A Density Based Clustering Approach for Web Robot Detection

Mahdieh Zabihi
M.Sc. Student of
Computer Engineering,
Imam Reza International
University,
Mashhad, Iran
m.zabihi@imamreza.ac.ir

Majid Vafaei Jahan
Department of Computer
Engineering,
Islamic Azad University,
Mashhad, Iran
vafaeiJahan@mshdiau.ac.ir

Javad Hamidzadeh
Faculty of Computer
Engineering and
Information Technology,
Sadjad University of
Technology, Mashhad, Iran
j_hamidzadeh@sadjad.ac.ir

*Abstract***— Distinction between humans and web robots, in terms of computer network security, has led to the robot detection problem. An exact solution for this issue can preserve web sites from the intrusion of malicious robots and increase the performance of web servers by prioritizing human users. In this article, we propose a density based method called DBC_WRD (Density Based Clustering for Web Robot Detection) to discover the traffic of web robots on two large real data sets. So, we assume the visitors as the spatial instances and introduce two new features to describe and distinguish them. These attributes are based on the behavioral patterns of web visitors and remain invariant over time. By focusing on one of the disadvantages of DBSCAN as the density based clustering algorithm used in this paper, we just utilize 4 features to reduce the dimensions. According to the supervised evaluations, DBC_WRD can have the 96% of Jaccard metric and produce two clusters which have the entropy and purity rates of 0.0215 and 0.97, respectively. Furthermore, the comparisons show that from the standpoint of clustering quality and accuracy, DBC_WRD performs better than state-of-the-art algorithms. Finally, it can be concluded that some non-malicious popular web robots, through imitating the human's behavior, make it difficult to be identified.**

Keywords- *density based Clustering; DBSCAN; data mining; web robot detection; behavioral patterns of web visitors*

I. INTRODUCTION

Internet, one of the most important technologies these days, is a massive information repository and a new medium for communication and collaboration. Undoubtedly, to manage and update this repository and gain some knowledge from the information, appropriate solutions are necessary. Web robots as one of these solutions, send requests to web servers and analyze the results received to fulfill their specific purposes. Well-behaved or non-malicious robots with useful aims, such as collecting the statistics about the web structure, site maintenance and checking for broken hyperlinks, are active researchers in the World Wide Web. Unlike these autonomous systems, some robots with malicious goals, such as click fraud, harvesting Email addresses, collecting business intelligence and DDoS [1] attacks, threaten the security of web servers. From the standpoint of occupying the network bandwidth and reducing the performance of web servers, malicious and non-malicious web robots are similar to each other. Negligence, failure to follow instructions on how to design a robot, and changing its characteristics, in order to imitate human's behavior are some obstacles for robot detection.

Most related studies, in this field, have used supervised learning methods to classify visitors and detect the traffics of web robots. One of the latest of such researches is the first study which employs unsupervised leaning and applies the SOM and ART2 clustering algorithms to detect malicious web robots [7].

In this paper, we propose a density based method called DBC_WRD (Density Based Clustering for Web Robot Detection) to distinguish the robot traffics of two real large data sets. This method utilizes DBSCAN (Density Based Spatial Clustering of Applications with Noises) as the density based clustering algorithm.

Due to the dynamism and diversity in behavior and operation of web robots, using the features that are well able to distinguish them is important. Moreover, the DBSCAN algorithm is sensitive to the number of dimensions. So, the large number of attributes makes it difficult to find the proper value for one of the input parameters of this algorithm. Therefore, we just utilize 4 features two of which are the new attributes proposed in this paper. These new features are based on the behavioral patterns of web visitors and remain invariant over time. We know that if we can demonstrate the efficiency of a clustering algorithm by supervised evaluation metrics, we will be able to have more confidence in that clustering method. Therefore, we identify the labels of instances (web visitors) and substantially, we utilize some popular supervised metrics to evaluate the DBSCAN algorithm in experimental section.

Results show that by solving the curse of dimensionality problem and choosing the appropriate features, DBC_WRD can be very effective to distinguish robots from human users. Furthermore, from the standpoint of clustering quality and accuracy, DBC_WRD performs better than state-of-the-art algorithms. Finally, it can be concluded that some non-malicious popular robots, through acting like humans, make it difficult to be identified.

The remainder of the paper is organized as follows: In section II, a survey of web robot detection is presented. Section III formulates the web robot detection problem

[1] Distributed Denial of services

while section IV introduces DBC_WRD as the proposed algorithm for web robot detection. In section V, the experimental results are discussed and eventually, section VI finalizes our conclusions and remarks.

II. RELATED WORKS

In this section, we divide the related researches into two categories: several studies based on supervised learning techniques for web robot detection and the others that utilize unsupervised learning methods for more general purposes of web log analysis.

At the beginning, we address some samples of the first category. In one of the earliest of such studies [14], the authors use a new approach to extract web sessions and introduce 25 new features to distinguish humans from web robots and utilize C4.5 decision tree to classify web users. They apply their method on an academic access log collected over a period of a month in year 2001 and achieve very accurate web robot detection rates. In 2005, Bomhardt et al. develop a web log pre-processing tool called RDT and use neural network and decision tree to detect web robots. They use two log files to evaluate their method, one from an educational web site and another from an online shop [3]. In [1], the authors present a Bayesian approach to crawler detection and declare a dynamic threshold for session identification. Also, they compare their results to the results obtained with the decision tree and achieve very high recall and precision values in web robot detection. In our earlier work [11], we have proposed a fuzzy inference system based on decision trees to detect the traffic of web robots on a real web server. The proposed system uses a decision tree to fuzzify features which describe web users and facilitate designing of fuzzy inference system.

In contrast to these studies based on supervised learning, several studies apply unsupervised clustering algorithms for more general web log analysis. For example, Hiltunen et al. use the number and sequence of web pages visited by human users, to achieve an automatic demographic-based classification of these visitors, by utilizing the kohonen's self-organizing-map (SOM) [16]. In 2008, Petrilis et al. propose a method based on SOM that utilizes both content and context mining clustering techniques to group human visitors thematically and help them identify relevant information more quickly [6].

For the first time, in 2013, Stevanovic et al. use SOM and ART2 clustering methods for web robot detection, because previous studies, based on classification techniques, suffer from the untrustworthiness of the conventional methods used for session labeling. So, the authors use the labels of sessions only for the supervised evaluation of their proposed methods and categorize web visitors into four classes: malicious and well-behaved web robots, humans and unknown visitors [7].

According to the reasons mentioned, we postpone the use of labels of sessions to the evaluation operations and utilize an unsupervised learning technique. In the comparison of the previous works, the contribution of our research is twofold:

- This paper is the first study which utilizes a popular clustering algorithm used for spatial databases, DBSCAN, in the field of web robot detection. This algorithm has received a lot of attention in KDD (Knowledge Discovery in Databases) and several studies have found it to be very effective at clustering the data sets of significantly more than just a few thousand objects [12]. Moreover, DBSCAN can solve the common problems included when using other classification techniques for such data sets. It can find complicated cluster shapes in a reasonable amount of time, with only two assigned input parameters and also supports the user in determining the values for both of them. These potentials encouraged us to organize web users as the spatial instances and utilize the ability of the DBSCAN algorithm in clustering them.
- In this paper, we propose two new features to describe web visitors and distinguish robots from human users. These attributes are based on the navigational patterns of visitors and the resources requested by them. We have tried to select these features in a way that does not change over time, in order to keep the techniques effective across server domains and in the face of evolving robot traffics.

III. WEB ROBOT DETECTION PROBLEM

This problem tries to identify the traffic of web robots, according to the requests stored in an access log file of a web server [5]. In such a file, for each request received, some information is stored as an individual record. A session is a collection of all requests from a user during a single visit. In this paper, we concentrate on web sessions rather than individual records and formalize the session identification as follow:

Given a set of HTTP records $R= \{r_1,r_2,...,r_n\}$; find the set of sessions $S= \{s_1,s_2,...,s_m\}$, such that:

$$S = \{s_i \mid s_i \subseteq R, \bigcup_{i=1}^{m} s_i = R, \bigcap_{i=1}^{m} s_i = \varnothing\} \quad (1)$$

Where, n shows the number of http records stored in the original access log file and m is the number of all sessions in S. Finally, we declare f as a function to detect the type of each session and assign 0 to it, if it belongs to human visitors and vice versa.

$$f : S \rightarrow \{0,1\}, f(s_i) = \begin{cases} 0 & s_i \text{ is Human} \\ 1 & s_i \text{ is Robot} \end{cases} \quad (2)$$

In this paper, the f function is the DBSCAN clustering algorithm.

IV. THE PROPOSED METHOD

In this section, we present the method proposed for web robot detection. Fig. 2 shows the flowchart of the general procedure used which contains two steps: preprocess and clustering. The preprocessing step prepares the necessary backgrounds for the clustering which uses the DBSCAN algorithm. In the following, all basic essentials of the preprocessing step are described in details.

As previously mentioned, in our study, the clustering instances are the sessions belonging to web visitors. As a result, we should analyze the web server access logs to extract all sessions and define some features (attributes) to describe them. On the other hand, we need to identify the labels of sessions and utilize them for supervised evaluations of clustering algorithm. We implement a java-based log analyzer to preprocess the web server access logs.

```
type requestRecord {
    ip: string , time: string, request: string,
    status: string, size: integer, referrer: string,
    agent: string, method: string

}
type session {
    label: boolean, LastTime: string, list: array of
    requestRecord
}
// LastTime: the time of last requestRecord assigned to the
session.
```

```
The session identification procedure:

S: set of sessions found.
R: set of request records ordered by time (ascending).
thre: the static threshold for session identification. (thre =30
minutes).

for each $r \in R$ do
  for each $s \in S$ do
    if ( $r.time - s.LastTime > thre$ ) then
      close s;
    else
      if (containsIP(s, r.ip) or containsAgent(s, r.agent))
      then
        add r to s.list;
      else
        create new session $s'$
        add r to $s'.list$
      end;
end;
```

```
The session labeling procedure:

DB_robots: set of web robots detected by WebLog Expert.

for each $s \in S$ do
  if containsTrapFile (s) then
    s.label =1;
  else if contains (DB_robots, s.ip) or contains (DB_robots,
s.agent) then
    s.label =1;
  else
    s.label =0;
  end;
end;
```

Figure 1. the proposed procedures for session identification and session labeling

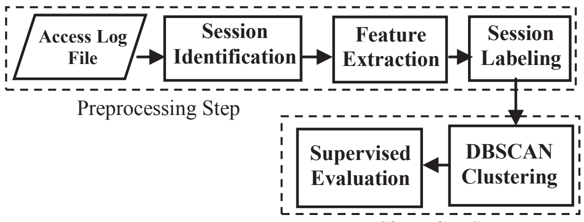


Figure 2. The flowchart of the general procedure used in this paper

A. *Session Identification*

It is essential to note that most web robots and even conventional web browsers tend to parallelize and divide their tasks among multiple threads to accelerate and facilitate achieving their goals. So, it is common that a session contains different user agent strings or IP addresses.

According to the above arguments, two consecutive HTTP requests that have the same IP addresses or same user agents, will belongs to a same session, if the time-lapse between them is within a pre-defined threshold (30 minutes in the majority of web-related literature). Fig. 1 shows the pseudo code of the procedure used for the session identification.

B. *Feature Extraction*

In this step, four attributes are extracted for each session to identify and distinguish automated and human visitors. Maximum rate of browser file request and Penalty are new features proposed in this paper.

1) Trap file request: a binary feature which demonstrates whether a session contains a request for Trap files. These files are the resources that should never be requested by human users because there is no link from the web site to these files and moreover, most users are unaware of them. Therefore, this feature can be a strong attribute to distinguish human users from web robots. Typical files used in security attacks like cmd.exe and robots.txt are common Trap files used in most related researches [5].

In this paper, we consider sitemap.xml another Trap file which is a guideline for most search engine web robots. This file contains a list of all web pages and URL addresses of a web site that cannot be discovered easily by search engine web robots.

According to these arguments, the presence of a Trap file request in a session will have a greater impact on it being classified as web robots. So, we use this feature as a basic attribute for session labeling.

2) Maximum rate of browser file request: a numerical attribute proposed based on the resources requested in a session and it is not expected to change over time.

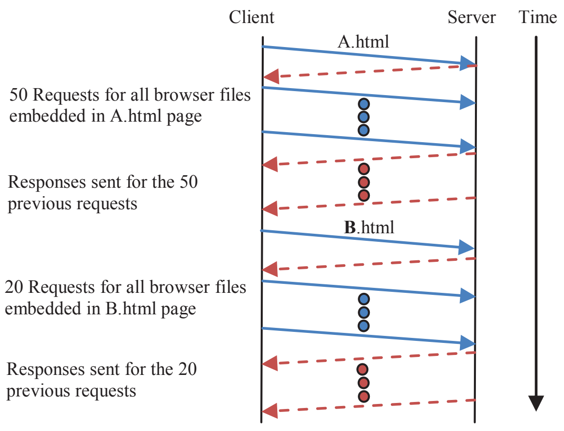
When a human user types a URL address in the address bar or clicks on a link to visit a new page, the browser analyzes the requested web page and then automatically, starts sending a barrage of requests to the server to achieve all embedded files on the page, such as videos, images, sounds and client side scripts

[5]. In this paper, we call these resources browser files and consider them an index for patterns of human users because in contrast to humans, web robots can freely decide which resource is suitable to be requested. In addition, the majority of web robots do not need the files like client side scripts; and actually, do not have the ability to interpret and implement them. It is important to note that since the image robots are so interested in images and don't download other resources such as client side scripts; we consider the images the browser files just when they are associated with other browser files such as scripts.

As mentioned above, it is reasonable to expect that the maximum rate of browser file requests would be relatively high in human sessions and low for web robots. Fig. 3 shows the total requests in a session the maximum rate of browser file request of which is equal to 50. The arrows show the requests from the client to the server and the dashed ones are the responses of these requests.

3) Penalty: a numerical attribute proposed based on the navigational patterns of humans which involve a large number of frequent back-and-forward movements and loops. Having a view restricted by the structure of links of a site to find the required information, "back" and "forward" option in web browser's history and disorienting the humans during their visits are some reasons that cause such navigational patterns. While after the first crawl of a site, robots can detect where the required information resides and restrict their next requests to specific areas of that site [5].

Penalty attribute penalizes each back-and-forward navigation or loop, and it is reasonable to expect a larger value for this attribute among human users than web robots. If the sequence shown in equation 3, demonstrates the order of the pages visited by a client in a session, we will be able to calculate the Penalty by creating a tree shown in Fig. 4. To create such a tree, we assign a new node for a page that is visited for the first time. The parent of such a node is the node which has exactly been visited in the previous step.



Client
Server
Time
A.html
50 Requests for all browser files embedded in A.html page
Responses sent for the 50 previous requests
B.html
20 Requests for all browser files embedded in B.html page
Responses sent for the 20 previous requests

Figure 3. A session with Maximum rate of browser file request = 50

$$S = a, b, a, c, d, c, d, e, d, a \qquad (3)$$

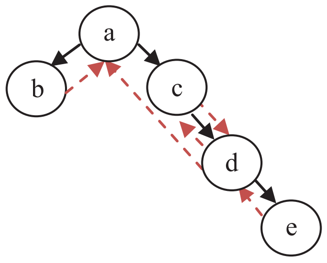


Figure 4. This tree shows the pages visited by a client

The relationship between such these child and parent nodes is shown by the arrows. While if a page is visited again, we will assign a relationship between the previous page and the node associated with the last page and show it with a dashed arrow. Obviously, the Penalty attribute is equal to the number of such arrows (in Fig. 4, the Penalty is equal to 5).

4) Percentage of 304 response codes: a numerical attribute calculated as the percentage of responses with status code 304 in a session. A response sent with this code indicates that the resource has not been modified since the version specified by the request headers. According to [3], web browsers tend to have higher percentage of 304 response code than humans.

C. Session Labeling

Since our data set contains thousands of sessions and is excessively large to be labeled manually; we implement an automatic multistage method described below for session labeling:

- The first stage is performed by observing whether a web visitor has attempted to access a Trap file. As previously mentioned, the presence of a Trap file request in a session will have a greater impact on it being classified as web robots.
- In the second stage, we use the statistical information reported by the WebLog Expert, a fast and powerful access log analyzer [17], in order to create a database that contains the IP addresses and the user agent strings of web robots known by this analyzer. So, each session is compared against this database and if the IP or user agent matches, we will label the session as a web robot.
- Finally, all remainder sessions are labeled as human.

Fig. 1 shows the pseudo code of the procedure used for the session labeling.

V. EXPERIMENTS

now, we use two large access log files, one generated from an academic web site[2] (from 26 October 2013 through 26 November 2013) and another, from an educational site[3] (over a one-month period in November 2013) to evaluate the proposed algorithm for web robot

[2] www.imamreza.ac.ir
[3] www.articlebaz.com

detection. Table 1 shows the number of sessions and class label distributions in the data sets mentioned above.

In this paper, we utilize the implementation of the DBSCAN algorithm provided in WEKA data mining software[4] and use the default set of parameters specified in these tools (*Epsilon=1,MinPts=6*). Results show that for both data sets, two final clusters and some noises are produced. Fig. 5 demonstrates the recall metric for each data set and the equation 4 shows how this metric is calculated. n_{ij} is the number of elements from the category i in cluster j; while the total number of all instances in this category is n_i [9].

$$recall(i, j) = \frac{n_{ij}}{n_i} \tag{4}$$

Now, we use the supervised evaluation metrics listed in table 2 to evaluate the performance of the proposed method and compare it with SOM (self-organizing map) used in the previous related work [7].

We use the MATLAB's Neural Network Toolbox and choose a SOM comprising 100 neurons in a 10-by-10 hexagonal arrangement. The training is done by 200 epochs.

Rand Index shows the number of correct detections and indeed the accuracy of a clustering method. *TP* and *TN* are respectively the number of robots and humans correctly identified while *FN* is the number of robots detected as humans and *FP* shows the number of humans seen as robots [2].

The entropy of a cluster reflects how the members of two categories are distributed within each cluster. In the equation of entropy, n is the total number of instances while n_c means the number of categories, humans and robots. Moreover, n_{ij} is the number of elements from the category i in cluster j which has n_j elements [9]. On the other hand, the purity metric focuses on the frequency of the most common category into each cluster and it is computed by taking the weighted average of maximal precision values [10]. Jaccard coefficient shows the proportion of robots correctly classified to the number of all instances which are incorrectly identified [9].

As an interpretation of the final clusters, some web robots detected as humans have none-zero values for the Maximum rate of browser file request and in this respect, they behave like humans.

Moreover, sometimes they have none zero values for the Penalty and undoubtedly, zero for the Trap file request.

TABLE I. CLASS DISTRIBUTION IN THE DATA SETS

Specifications	*Imam Reza University*	*ArticleBaz*
Name	D1	D2
# of Requests	311633	372304
# of Sessions	17969	22092
# of Humans	16799	18144
# of Robots	1170	3948

[4] WEKA 3.6.6

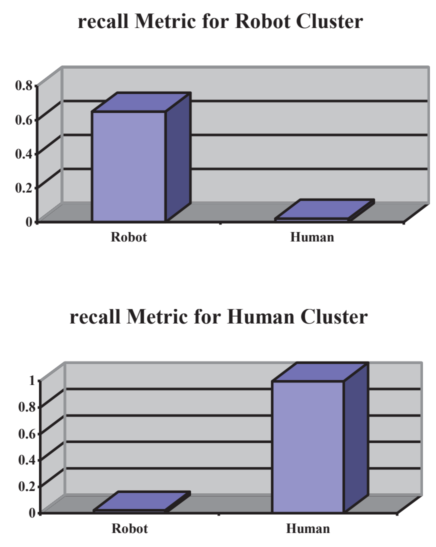


Figure 5.a. The recall metric for data set D1

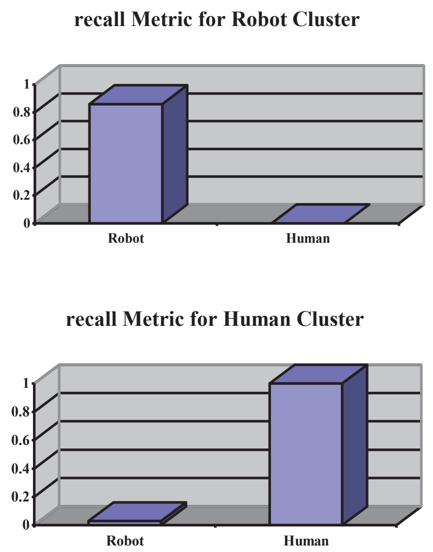


Figure 5.b. The recall metric for data set D2

Table 3 lists some examples of such web robots which imitate the human's behaviors. On the contrary, these robots are some popular non-malicious ones which work for search engines.

According to Fig. 5.a, some humans are incorrectly placed in the robot's cluster. Since these humans requested a 'robots.txt' file, they have been clustered as web robots. So, the mislabeling caused this mistake. It shows why the conventional methods used for session labeling are untrustworthy.

Finally, noises are the robots which have zero values for all 4 features and are interested in other types of resources. Thus, they are not like others.

TABLE II. EVALUATION METRICS

Evaluation metric	Metric value for DBSCAN			Metric value for SOM		
	on data set D1	*on data set D2*	*Average*	*on data set D1*	*on data set D2*	*Average*
$RI = \frac{TP+TN}{TP+TN+FP+FN}$	0.997	0.998	0.997	0.7	0.74	0.72
$Jaccard = \frac{TP}{TP+FP+FN}$	0.94	0.988	0.964	0.15	0.38	0.265
$Entropy = \sum_{j=1}^{n_c} \frac{n_j}{n} e_j,\ \ e_j = -\sum_{i=1}^{n_L} \frac{n_{ij}}{n_j} \log_2 \frac{n_{ij}}{n_j}$	0.025	0.018	0.0215	0.3	0.39	0.345
$Purity = \sum_{j=1}^{n_c} \frac{n_j}{n} p_j,\ \ p_j = \max_i \frac{n_{ij}}{n_j}$	0.982	0.973	0.977	0.91	0.81	0.86

TABLE III. SOME EXAMPLES OF WEB ROBOTS THAT TRY TO IMITATE THE HUMAN'S BEHAVIORS

Robot name	User-agent string
BingPreview	Mozilla/5.0(Windows NT 6.1; WOW64) AppleWebKit/534+ (KHTML, like Gecko) BingPreview/1.0b
Google Web Preview	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko; Google Web Preview)
Spider Symantec Huawei	HuaweiSymantecSpider/1.0+DSE-support@huaweisymantec.com

VI. CONCLUSIONS AND REMARKS

Web administrators should pay special attention and closely inspect web sessions that correspond to web robots because the traffic of these autonomous systems occupies the bandwidth, reduces the performance of web servers and in some cases, threatens the security of human users.

In this paper, we propose a density based method called DBC_WRD to distinguish the robot traffics of two real large data sets. We assume the web visitors as the instances of a multi-dimensional space and use DBSCAN as a density based clustering algorithm. Moreover, based on the navigational patterns and the resources requested by web visitors, we propose two new features to describe them and differentiate robots from humans. Also, by focusing on one of the disadvantages of the DBSCAN algorithm, we just utilize 4 features and overcome the curse of dimensionality.

The supervised evaluations show that from the standpoint of clustering quality and accuracy, DBC_WRD performs better than state-of-the-art algorithms.

Eventually, it can be concluded that some non-malicious popular web robots, through imitating the behavior of human users, make it difficult to be identified.

In future works, it would be interesting to focus on the behavior of malicious web robots and try to find the features which are able to well distinguish them from others, especially the non-malicious robots.

REFERENCES

[1] A. Stassopoulou and M. D. Dikaiakos, "web robot detection:a probabilistic reasoning approach", Computer Network, vol. 53, pp. 265-278, 2009.

[2] B. S. Everitt, S. Landau, M. Leese, and D. Stahl, Cluster Analysis. Whiley Press, 2009.

[3] C. Bomhardt, W. Gaul, and L. Schmidt-Thieme, "Web Robot detection pre-processing web logfiles for Robot Detection", New Developments in Classifiation and Data Analysis, pp. 113-124, 2005.

[4] D. Doran and S. S. Gokhale, "A classification framework for web robots", Journal of the American Society for Information Science and Technology, vol. 63, no. 12, pp. 2549-2554, 2012.

[5] D. Doran and S. S. Gokhale, "Web robot detection techniques: overview and limitations", Data Mining and Knowledge Discovery, vol. 22, pp. 183-210, 2010.

[6] D. Petrilis and C. Halatsis, "Two-level Clustering of Web Sites Using Self-Organizing Maps", Neural Processing Letters, vol. 27, no. 1, pp. 85-95, 2008.

[7] D. Stevanovic, N. Vlajic, and A. An, "Detection of malicious and non-malicious website visitors using unsuperviesd neural network learning", Applied Soft Computing, vol. 13, no. 1, pp. 698-708, 2013.

[8] D. Stevanovic, A. An, and N. Vlajic, "Feature evaluation for web crawler detection with data mining techniques", Expert System with Application, vol. 39, pp. 8707-8717, 2012.

[9] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo, "A comparison of extrinsic clustering evaluation metrics based on formal constraints", Information Retrieval, vol. 12, no. 4, pp. 461-486, 2009.

[10] J. Han and M. Kamber, Data Mining: Concepts and Techniques. Morgan Kaufmann, 2011.

[11] J. Rajabnia, M. Zabihi, and M. Vafaei Jahan, "Web Robot Detection With Fuzzy Inference System Based on decision trees", The Seventh Iran Data Mining Conference, Tehran, 2013.

[12] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", Knowledge Discovery and Data Mining (KDD '96), Portland, Oregon, 1996.

[13] P. Hayati, V. Potdar, K. Chai, and A. Talevski, "Web Spambot Detection Based on Web Navigation Behaviour", 24th IEEE International Conference on Advanced Information Networking and Applications (AINA), Perth, Western Australia, pp. 797-803, 2010.

[14] 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.

[15] S. Kwon, YG. Kim, and S. Cha. "Web robot detection based on pattern-matching technique", Journal of Information Science, vol. 38, no. 2, pp. 118-126, 2012.

[16] Y. Hiltunen, M. Lappalainen, "Automated personalization of internet users using self organizing maps", in IDEAI, Manchester, UK, pp. 31-34, 2002.

[17] WebLog Expert [Online], http://www.weblogexpert.com, 2014.