The ISC Int'l Journal of Information Security
ISeCure

January 2014, Volume 6, Number 1 (pp. 77–89)
http://www.isecure-journal.org

A Density Based Clustering Approach to Distinguish Between Web Robot and Human Requests to a Web Server

Mahdieh Zabihi [1,*], Majid Vafaei Jahan [2], and Javad Hamidzadeh [3]

[1] *Imam Reza International University, Mashhad, Iran*
[2] *Department of Computer Engineering Mashhad Branch-Islamic Azad University, Mashhad, Iran*
[3] *Faculty of Computer Engineering and Information Technology, Sadjad University of Technology, Mashhad, Iran*

ARTICLE INFO.

Article history:
Received: 4 July 2014
Revised: 16 September 2014
Accepted: 7 October 2014
Published Online: 11 October 2014

Keywords:
Behavioral Patterns of Web Visitors, DBSCAN, Density Based Clustering, Significance of the Difference Test, Web Robots.

ABSTRACT

Today world's dependence on the Internet and the emerging of Web 2.0 applications is significantly increasing the requirement of web robots crawling the sites to support services and technologies. Regardless of the advantages of robots, they may occupy the bandwidth and reduce the performance of web servers. Despite a variety of researches, there is no accurate method for classifying huge data sets of web visitors in a reasonable amount of time. Moreover, this technique should be insensitive to the ordering of instances and produce deterministic accurate results. Therefore, this paper presents a density-based clustering approach using Density-Based Spatial Clustering of Applications with Noises (DBSCAN), to classify web visitors of two real large data sets. We propose two new features based on the behavioral patterns of visitors to describe them. What's more, we consider 12 common features and use the significance of the difference test (T-test) to reduce the dimensions and overcome one of the disadvantages of DBSCAN. Based on the supervised evaluation metrics, the proposed algorithm has the 95% of Jaccard metric and produces two clusters having the entropy and purity rates of 0.024 and 0.97, respectively. Furthermore, from the standpoint of clustering quality and accuracy, the proposed method performs better than state-of-the-art algorithms. Finally, it can be concluded that some known web robots through imitating human users make it difficult to be identified.

1 Introduction

The internet regarded as one of today's most important technologies is a massive information repository and a new medium for communication and collaboration. Undoubtedly, to manage and update this repository and gain some knowledge, appropriate solutions are needed. Web robots one of these solutions send requests to web servers and analyze the received data to fulfill their specific purposes. Organizations tend to use these active researchers to collect the statistics of the dynamic content generated by users of their websites. People's favorites, opinions, requirements, and their reactions are noteworthy for web robots. Also, site maintenance and checking for broken hyperlinks are in the realm of these web robots.

[*] Corresponding author.

Email addresses: `m.zabihi@imamreza.ac.ir` (M. Zabihi), `vafaeiJahan@mshdiau.ac.ir` (M. Vafaei Jahan), `j_hamidzadeh@sadjad.ac.ir` (J. Hamidzadeh).

ISSN: 2008-2045 © 2014 ISC. All rights reserved.

According to a statistical report [1], web robots can occupy more than half of network bandwidth on several different domains and reduce the performance of web servers. So, the web robot detection not only leads us to design websites more effectively, but also it causes to configure web servers more efficiently. Yet, negligence, failure to follow instructions on how to design robots, and changing their characteristics in order to imitate human behavior are some hinders to solve this problem [2]. Today, there are many websites providing the information of known web robots. However, in the emerging era of sophisticated and ever-evolving web bots, gathering and updating this dynamic information is impractical and far-fetched [3]. Therefore, relying on the syntactical log analysis using merely some usual characteristics of web robots cannot be very accurate and applicable. In contrast, based on analytical learning techniques focusing on web robot behavior and its navigation pattern can provide a boarder perspective on real nature of web visitors [4].

Some related studies for web robot detection concentrate on behavioral details of web visitors and introduce some attributes to distinguish robots from human users. Also, they use supervised learning methods to classify them. One of the same latest researches is the first study employing the SOM and ART2 clustering algorithms as the unsupervised learning techniques [3]. The authors inform that the visitors labeling necessary for classification techniques suffers from the lack of accuracy and cause the non-generalized results in practice.

In this paper, DBSCAN (Density-Based Spatial Clustering of Applications with Noises) is used to distinguish the robot traffics on two real websites.

Due to the dynamism and diversity in behavior and operation of web robots, using features that enable us to fully distinguish them is important. Hence, the significance of the difference test (T-test) is utilized and the distribution pattern of 14 features is considered in order to reduce the dimensions and surmount one of the disadvantages of the DBSCAN algorithm. Finally, 4 features are just selected as the appropriate attributes two of which are the new features proposed in this paper. These new features are based on the behavioral patterns of web visitors and remain invariant over time.

According to data mining concepts [5], if we can demonstrate the efficiency of a clustering algorithm by using supervised evaluation metrics, we will be able to have more confidence in clustering method in question. Therefore, in one of the preprocess steps, the label of instances (web visitors) are identified and substantially, some popular supervised metrics are utilized to evaluate the proposed method.

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

The content of this paper is organized as follows: In Section 2, a survey of web robot detection is presented. Section 3 formulates the web robot detection problem while Section 4 presents the proposed algorithm to solve this problem. In Section 5, the details of the DBSCAN algorithm is covered. In Section 6, the proposed method is evaluated and compared with state-of-the-art algorithms. Eventually, Section 7 finalizes our conclusions and future works.

2 Related works

To date, several studies have looked at the use of different techniques to detect web robots. In one of the latest of such studies [6], the authors provide two schemes (TSSNB and ABS algorithms) to fight against unwanted automatic web crawlers. TSSNBS (Too Simple Sometimes Naive Blocking Schema) presents a quick and relatively inaccurate method to detect malicious web spiders based on syntactical log analysis while ABS (Adaptive Blocking Schema) is a combination of machine learning techniques and some advanced metrics introduced for malicious crawlers. Another similar study [7] uses four syntactical analysis methods and integrates the results by intersection and union operations to confirm a request as a robot or human. User-Agent check, IP-Address Check, Count of HTTP requests and HTTP requests with unassigned referrers are the syntactical log analysis utilized in this paper. The research provided in [8] discusses the robots exclusion protocols helping to detect and control the behavior of web crawlers. The authors use the syntactical log analysis to detect the web robots and utilize the .htaccess file to find the spam bots.

According to [4], the researches mentioned above are based on the information recorded in the server access log, and in fact they syntactically analyze each entry in that log while some other studies employ analytical learning techniques using behavioral pattern analysis and characteristics of web visitors. These techniques are stronger than syntactical log analysis ones [4].

In one of such studies [1], the types of resources requested by web robots and human users are considered by using recent web logs from an academic web server. The distribution of response sizes, response codes, and the popularity of resources for requests are another metrics discussed for web visitors. According to this research, it is clear that why web robots severely handicap the ability of web server caches to operate with high performance. Based on analytical techniques used in [9], the authors present a Bayesian approach to crawler detection and declare a dynamic threshold for session identification. Also, they compare their results to the obtained results with the decision tree and achieve very high recall and precision values in web robot detection. Hidden Markov model and neural network employed in [10, 11] are another classification model used for web robot detection. In our earlier work [12], 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 describing web users and facilitate the designing of fuzzy inference system.

Based on our recent searches, the studies presented in [13] and [3] are the first works that apply unsupervised learning to the problem of web visitor categorization. In [13], the authors utilize K-means clustering algorithm to detect the focused web crawlers visiting pages related to a particular subject by using general crawling mechanisms. In [3], the authors use SOM and ART2 clustering methods for web robot detection and explicitly point out that the previous works based on classification techniques are effective and accurate if and only if preceded by a reliable data-labeling process. On the other hand, an accurate web session-labeling strategy relies on an expert who is sufficiently familiar with the sophisticated and dynamic web robot's nature [3]. As a result, the authors postpone the session labeling to the supervised evaluations of the clustering algorithms to facilitate the understanding and validating of the final results and obtain a better perception of the cluster's nature and concepts.

According to the reasons mentioned, a clustering algorithm is utilized and appraised by supervised evaluation metrics.

In the comparison of the previous works, the contribution of our research is twofold:

(1) Our research is the first study utilizing 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 adequately more than just a few thousands of objects [14]. Despite previous works using the SOM, ART2, and K-means clustering, DBSCAN can produce deterministic clusters of huge databases in a reasonable amount of time. It helps the user to determine the values for its two input parameters and it is insensitive to the ordering of instances [14, 18]. This potential encouraged us to organize web users as the instances of a spatial space and utilize the capability of the DBSCAN algorithm in clustering them.

(2) In this paper, two new features are proposed 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. An attempt is made to select these features in a way that not change over time in order to keep the techniques effective across server domains and encountering of evolving robot traffics.

3 Web Robot Detection Problem

According to the requests stored in an access log file of a web server, the problem of web robot detection tries to identify the traffic of web robots [4]. In such a file, for each received request, some information is stored as an individual record. A session is a collection of all requests from a user during a single visit. The principle of this paper concentrates on web sessions rather than individual records and formalizes the session identification as follows:

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

$$S = \{S_i | s_i \subseteq R, \bigcup_{i=1}^{m} s_i = R, \bigcap_{i=1}^{m} s_i = \emptyset\} \qquad (1)$$

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, f function is declared to detect the type of each session and assign 0 to it if it belongs to a human visitor and vice versa. In this paper, the f function is the DBSCAN clustering algorithm.

$$f : S \to \{0, 1\}, f(s_i) = \begin{cases} 0 \;\; \textbf{if} \;\; s_i \; is \; Human \\ 1 \;\; \textbf{if} \;\; s_i \; is \; Robot \end{cases} \qquad (2)$$

4 The Proposed Method

In this section, the method proposed for web robot detection is presented. Figure 1 shows the flowchart of the general procedure which is used in this paper and involves two steps: preprocess and clustering. Before using the DBSCAN algorithm, the preprocessing step prepares the necessary backgrounds for the clustering.

As mentioned earlier, the clustering instances are the sessions belonging to web visitors. As a result, the web server access logs should be analyzed to extract all sessions and define some features (attributes) to describe the visitors. On the other hand, it is needed to identify the label of sessions and utilize them for

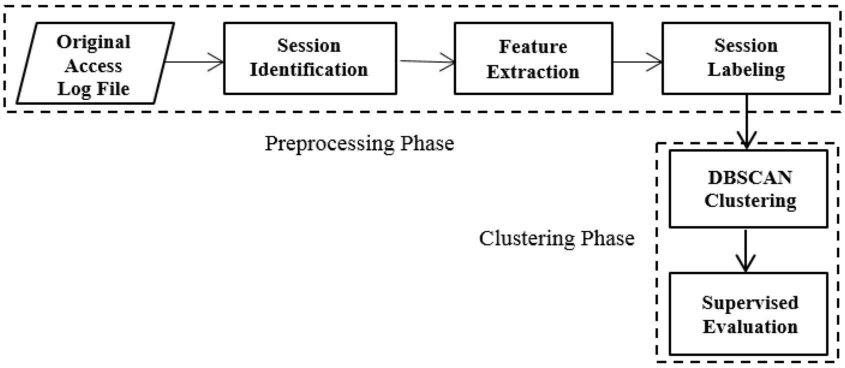


Figure 1. The flowchart of the process used in this paper.

supervised evaluations of clustering algorithm. In steps 1 and 2, an implemented java-based log analyzer is employed to preprocess the web server access logs.

4.1 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 [4].

According to what mentioned above, two consecutive HTTP requests having the same IP addresses or same user agents will belong 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).

4.2 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 the new features proposed in this paper.

(1) Trap file request: a binary feature demonstrating 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 website to these files and moreover most users are not aware of them. Therefore, a strong feature to distinguish human users from web robots is the access to such resources [15]. Typical files used in security attacks like cmd.exe and robots.txt are some of these resources [2].

In this paper, sitemap.xml is regarded as another trap file guiding most search engine web robots. This file contains a list of all web pages and URL addresses of a website that cannot be discovered easily by search engine web robots [16]. Accordingly, the presence of a trap file request in a session will have a greater impact on it to be classified as web robots. In this paper, this feature is used as a basic attribute for session labeling.

(2) The 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 [2]. In this paper, these resources are called browser files and regarded as 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. However, a small number of robots are able to understand the meaning of these scripts, and substantially request them. BingPreview web robot, the malicious robots responsible for SQL injection attacks, and some spam bots are such these robots [17].

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.

Figure 2 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. It is important to note that, image robots are so interested in images and do not download other files such as client side scripts. So, the images

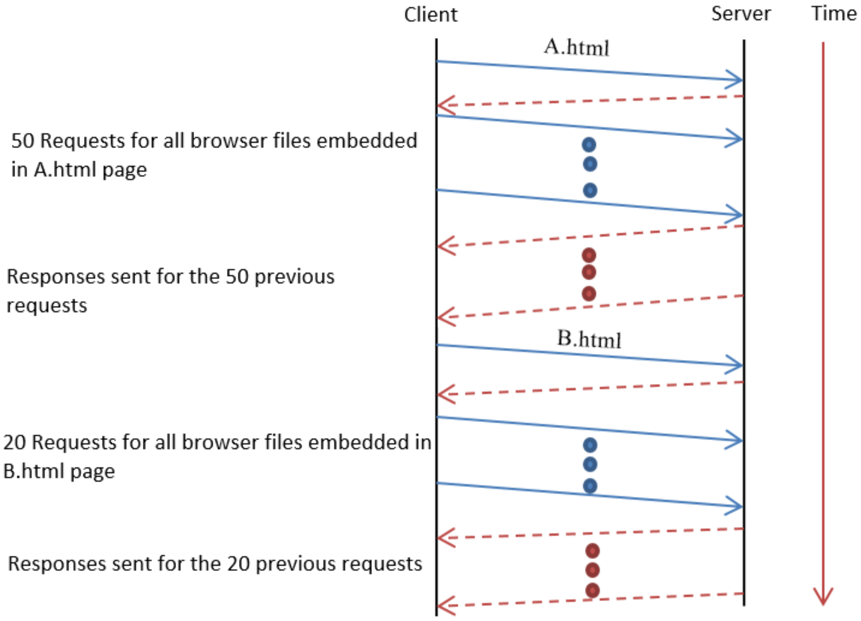

Client
Server
Time
A.html
50 Requests for all browser files embedded in A.html page
Responses sent for the 50 previous requests
B.html
20 Requests for all browser files embedded in B.html page
Responses sent for the 20 previous requests

Figure 2. An example of a session the Maximum rate of browser file request of which is equal to 50.

are regarded as the browser files just when they are associated with other browser files such as scripts. Thus, the Maximum rate of browser file request for such robots will be equal to zero.

(3) The Penalty: a numerical attribute proposed based on the navigational patterns of humans involving a large number of frequent back-and-forward movements and loops. Having a view restricted by the links structure of a website 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. In contrast to humans, 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. Moreover, robots never need to curb their crawls only to those areas specified by the links structure of a site. So, the navigational patterns of web robot are so simpler than human's [2].

The Penalty attribute penalizes each back-and-forward navigation or loop, and it is reasonable to expect a larger value of this attribute among human users.

If the sequence shown in 3 demonstrates the order of the pages visited by a client, the Penalty of this client can be calculated by a tree shown below:

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

To create such a tree, a new node is assigned for each page visited for the first time. The parent of such a node is the node having exactly been visited in the previous step. The relationship between such these child and parent nodes is shown by the edges. However, if a page is visited again, a dashed arrow is appointed from the previous page to the current node, and obviously the Penalty attribute is equal to the number of such these edges (in Figure 3, the Penalty is equal to 5).

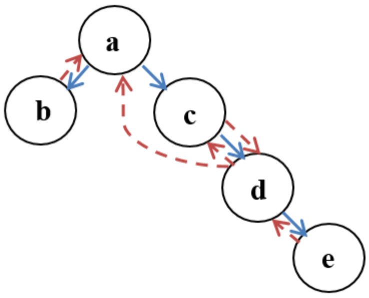


Figure 3. The tree that shows the sequence of the pages visited by a client.

(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. This means that there is no need to retransmit the resource because the client has still a previously-downloaded copy.

Since the information caching can save the server resources and also increase the processing speed of clients, it is clear why most web browsers usually cache web resources while web robots rarely use information caching.

It can be concluded that web browsers tend to have higher percentage of 304 response code than humans.

4.3 Session Labeling

As previously mentioned, the goal of this step is to identify the label of each session in order to utilize them for supervised evaluation of clustering algorithm. Since our data set contains thousands of sessions and is excessively large to be labeled manually, an automatic multistage method is implemented and described as follows:

(1) 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 to be classified as web robots.

(2) In this step, the WebLog Expert [4, 18], a fast and powerful access log analyzer, is utilized so as to create a database that contains the IP addresses and the user agent strings of web robots known by this analyzer. Therefore, each session is compared against this database and if the IP or user agent string matches, the session will be labeled as a web robot.

(3) All the remainder sessions having not been labeled yet are identified as human users.

In the following, the DBSCAN clustering algorithm is described in details.

5 DBSCAN Clustering Algorithm

DBSCAN [19] regards clusters as dense regions of instances in a data space. The noises are the regions with low density separating the clusters from each other. Each cluster contains at least *MinPts* objects placed in a neighborhood with a radius ϵ of a given object named 'core‘. Indeed, each core forms an initial cluster, *MinPts*, and more importantly, ϵ are input parameters defined by the user. The most common distance metric used to calculate the neighborhoods is Euclidean distance. For given two objects, p and q, and with respect to *MinPts* and ϵ, it can be said that:

- The object p is **directly density-reachable** from the *core* object q if p is within ϵ-neighborhood of q.
- The object p is **density-reachable** from the core object q if a chain of direct density-reachable objects $p_1 \cdots p_n$ can be found.
- The object p is **density-connected** to object q if there is an object o such that both p and q are density-reachable from o.

Algorithm 1 DBSCAN

```
Input: a set of objects $D$, the parameters $MinPts$ and $\epsilon$
for each $o \in D$ do
    if $o$ is not yet classified then
        if $o$ is a core object then
            Collect all objects density-reachable from
            $o$ and assign them to a new cluster.
        else
            assign $o$ to NOISE.
        end if
    end if
end for
```

Algorithm 1 shows the pseudo-code of DBSCAN. This algorithm searches for clusters by checking the -neighborhood of each point in the database. If the ϵ-neighborhood of a point p contains more than *MinPts*, a new cluster with p as a *core* object is created. Then iteratively, DBSCAN collects directly density-reachable objects from these *core* objects which may involve the merge of a few density-reachable clusters. The process terminates when no new point can be added to any cluster. As it is clear, every object not contained in any cluster is considered to be a noise.

6 Experimental Results and Comparisons

In the experimental section, two access log files, one generated from an academic website [1] (from 26 October 2013 through 26 November 2013) and another from a commercial website [2] (over a one-month period in September 2013) are used to evaluate the proposed method. To do so, two different experiments are conducted in this section. Table 1 shows the number of sessions and class label distributions in the data sets mentioned above. It is important to note that the sessions with less than 6 and 8 requests containing no access to trap files are eliminated from both data sets. Such sessions are the meaningless ones which do not contain any useful information to detect the visitors.

6.1 Experiment 1

In this section, one of the disadvantages of DBSCAN is discussed and a solution is offered.

As a point to this detriment, DBSCAN may not work well for high-dimensional spaces where the distance metric (Euclidean distance) can be rendered almost useless due to the curse of dimensionality making it difficult to find an appropriate value for the parameter.

[1] www.imamreza.ac.ir

[2] www.torghe.com

Table 1. Class distribution in the data sets.

Website	Data set Name	Number of requests	Number of sessions	Number of robots	Number of humans
Imamreza International University	D1	311633	17969	1170	16799
Torghe	D2	500781	21553	6093	15460

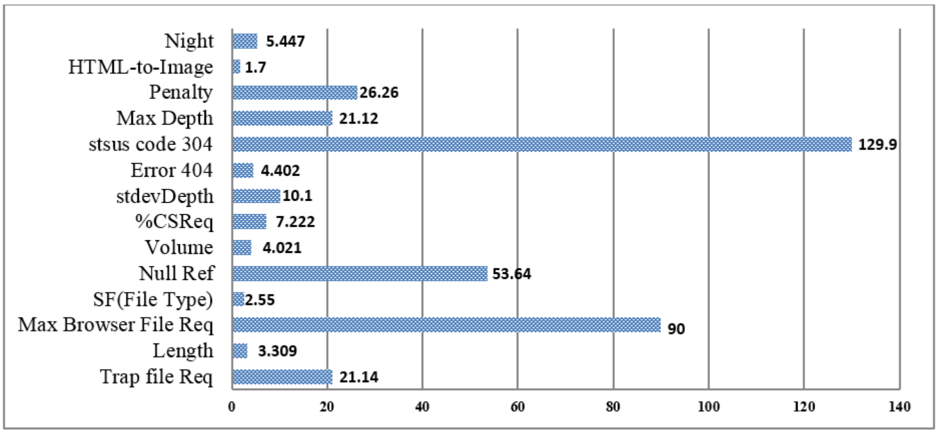


Figure 4. T-test for all features.

To solve this problem, a solution is offered trying to select the relevant and proper attributes and reduce the diminutions. Table 2 shows a list of 10 features used in other related papers to distinguish web robots from human users. The goal is to explain why the 4 features proposed in this paper are preferred to these 10 attributes. So, the distribution of the values of all features, between humans and robots, are considered and a T-test (significance of the difference test) is used to show that which attribute is significantly different between these two groups. Equation 4 shows how to calculate the T-test:

$$t(f_i) = \frac{|mean_1(f_i) - mean_2(f_i)|}{\sqrt{var_1(f_i)/n_1 + var_2(f_i)/n_2}} \qquad (4)$$

In this equation, $mean_1$ and $mean_2$ are the mean values of i^{th} feature (f_i) in robot and human groups, respectively. Also, var_1 and var_2 are the corresponding variances. The total numbers of robot and human classes are respectively shown by n_1 and n_2. Also in this test, the degree of freedom is defined as n_1+n_2-1. It is important to note that for the degree greater than 100, the i^{th} feature will be considered significantly different between humans and robots if the $t(f_i)$ is greater than 1.96 [20].

Undoubtedly, for a given attribute, the greater result shows that the relevant feature is more different between two groups [20]. Figure 4 shows the T-test results of all 14 attributes: According to Figure 4, except for the html-to-image ratio, other features can pass the test with a value greater than 1.96. Although, of all 13 remaining features, 6 attributes have significantly different values than the others. Moreover, the Maximum rate of browser file request and the percentage of 304 response codes are the features that outshine others and their values are zero for most web robots. So, up to this point, they are selected as the final features. Additionally, the trap file request attribute used in session labeling is another attribute selected in the first consideration

Now, the distributions of the values of 3 remaining features are considered to select other final attributes. Figure 5 shows how the values of these attributes have been distributed between robots and humans. In these figures, the vertical axis show the robot or human class while the horizontal one is the values of each attribute extended in [0,1] interval.

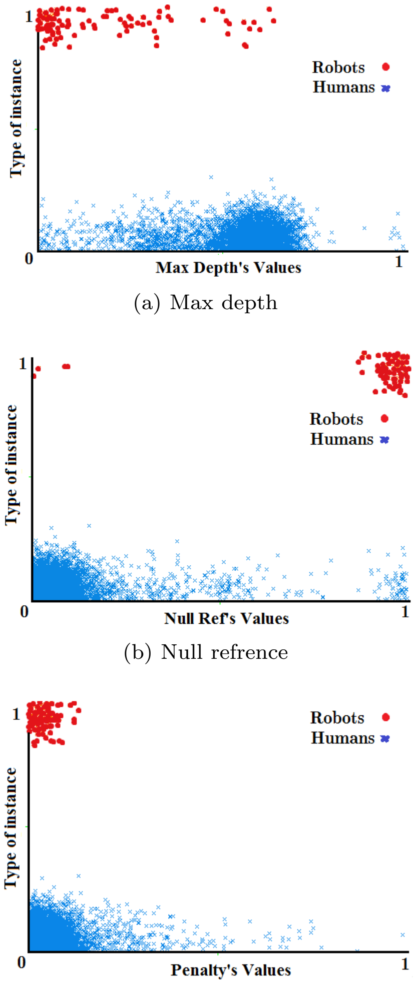
According to Figure 5, the upper part of each diagram shows how the values of the relevant feature have been distributed between robots and vice versa. Certainly, the less similarity between these two parts, the more appropriate to distinguish robots from humans.

As it is clear, the similarity between two parts of Max Depth's diagram is more than the others. Moreover, according to Null Ref's chart, for most web robots this feature is equal to 1, so it imprecisely makes some humans to be similar to the most of web robots. As a result, the Penalty attribute is selected as another final feature.

In the following, the similarities between instances

Table 2. Common Features used in other relevant papers.

Remark	Feature name	Remark	Feature name
The volume of data transmitted in a session	Volume	Duration time of a session	Length
Ratio of switching the type of files requested	Switching factor of file type (SF(File Type))	% of consecutive sequential HTTP requests	% CSRequests
% of requests with status=404	Error 404	Number of HTML page requests over the number of image files requests	HTML-to-Image
Maximum depth of all requests	Max Depth	% of requests with unassigned referrer	Null Ref
Standard deviation of requested page's depth	stdevDepth	% of requested made between 12 am to 7 am	Night



(a) Max depth

(b) Null refrence

(c) Penalty's value

Figure 5. Distribution of the values of 3 remaining attributes, between robots and humans.

within a same class are considered when the 4 proposed attributes and 10 mentioned ones are used.

According to Figure 6, based on these 4 attributes, the similarities between web robots are just equal to 1 or 0 while the similarities of humans extend in [0,1]. So, since humans and majority of web robots are not similar to each other, the proper can be easily found (default ϵ=1). In contrast, for those 10 attributes, the similarities between robots or humans are spread in [0,1] and it is hard to find ϵ because for each candidate value of this parameter, many web robots can incorrectly resemble to some humans. Although the similarities of humans based on these 10 attributes are stronger than one's based on 4 features. So, the effect of the curse of dimensionality can be neutralized by utilizing the T-test and considering the distribution of values of attributes. In second experiment, the DBSCAN algorithm is exerted on two data sets introduced above and compared with state-of-the-art algorithms.

6.2 Experiment 2

In this experiment, the implementation of DBSCAN provided in WEKA data mining software[3] and the default set of parameters specified in these tools ($\epsilon =$ 1 , $MinPts$ =6, and Euclidian distance as similarity formula) are used. The results of the study show that for both data sets, two final clusters and some noises are produced. Figure 7 demonstrates the recall metric for each data set and the Equation 5 shows how to calculate this metric. n_{ij} shows the number of elements from the category i in cluster j while the total number of all instances in this category is n_i [21].

$$recall(i,j) = \frac{n_{ij}}{n_i} \tag{5}$$

In both data sets, the noises are the robots which are not similar to others.

Now, some supervised evaluation metrics listed in Table 3 are used to evaluate the performance of the proposed method.

[3] WEKA 3.6.6

Table 3. Evaluation metrics for the DBSCAN clustering algorithm.

Evaluation metric	Metric value for data set D1	Metric value for data set D2	Average value of the metric
$RI = \frac{TP+TN}{TP+TN+FP+FN}$	0.997	0.98	0.988
$Jaccard = \frac{TP}{TP+FP+FN}$	0.94	0.951	0.945
$Entropy = \sum_{j=1}^{n_c} \frac{n_j}{n} e_j, e_j = -\sum_{i=1}^{n_L} \frac{n_{ij}}{n_j} \log_2 \frac{n_{ij}}{n_j}$	0.025	0.023	0.024
$Purity = \sum_{j=1}^{n_c} \frac{n_j}{n} p_j, p_j = max_i \frac{n_{ij}}{n_j}$	0.982	0.95	0.966

Table 4. Some samples of web robots trying to imitate the human's behaviors.

Robot name	Trap file Req	Maximum rate of browser file Req	Status code 304	Penalty	User-agent string
BingPreview	0	57	0	0	Mozilla/5.0(Windows NT 6.1; WOW64) AppleWebKit/534+ (KHTML, like Gecko) BingPreview/1.0b
Google Web Preview	0	66	0	0.530	Mozilla/5.0(X11; Linux x86_64) AppleWebKit/537.36(KHTML, like Gecko; Google Web Preview) Chrome/27.0.1453Safari/537.36
Huawei Symantec Spider	0	10	0	0.33	HuaweiSymantecSpider/1.0+DSE-support@huaweisymantec.com+(compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; .NET CLR 3.0.4506.2152; .NET CLR ;)

Rand Index shows the number of correct detections and the accuracy of a clustering method. *TP* and *TN* are the number of robots and humans correctly identified, respectively; while *FN* shows the number of robots detected as humans and *FP* is the number of humans seen as robots [23].

The entropy of a cluster reflects how the members of the two categories are distributed within each cluster. Obviously, a good clustering algorithm has a small value for this metric. In the equation of entropy, n is the total number of instances while nc 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 [23]

One of the most popular measures for cluster evaluation is purity. This 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 [5].

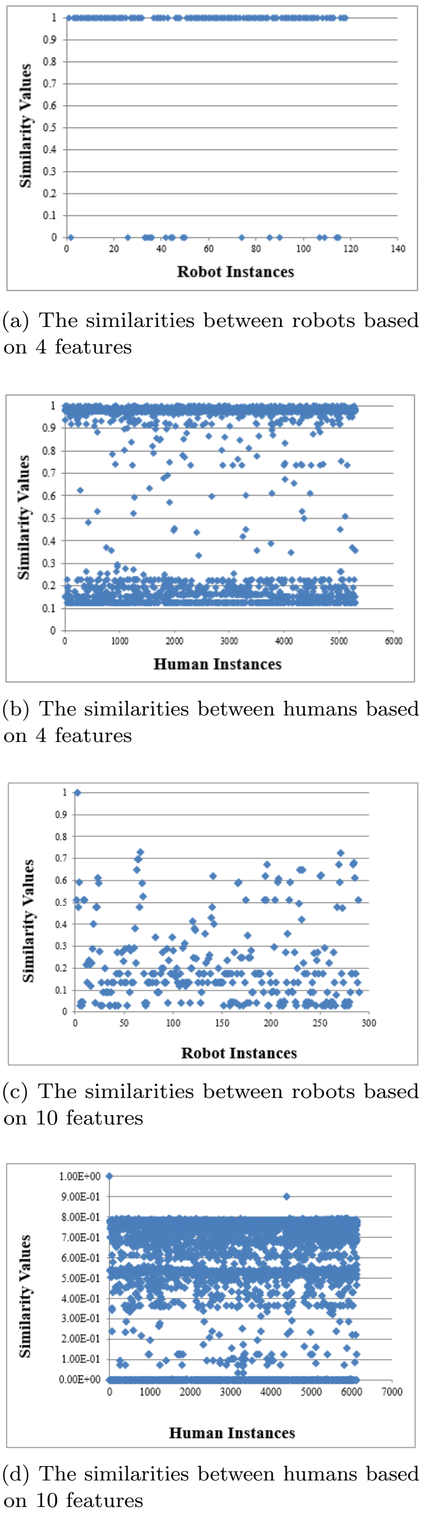
Jaccard coefficient is an important metrics used for data mining problems having asymmetric binary classes. Web robot detection is one of these issues in which the robot's class is more noteworthy than the human's. In fact, the Jaccard metric is derived from the Rand index equation *TN* of which is eliminated to emphasize the importance of *TP* [21].

As an interpretation of the final clusters, some web robots placed in human clusters 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 attribute and undoubtedly, zero for the Trap file request. It is clear that these robots are trying to imitate the human's behavior and hide themselves. Table 4 lists some samples of such web robots. It is important to note that these robots are some known ones working for search engines.

According to the definition of the Maximum rate of browser file request, the robots interested in resources like video or flash files have none-zero values for this feature, and substantially are regarded as humans. Fortunately, both data sets used in this paper have little number of such files.

According to the recall metric for robot cluster in data set D1 (shown in Figure 7a), some humans are incorrectly placed in robot cluster. Since these humans requested '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 can be untrustworthy. Robot clusters contain robots having nonzero value for Trap file request and zero for other 3 attributes. So, the similarity between these robots and humans placed in the same cluster is equal to 1.

Finally, noises are the robots interested in other types of files (like compressed or text files) and have



(a) The similarities between robots based on 4 features

(b) The similarities between humans based on 4 features

(c) The similarities between robots based on 10 features

(d) The similarities between humans based on 10 features

Figure 6. Distribution of the values of 3 remaining attributes, between robots and humans.

zero values for all 4 features. So, they are not like other instances.

Table 5 summarizes the specifications of all instances placed in final clusters.

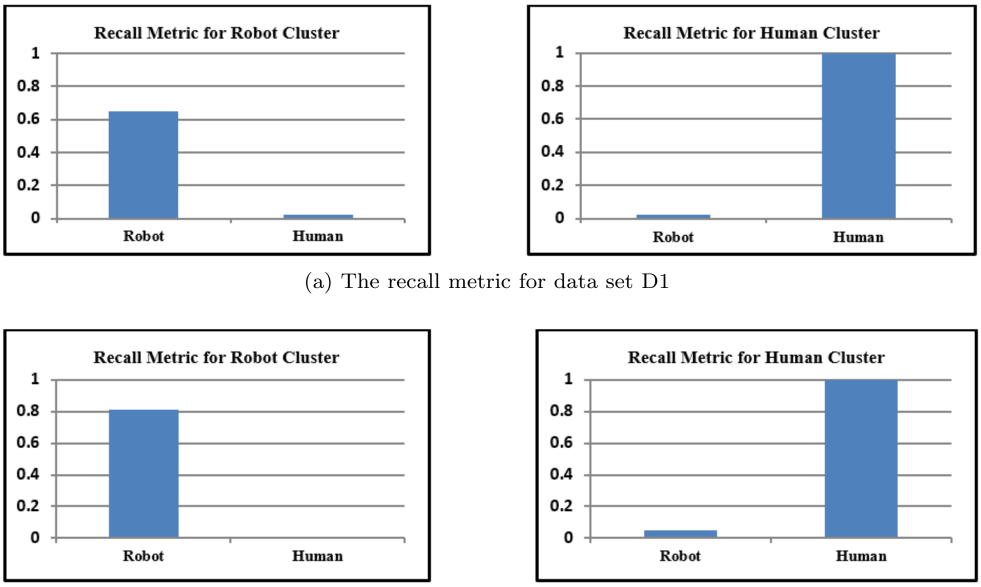
Table 5. Summary of the specifications of all instances placed in final clusters.

Instances	Trap file Request	Maximum rate of browser file Request	Status Code 304	Penalty
Robots incorrectly detected as Humans	0	Non-zero	0	Zero or non-Zero
Humans correctly detected	0	Non-zero	Zero or non-Zero	Zero or non-Zero
Robots correctly detected	1	0	0	0
Humans incorrectly detected as Robots	1	Non-zero	Zero or non-Zero	Zero or non-Zero
Noises	0	0	0	0

After the interpretation of final clusters, the proposed method is compared with state-of-the-art algorithm containing SOM (S elf Organizing Map), ART2 (Modified Adaptive Resonance Theory 2), and K-means clustering. SOM is a type of artificial neural network trained using competitive unsupervised learning. It is used to map the multidimensional input data to a lower dimensional subspace where geometric relationships between points indicate their similarity (usually Euclidean distance) [24]. To employ SOM algorithm, the MATLAB's Neural Network Toolbox with a network comprising 100 neurons in a 10-by-10 hexagonal arrangement is used (*epochs*=200). Similar to SOM, ART2 is an unsupervised neural network algorithm, and at the same time, it is based on competitive learning mechanism. Its network is trained by the input data and winning neurons are found by means of minimum Euclidean distance. Unlike SOM, ART2 is based on the winner's weight vector deemed sufficiently close to the training sample and relies on the so-called winner-takes-all rule causing different results in practice [25]. To employ this algorithm, an implemented MATLAB code base on the pseudo-code introduced in [25] is used and the ρ_{max}, $\Delta\rho$, and n_{max} are set to 1.5, 0.1 and 5, respectively.

K-means is one of the simplest unsupervised data mining methods used to classify numerical data sets into a certain number of clusters fixed prior [26]. To exert this method, the implementation of K-means provided in WEKA data mining software is used and the K parameter is set to 2.

As in the case of DBSCAN, in all these clustering methods, the Euclidian distance is used to calculate the similarity between input instances.



(a) The recall metric for data set D1

(b) The recall metric for data set D1

Figure 7. The recall metrics.

Table 6. Evaluation metrics for state-of-the-art algorithms.

Evaluation metric	Metric value for data set D1	Metric value for data set D2	Average value of the metric
$RI = \frac{TP+TN}{TP+TN+FP+FN}$	0.7	0.86	0.61
$Jaccard = \frac{TP}{TP+FP+FN}$	0.36	0.82	0.25
$Entropy = \sum_{j=1}^{n_c} \frac{n_j}{n} e_j, e_j = -\sum_{i=1}^{n_L} \frac{n_{ij}}{n_j} \log_2 \frac{n_{ij}}{n_j}$	0.38	0.034	0.42
$Purity = \sum_{j=1}^{n_c} \frac{n_j}{n} p_j, p_j = max_i \frac{n_{ij}}{n_j}$	0.85	0.91	0.63

Table 6 shows the average values of supervised evaluation metrics for these 3 clustering algorithms.

In addition to the results shown in above table, According to [27], if ART2 network is trained by the same data set but with different input orders, it may exhibit great different classification results. Moreover, it can cause a drift of cluster center in practice [28]. As one of the disadvantages of SOM, the randomized initial weights of neurons can cause different results for per clustering on the same input data. The lack of any specific standard to specify the number of neurons (input parameter) is another drawback of SOM. Moreover, the interpretation of the final clusters can be difficult and time consuming. To gain more accurate results, the number of iterations of learning step must be increased, and subsequently it requires much time to be done [29, 30]. Providing non deterministic results caused by randomly initial cluster centers and weakness against noises and local optima of the squared error function are the known basic disadvantages of the K-means algorithm [23].

As for the benefits of the DBSCAN algorithm, it requires only two input parameters and helps the user to determine the values for them. Moreover, it can handle the noises well and is mostly insensitive to the ordering of the instances. As previously mentioned, it can discover the clusters having arbitrary shapes and be very effective at clustering the data sets of significantly more than just a few thousands objects [14, 19].

7 Conclusion and Future Works

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, causes misunderstanding about the number of real visitors of websites.

In this paper, the web visitors are regarded as the instances of a multi-dimensional space and DBSCAN is used as a density-based clustering algorithm for two real large data sets. By focusing on one of the disadvantages of DBSCAN, 14 features describing web visitors are considered and the T-test is used to overcome the curse of dimensionality problem. In addition, it is demonstrated that considering the distribution of values of features between robots and humans can be useful in order to choose the appropriate attributes. Finally, 4 attributes are just selected as the final features two of which are the new ones proposed in this paper. These attributes are based on the behavioral patterns of web visitors and remain invariant over time.

The supervised evaluation metrics used in the experimental section show that the proposed method has the 95% of Jaccard metric and produces two clusters having the entropy and purity rates of 0.024 and 0.97, respectively. Furthermore, from the standpoint of clustering quality and accuracy, the proposed method performs better than state-of-the-art algorithms. Finally, it can be concluded that some known web robots, through acting like 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 enabling us to make a distinction between them and others, especially the non-malicious web robots. So, we intend to consider the behavioral patterns of web visitors in order to cluster them into four groups: human users, well-behaved web robots, malicious robots, and unknown visitors.

References

[1] D. Doran, K. Morillo, and S. S. Gokhale. (2013). A comparison of web robot and human requests. In Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM '13). ACM, New York, NY, USA, 1374-1380.

[2] D. Doran, and S. S. Gokhale. (2010). Web robot detection techniques: overview and limitations. Data Mining and Knowledge Discovery, 22, 183-210.

[3] D. Stevanovic, N. Vlajic, and A. An. (2013). Detection of malicious and non-malicious website visitors using unsuperviesd neural network learning. Applied Soft Computing, 13(1), 698-708.

[4] D. Doran. (2014). Detection, Classification, and Workload Analysis of Web Robots. Ph.D. thesis, university of connecticut.

[5] O. Maimon, and L. Rokach. (2005). Data Mining and Knowledge Discovery Handbook. Springer Press.

[6] D. Zhang, D. Zhang, and X. Liu. (2013). A Novel Malicious Web Crawler Detector: Performance and Evaluation. International Journal of Computer Science Issues (IJCSI), 10(1).

[7] T. H. Sardar, and Z. Ansari, (2014). Detection and confirmation of web robot requests for cleaning the voluminous web log data. IMpact of E-Technology on US (IMPETUS), 2014 International Conference on the, 13-19.

[8] S. Gupta, S. Tarun, and P. Sharma. (2014). Controlling Access of Bots and Spamming Bots. IJCER, 3(2), 87-92.

[9] A. Stassopoulou, and M. D. Dikaiakos. (2009). Web Robot Detection: A probabilistic reasoning approach. Computer Networks, 53, 265-278.

[10] W. Lu, and S.Yu. (2006). Web robot detection based on hidden Markov model. Proceedings of international Conference on communications, circuitsand systems, pp18061810.

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

[12] M. Zabihi, and J. Hamidzadeh, M. Vafaei Jahan. (2014). Fuzzy Inference for intusion detection of web robots in computer networks. The 45th Annual Iranian Mathematic Conference, Semnan.

[13] N. Jain, and M. P. Mangal. (2014). An Approach to build a web crawler using Clustering based K-Means Algorithm. Journal of Global Research in Computer Science, 4(12), 14-22.

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

[15] P. Jha, S. Goyal, T. Kumari, and N. Gupta. (2014). Robots Exclusion Protocol. Internation journal of emerging science and engineering, 2(5).

[16] R. Cyganiak, H. Stenzhorn, R. Delbru, S. Decker, and G. Tummarello. (2008). Semantic sitemaps: Efficient and flexible access to datasets on the semantic web. Springer Berlin Heidelberg, 690-704.

[17] N. Yousefi. (2013). Detection of Malicious Web Robots Using Machine Learning Techniques. M.Sc. Thesis, Imam Reza International University.

[18] (2014) WebLog Expert. [Online]. http://www.weblogexpert.com

[19] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu. (1998). Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and its Applications. Data Mining and Knowledge Discovery, an International Journal, 2 (2), Kluwer Academic Publishers, 169-194.

[20] G. K. Kanji. (2006). 100 Statistical Tests, 3rd ed. SAGE Publication.

[21] E. Amig, J. Gonzalo, J. Artiles, and F. Verdejo. (2009). A comparison of extrinsic clustering evaluation metrics based on formal constraints. Information Retrieval, 12(4), 461-486.

[22] X. Lin, L. Quan, and H. Wu. (2008). An Automatic Scheme to Categorize User Sessions in Modern HTTP Traffic. Global Telecommunications Conference, 1485-1490.

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

[24] Kohonen, T. (1995). Self-organizing Maps. 2nd ed., Springer-Verlag, Berlin.

[25] N. Vlajic, and H.C. Card. (2001). Vector quantization of images using modied adaptive resonance algorithm for hierarchical clustering. IEEE Transactions on Neural Networks 12, 11471162.

[26] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrdl. (2001). Constrained k-means clustering with background knowledge. ICML, 1, 577-584.

[27] J. Luo, and D. Chen. (2008). An enhanced ART2 neural network for clustering analysis. Knowledge Discovery and Data Mining, 81-85.

[28] H. Zhang, W. Guan, and G. Guan. (2008). On-Line Diagnosis of Faulty Insulators Based on Improved ART2 Neural Network. Advances in Neural Networks, 465-472.

[29] M. Lotfi Shahreza, D. Moazzami, B. Moshiri, and M. R. Delavar. (2011). Anomaly detection using a self-organizing map and particle swarm optimization. Scientia Iranica, 18(6), 1460-1468.

[30] T. Vijaya Kumar, and H. S. Guruprasad. (2012). Clustering Web Usage Data using Concept Hierarchy and Self Organizing Map. International Journal of Computer Applications, 56.

[31] Bots vs Browsers. (2014). http://www.botsvsbrowsers.com

[32] user-agent-string.info. (2014). http://user-agent-string.info

Mahdieh Zabihi is currently a M.S. student of computer engineering at Imam Reza International University, Mashhad, Iran. She received her B.S. degree in computer engineering from Ferdowsi University of Mashhad. She is a member of Iranian Operations Research Society and her research interests are Comprised of Machine Learning, Soft Computing, Artificial Intelligence, Data Mining, Pattern Recognition, and Evolutionary Computation.

Majid Vafaei Jahan received B.S. degree from Ferdowsi University of Mashhad, Mashhad, Iran, in 1999, and the M.S. degree from Sharif University of Technology, Tehran, Iran, in 2001. He received the Ph.D. degree from the Department of Computer Engineering, Islamic Azad University, Science and Research Branch, Tehran, Iran, in 2009. He is with the Faculty of Islamic Azad University, Mashhad Branch, Mashhad, Iran. His research interests include Systems Modeling and Simulation, Soft Computing, Evolutionary Computation, and Software Design and Implementation. He received the Outstanding Graduate Student Award from Ferdowsi University of Mashhad in 1999 and Top Researcher Award in Engineering Field in 2012 by Islamic Azad University.

Javad Hamidzadeh is currently an Assistant Professor at Faculty of Computer Engineering and Information Technology, Sadjad University of Technology, Mashhad, Iran. He received the B.S. and M.S. degrees in computer engineering from Sharif University of technology in Iran, and Ph.D. degree in computer engineering from Ferdowsi University of Mashhad in Iran. His research interests include Machine Learning, Statistical Learning Theory, and Pattern Recognition. He is a member of Computer Society of Iran (CSI).