed framework for spambot classification. Our algorithm starts with a loop for each session, t_i. For each requested webpage in t_i, if the webpage is associated to a particular action, our algorithm looks for the next requested webpage in the session. If a group of webpages forms an action, a' , the system adds a' to the set of action performed by user k . Next, for each set of actions , s_k , the framework uses the classifier to mark the session as a spambot or human session and classifies the corresponding user k who made s_k as a spambot or human user.

TABLE II. PROPOSED METHOD ALGORITHM

Let T refer to the set of sessions.
for each $t_i \in T$ do
create new action a' .
for each $w_j \in t_i$ do
if $w_j \in a$ then
add w_j to a' .
if $a' \in A$ then
add a' to s_k .
remove all w_j in a' from t_i .
if $t_i.length > 0$ then
create new action a' .
else
continue to new w_i .
end.
end.
for each s_i do
if $\phi(s_k)_{spam} = 1$ then
mark s_i as spambot session and i as spambot user.
else
mark s_i as human session and i as human user.
end.

IV. EXPERIMENTAL RESULT

A. Data Preparation

Two major tasks in web data mining are *sessionsiation* and *user identification* from web server logs [21]. However we do not need to conduct these tasks as we track human and spambot navigation directly through the forum for each user account and for each of their sessions.

The spambot data used for our experiment is collected from our previous work [3] over a period of 60 days. Additionally, we host an online discussion forum for a human community that is under human moderation. We removed forum specific data such as the domain name, web server IP address, etc from both datasets. We grouped user navigation records based on sessions and extracted a set of actions from each dataset along with user accounts. We merge both datasets into a single dataset which contains the username and set of actions that belong to the particular user. We split the dataset into a training set (2/3) and a test set (1/3). Table III present a summery of our dataset (Table IV).

TABLE III. SUMMERY OF COLLECTED DATA

Data	Frequency
# of human records	5555
# of spambot records	11039
# of total sessions	4227 (training: 2818, test: 1409)
# of actions	34

As shown Table 5, there are 34 action navigations (34 columns) used as 34 feature and 2818 session (2818 rows) to train the SVM classifier.

TABLE IV. TRAINING DATA FORMAT USES IN THE SVM CLASSIFIER

	a_1	a_2	...	a_{34}
t_1	1	0	…	1
t_2	0	0	…	1
…	…	…	…	…
t_{2818}	0	1	…	0

B. Performance Measurement

We used *Matthew Correlation Coefficient (MCC)* method to measure the performance of our proposed framework [19]. MCC is one of the best performance measurement methods of binary classifications especially when the data among two classes of data are not balanced [20]. It considers true and false positives and returns a value between -1 and +1. If the return value is closer to +1 the classification result is better and the decision can be considered to have greater certainty. However, if the result value is close to 0 it shows the output of the framework is similar to random prediction. A result value closer to -1 shows a strong inverse ability of the classifier. MCC is defined as follows;

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (9)$$

In Eq. 9, TP is number of true positives, TN is number of true negatives, FP is number of false positives and FN in number of false negatives.

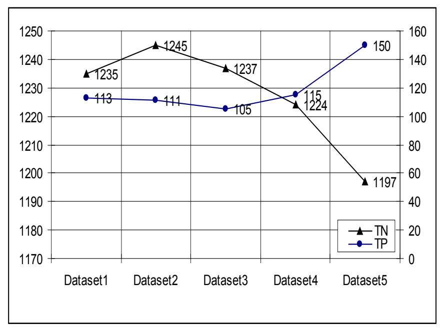
C. Results

We run our experiment on 5 randomly split training and test datasets. For each dataset we measure the accuracy and MCC value. The average accuracy 95.55% was achieved which is range from 95.03% on dataset 4 to 96.24% on dataset 2 as shown in Table V.

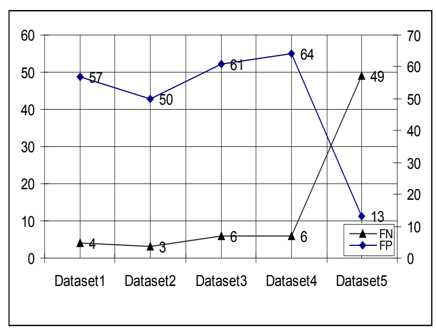
TABLE V. EXPERIMENT RESULT ON 5 RANDOM DATASETS

	Accuracy	MCC
Dataset 1	95.67%	0.780
Dataset 2	96.24%	0.801
Dataset 3	95.24%	0.751
Dataset 4	95.03%	0.757
Dataset 5	95.60%	0.809

Fig. 3 shows a comparison among the number of true negatives (correctly detected spambots), true positives (correctly detected humans), false negatives (spambots classified as humans) and false positives (humans classified as spambots) in each dataset. Although the highest accuracy was achieved by dataset 2, its MCC value is slightly lower than the MCC value corresponding to dataset 5. The reason is the number of false positives in dataset 5 is a quarter of those of dataset 2 as shown in Figure 2b.



(a)



(b)

Figure 3. **(a)** Left x Axis, True Negatives (TN), Right x Axis, True Positives (TP). **(b)** Left x Axis, False Megatives (FN) and Right x Axis, False Positives (FP)

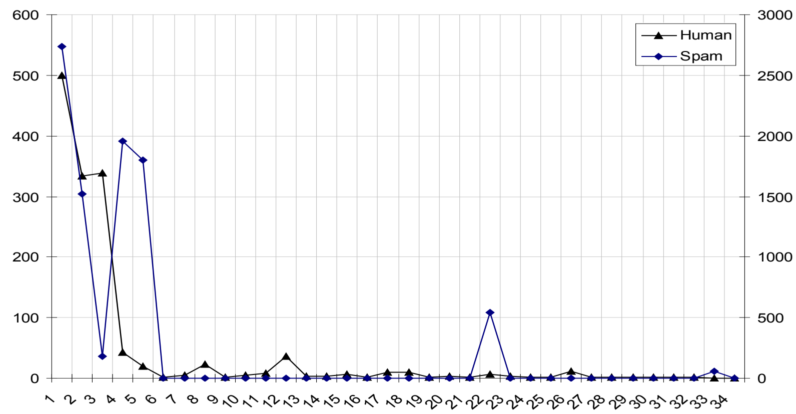


Figure 4. Frequency of Humans and Spambots. Left x Axis, Humans, Right x Axis, Spambots. Y Axis Represents Actions.

Figure 4 illustrates action frequencies of spambots and humans. A closer look at Figure 4 reveals that spambots are more active in action 5 and 6. These actions belong to *posting new thread* in online discussion forums. On the other hand humans are more active in actions 3 and 4 which belong to *read forums topics*.

V. RELATED WORK

The identification of web robots has been of interest to the research community as robot navigation on websites add noise and interferes with web usage mining of human navigation. Tan et al. [22] presented a framework to detect search engine crawlers that are camouflaged and previously unseen web robots. In their proposed framework, they use navigation patterns such as the session length, depth and width of webpage coverage and request methods. Park et al. [23] proposed a malicious web robot detection method based on the request type in *HTTP* headers along with *mouse movement*. However the focuses of these studies were not on spambot detection.

In the area of email spambots, Göbel et al. [24] introduced a *proactive* approach to monitor current spam messages inside *botnets*. They interact with email spambot controllers to collect latest email spam messages and generate *templates* and employ them inside spam filtering techniques.

In the web spam domain, Yu [25] and Liu [26] proposed a framework based on user web access data to classify spam and legitimate webpages. Their framework is based on the assumption that user navigation behaviour on spam webpages is different from legitimate webpages. However, from our study we show that one cannot assume that navigation behaviour being evaluated is from a human user but could also be from a spambot.

VI. CONCLUSION AND FUTURE WORK

While most of the research in the anti-spam filtering concentrates on the identification of spam content, we understand that the work we have proposed in this research is innovative by focusing on spambot identification to manage spam rather than analysing spam content. The importance of this approach (i.e. detecting spambots rather than spam content) is that it is a completely new approach to identify spam content and can be extended to other Web 2.0 platforms rather than only forums.

We proposed a novel framework to detect spambots inside Web 2.0 applications, which lead us to Spam 2.0 detection. We proposed a new feature set i.e. action navigations, to detect spambots. We validated our framework against an online forum and achieved 96.24% accuracy using the MCC method. In the future we will extend this work on a much larger dataset and improve our feature selection process.

REFERENCES

[1] Hayati, P., Potdar, V.: Toward Spam 2.0: An Evaluation of Web 2.0 Anti-Spam Methods 7th IEEE International Conference on Industrial Informatics, Cardiff, Wales (2009)

[2] Live-Spam-Zeitgeist: Some Stats, Akismet. [Accessed online by May 2009] http://akismet.com/stats/ (2009)

[3] Hayati, P., Chai, K., Potdar, V., Talevski, A.: HoneySpam 2.0: Profiling Web Spambot Behaviour. 12th International Conference on Principles of Practise in Multi-Agent Systems, Vol. 5925. Lecture Notes in Artificial Intelligence, Nagoya, Japan (2009) 335-344

[4] Webb, S., Caverlee, J., Pu, C.: Social Honeypots: Making Friends with a Spammer Near You. Proceedings of the Fifth Conference on Email and Anti-Spam (CEAS 2008), Mountain View, CA (2008)

[5] Hayati, P., Potdar, V.: Evaluation of Spam Detection and Prevention Frameworks for Email and Image Spam - A State of Art. The 2nd International Workshop on Applications of Information Integration in Digital Ecosystems (AIIDE 2008), Linz, Austria (2008)

[6] Qingqing, G., Torsten, S.: Improving web spam classifiers using link structure. Proceedings of the 3rd international workshop on Adversarial information retrieval on the web. ACM, Banff, Alberta, Canada (2007)

[7] Zolt, n, G., ngyi, Hector, G.-M., Jan, P.: Combating web spam with trustrank. Proceedings of the Thirtieth international conference on Very large data bases - Volume 30. VLDB Endowment, Toronto, Canada (2004)

[8] von Ahn, L., Blum, M., Hopper, N., Langford, J.: CAPTCHA: Using Hard AI Problems for Security. Advances in Cryptology — EUROCRYPT 2003 (2003) 646-646

[9] Abram, H., Michael, W.G., Richard, C.H.: Reverse Engineering CAPTCHAs. Proceedings of the 2008 15th Working Conference on Reverse Engineering - Volume 00. IEEE Computer Society (2008)

[10] Baird, H.S., Bentley, J.L.: Implicit CAPTCHAs. Proceedings SPIE/IS&T Conference on Document Recognition and Retrieval XII (DR&R2005), San Jose, CA (2005)

[11] Chellapilla, K., Simard, P.: Using Machine Learning to Break Visual Human Interaction Proofs (HIPs). NIPS (2004)

[12] Le, Z., Jingbo, Z., Tianshun, Y.: An evaluation of statistical spam filtering techniques. ACM Transactions on Asian Language Information Processing (TALIP) 3 (2004) 243-269

[13] Nitin, J., Bing, L.: Opinion spam and analysis. Proceedings of the international conference on Web search and web data mining. ACM, Palo Alto, California, USA (2008)

[14] Zinman, A., Donath, J.: Is Britney Spears spam. Fourth Conference on Email and Anti-Spam, Mountain View, California (2007)

[15] Uemura, T., Ikeda, D., Arimura, H.: Unsupervised Spam Detection by Document Complexity Estimation. Discovery Science (2008) 319-331

[16] Cooley, R., Mobasher, B., Srivastava, J.: Data Preparation for Mining World Wide Web Browsing Patterns. Knowledge and Information Systems 1 (1999) 5-32

[17] Liu, B.: Web Data Mining. Springer Berlin Heidelberg (2007)

[18] Chang, C., Lin, C.: LIBSVM: a library for support vector machines. In: http://www.csie.ntu.edu.tw/~cjlin/libsvm, S.a.a. (ed.): (2001)

[19] Matthews, B.W.: Comparison of the predicted and observed secondary structure of T4 phage lysozyme. Biochim Biophys Acta 405 (1975) 442-451

[20] Baldi, P., Brunak, S., Chauvin, Y., Andersen, C.A.F., Nielsen, H.: Assessing the accuracy of prediction algorithms for classification: an overview. Bioinformatics 16 (2000) 412-424

[21] Cooley, R., Mobasher, B., Srivastava, J.: Web mining: information and pattern discovery on the World Wide Web. Tools with Artificial Intelligence, 1997. Proceedings., Ninth IEEE International Conference on (1997) 558-567

[22] Tan, P.-N., Kumar, V.: Discovery of Web Robot Sessions Based on their Navigational Patterns. Data Mining and Knowledge Discovery 6 (2002) 9-35

[23] Park, K., Pai, V.S., Lee, K.-W., Calo, S.: Securing Web Service by Automatic Robot Detection. USENIX 2006 Annual Technical Conference Refereed Paper (2006)

[24] Jan, G., bel, Thorsten, H., Philipp, T.: Towards Proactive Spam Filtering (Extended Abstract). Proceedings of the 6th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment. Springer-Verlag, Como, Italy (2009)

[25] Yu, H., Liu, Y., Zhang, M., Ru, L., Ma, S.: Web Spam Identification with User Browsing Graph. Information Retrieval Technology (2009) 38-49

[26] Yiqun, L., Rongwei, C., Min, Z., Shaoping, M., Liyun, R.: Identifying web spam with user behavior analysis. Proceedings of the 4th international workshop on Adversarial information retrieval on the web. ACM, Beijing, China (2008)

Hayati, P., Chai, K., Potdar, V., Talevski, A.: HoneySpam 2.0: Profiling Web Spambot Behaviour. 12th International Conference on Principles of Practise in Multi-Agent Systems, Vol. 5925. Lecture Notes in Artificial Intelligence, Nagoya, Japan (2009) 335-344

Hayati, P., Potdar, V.: Toward Spam 2.0: An Evaluation of Web 2.0 Anti-Spam Methods 7th IEEE International Conference on Industrial Informatics, Cardiff, Wales (2009)

Hayati, P., Potdar, V.: Spammer and Hacker, Two Old Friends. 3rd IEEE International Conference on Digital Ecosystems and Technologies (IEEE-DEST 2009), Istanbul, Turkey (2009)

Hayati, P., Potdar, V.: Evaluation of Spam Detection and Prevention Frameworks for Email and Image Spam - A State of Art. The 2nd International Workshop on Applications of Information Integration in Digital Ecosystems (AIIDE 2008), Linz, Austria (2008)