Web Robot Detection Based On Fuzzy System and PSO Algorithm

[1] **Mohammad Ordouei; [2] Iman Namdar**

[1] Computer Engineering Dep., Islamic Azad University (IAU)
Tehran,Iran

[2] Computer Engineering Dep., Islamic Azad University (IAU)
Tehran, Iran

Abstract - Web robots are applications which recursively and automatically overview the content of website documents. Some robots are considered to be malicious. Hence, identifying web robots is a classification challenge. In this research we mainly present a hybrid particle swarm optimization method and fuzzy system aiming at increasing efficiency over Web robot detection and simulation by MATLAB software. Evaluation criteria are considered: Specificity, Accuracy, F- Measure, Recall and Precision. Results of the study for the proposed method are respectively: 99.81, 96.92, 96.10, 91.39 and 99.58. The yields for proposed basic fuzzy system and fuzzy network algorithm and ANFIS neural fuzzy network algorithm indicate the priority of the proposed method other than algorithms being compared.

Keywords - *Web robot detection; fuzzy system; particle swarm optimization algorithm.*

1. Introduction

Web robots are applications which automatically and recursively review data and website contents of which being malicious behaving differently including malicious and non-malicious. Besides, human users are considered web visitors. Web robot detection is a binary classification issue aiming at classifying web users into human and non-human web users. Of the most important classifying challenges is to identify and categorize those web robots attempting to imitate human behaviors. The focus is on investigating web robots using a hybrid fuzzy system and a particle swarm optimization algorithm (PSO). Our proposed method has four stages: Preprocessing, feature selection, session labeling and classifying by hybrid fuzzy system and PSO algorithm. PSO algorithm is applied to train fuzzy system and parameter configuration. Particle swarm optimization algorithm is a sample of evolutionary procedure. During recent years evolutionary methods have been applied as a research and optimization tool within various fields including business and engineering sciences. Among the reasons for the success of the mentioned methods are their appliance domain, ease of use, their absolute, optimal, approximate answers. The paper uses MATLAB software to simulate the results and main fuzzy system and ANFIS neural fuzzy net to compare with proposed method the criteria of which to evaluate the results are as follows: precision, recall, F-measure, accuracy and specificity.

2. Web robots

Robots or web crawls extract knowledge out of web pages, initially begin with few pages then, recursively scan whole available documents. Web crawls are considered web visitors behave both malicious and non-malicious along with humanistic behaviors.

Web robots are often mentioned with different names:
1. Spider
2. Harvester
3. Crawl

All of which share a unique meaning and the phrase "robot" is considered the main name. Robots don't move through sites on their own, instead they are systematically programmed. Search engine crawlers remove web contents based on a specified order. Shopping bots generally backtrack commercial sites to verify the prices and purchased goods.

Focused crawlers search the entire web to find and investigate particular semantic pages. Verifiers aim at finding damaged or broken links through checking the web pages out. Harvesters are other web robots which request HTML, photos or documents from web pages. All of which share the same meaning but the term "robot" is considered the main name. Robots don't generally move through the sites on their instead, being systematically programmed. Search engine crawlers remove the contents according to a defined order. Shopping bots generally backtrack with commercial sites to verify the deals and prices. Focused crawlers are those which search the entire web to find and investigate specific semantic domains. Verifiers check the web pages aiming at finding broken or damaged links. Harvesters are another type of web robots requesting sources including HTML, images or documents. These are only definite web robots acting differently within web platforms. Some web robots have a regular behavior and operate based on "robot.txt" protocol facing web servers. They also hedge sending numerous requests to the server. While some web robots send a large number of requests to the server purposefully. These are generally DDos robots. DDos disable web server services sending massive distributive requests to it, a particular type of which called "7 –layer" or "application layer" and use web robots to achieve their goals. It can be a solid reason for the need to detect web robots.

3. Fuzzy systems

Membership in a set is considered zero and one for classical areas. Such that in case of any member in complex then it will be represented by 1, otherwise, by zero. In fact, membership considered a function the range of which is {0, 1}. On the other hand in fuzzy logic, the concept of membership degree extends to [0, 1]. The concept, fuzzy logic is worth notice since, through the real world, many arguments and reasons are uncertain and approximate. Robots or web crawls extract knowledge out of web pages, initially begin with few pages then, recursively scan whole available documents. Web crawls are considered web visitors behave both malicious and non-malicious along with humanistic behaviors.

Definition of a fuzzy set [2]: A fuzzy set on a universe of discourse X is a set of

$$A = \{\mu_A(x)/x : x \in X, \mu_A(x) \in [0,1] \in R\}$$

Such that $\mu_A(x)$ is the X membership degree of set A. The membership degree can take either 0 or one.

$\mu_A(x) = 0$: Implies that X doesn't necessarily belong to fuzzy set A.

$\mu_A(x) = 1$: Implies that X definitely belongs to set A.

An expert fuzzy system generally has the following sections:

1. Fuzzifier: The section receives data as the system input, converting into fuzzy singleton and is used as fuzzy maker.
2. Rule base: The section involves expert rules. In addition, one can assign each rule a weight between 0, 1 which reflects the degree of our faith towards it.
3. Inference engine: The inference engine is based on the fuzzy output which is a fuzzy set, performing its inferences by the rule base and generates the output as a fuzzy set.
4. Defuzzifier: This section converts the fuzzy sets which are the output of the inference engine, into real data as if the true data is the system's final output itself.

4. Particle swarm optimization model

Particle Swarm Optimization algorithm operates such that a patch of particles (as optimization variables) spread over the search environment. It is clear that some particles have a better position than the others. As a result based on attack particles on the other side, other particles try to match their positions with superior ones. At the same time status of other particles also changes. It is worth saying that particle position changing occurs based on its previous and neighboring particles.

Here we define the parameters to simulate the behavior:

Pbest: It is the best position ever each particle acquires during algorithm.

Gbest: It represents the best position which particles acquired through the algorithm.

Individual recognition parameter: It causes particle movement towards the best point found by itself and neighbors, which is applied as the excitation coefficient.

Social recognition parameter: Causes particle movement towards the best point ever.

Inertia coefficient w: Balances the local and general algorithm searches.

Consider jth particle having g dimension expressed as:

$$X_j = [x_{j,1}, x_{j,2}, \dots, x_{j,g}] \tag{1}$$

Each particle having a Pbest and the entire particle Gbest is:

$$Pbest_{j,1}, Pbest_{j,2}, \dots, Pbest_{j,g} \tag{2}$$

Then the changing particle position will be based on equations 3, 4:

$$v_{j,g}^{(t+1)} = wv_{j,g} + c_1 \cdot Rand()\left(Pbest_{j,g} - x_{j,g}^{(t)}\right) \tag{3}$$

$$c_2 \cdot rand() \cdot \left(gbest_{j,g} - x_{j,g}^{(t)}\right) \quad , \quad v_{\min} \le v_{j,g}^{(t)} \le v_{\max}$$

$$x_{j,g}^{(t+1)} = x_{j,g}^{(t)} + v_{j,g}^{(t+1)} \qquad \begin{matrix} j = 1,2,K\,,n \\ g = 1,2,K\,,m \end{matrix} \tag{4}$$

X is the particle position, n number of particles, m number of particle constituents, Rand () and rand () generate a random value between 0 and 1.

5. Proposed method

Detecting a web robot has following stages:
1. Preprocess log file
2. Detect and extract features
3. Session labeling (based on robots and humans)
4. Determine classifying method for web users
5. Evaluation of the proposed model

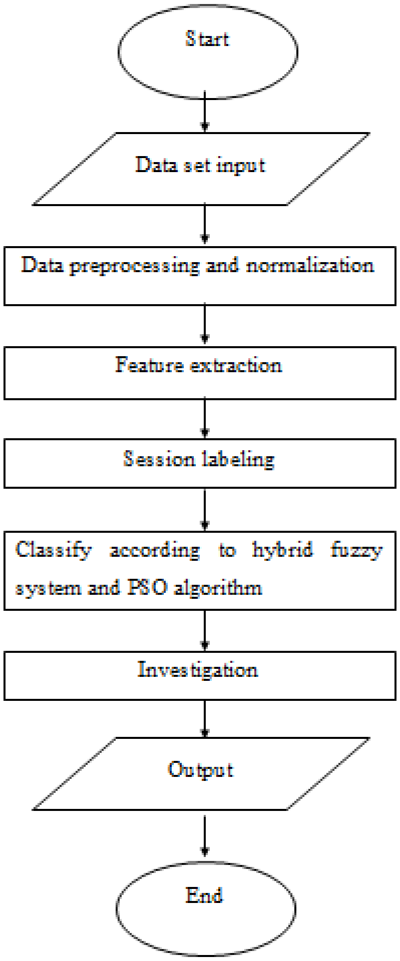
Start
Data set input
Data preprocessing and normalization
Feature extraction
Session labeling
Classify according to hybrid fuzzy system and PSO algorithm
Investigation
Output
End

Fig. 1 represents general flowchart of the research model.

The following describes each step:

○ *preprocess*

Detecting robots preprocessing includes determining existing sessions within the blog being organized by classified sessions based on IP address and user information. Each new session is created based on a specific time interval.

○ *Feature extraction*

Web user categorization operation by proposed model is considered by several features. Some features applied in resources are [6, 7] some of which are not applied here. The features above are mentioned according to dataset dominant academic samples. Features mentioned within the research are:

- ❖ Total time: It refers to the time between the first and last request. The longer the time, the closer is the user to the robot.
- ❖ Maximum click rate: HTML refers to the maximum request files. The higher the request file, the closer is user to the robot.
- ❖ "robot.txt" Requesting robots: Common robots request this file for information and accessing a specific site. For example a robot asking for information from "www.google.com" initially requests: "www.google.com/ robot.txt." session labeling.
- ❖ Head type HTTP request percentage: Web robots send a title to the server to reduce the amount of information requests. This is when human users send the request to the server using browsers using GET request model.
- ❖ Percentage of requests with blank citations: A feature is a number representing request percentage with blank citations within a user session. The feature is high in case of some robots since web browsers form feed some information by default as citations.
- ❖ Css file requests: Web browsers automatically send a file to css file while web robots don't require seeing the css file. So if the session requests are the entire same source, at the same time css file has not been viewed, then the session is suspected of web robots.
- ❖ Requested data volume: This numerical feature conveys requested data volume of a session in bytes, the higher it is, and the closer is the visitor to the robot.
- ❖ Sequential request rate: a feature is a number representing a percentage of sequential requests for the pages belong to the same web path over the session. For instance the request / google/ translate/ **is considered a sequence of HTTP request.
- ❖ Error response: Web robots are in danger of broken link selection than human users (to label the sessions)
- ❖ Page request depth: A feature is a number representing the page depth for the whole requests of a session. For example the request google/translate/persianpage.html " is determined to have a depth of 3 "google/translate.html/" and 2 for the request
- ❖ PDF and PSs file requests: Web robots request more for PDF and Postscript files to gather the information.

- The rate of HTML requests than images: Web robots request more HTML files than human users and on the contrary, human users request more images, so the more the ratio, the closer is the robot.
- Percentage of requests for other pages: Web browsers can automatically view the whole embedded resources within a web page. If a session consists of only one requested resource, then the session is suspected of a web robot. Hence, if a web page is viewed without seeing the resources embedded, then the session is suspected of a web robot.
- Percentage of cookies: Cookies are information that HTTP server sends the user's machine along with requested source. The user may save the information and subsequently send it to the server while sending extra requests. So if the percentage of cookies belonging to a session is zero, then the session is suspected of a robot.
- XX Error percentage: Robots are highly in danger of errors when requesting than humans.
- Depth standard deviation for the requested page it's a numerical computed feature among the entire sent requests during a session considered for that page.
- Total requested pages: The number of requested pages including (PDF, PS, HTML and ASP) files of a session. The total requested a web crawler is more than humans.
- Session length: Total available HTML numbers regardless the subordinate requests (such as the request for each saved image in a file log on HTML pages).
- Percentage of request repetition: Repetition of web robots is more compared to human requests.
- Bandwidth
- Web and document: Web pages such as (PHP, ASP and HTML), textual and non- textual documents (doc, ppt, pdf, and ps) are rather documents for the web robots.
- Script: Script files (JS, Css...) and searched pages by humans maintain CSS embedded codes through the browsers. Hence some requests have been added for CSS files to save logs but web robot sessions don't include theses requests. The differences work for non-robot sessions.
- Image: Image files (jpg, bmp, gif, png …) The requests for human images is higher than web robots. Also image files are embedded without user's knowledge on web pages. On the other hand, some web robots aim at collecting images. The mentioned robots make their requests during a session.
- Multimedia: Include music (wma, mp3, mid …), video, animation file (swf, avi, mpeg...). Except web robots aim at aggregating mentioned files is higher than human requests.
- Download compact files (rar, zip, exe, …)
- Hit: Total session requests regardless of their type or return.
- Number of unique pages: Text pages which are requested first time and without repetition.

- *Session labeling*

After feature determination for each session, it's time to label them. During the first step, whole sessions with equal "robot.txt." features, are considered as robots. Later, whole the left user agent sessions, are compared with user agent string list of known webs, if the user agent string matches the updated list of users of web robots, then the session itself is considered as a robot. If user agent string matches web browsers and the file “robot. Txt" is not viewed, then the session is labeled as human.

- *Classification using hybrid fuzzy system and PSO algorithm*

Fuzzy system and PSO algorithm are combined as follows:

1. Making a basic fuzzy system
2. Parameter adjustment of basic fuzzy system regarding the modeling error using optimization algorithm.
3. Assembling fuzzy system with the best values as the final result (the best parameters carry the least errors).
4. If the condition comes true (which can be the error or repetition times) then, move to the next step. Otherwise, go back to step 2.
5. 5. The end

Hybrid classification algorithm flowchart of the study is represented in Fig 2.

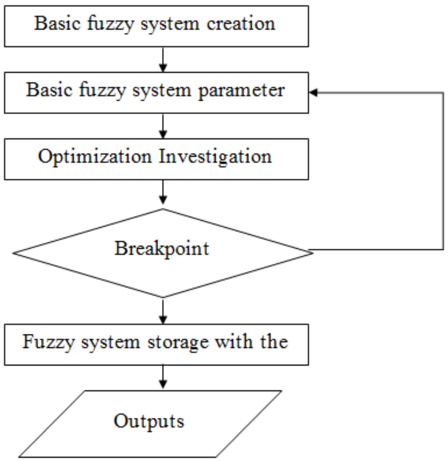
Basic fuzzy system creation
Basic fuzzy system parameter
Optimization Investigation
Breakpoint
Fuzzy system storage with the
Outputs

Fig. 2 The flowchart of the used combination method in this study

Fuzzy system training in this paper is conducted using PSO algorithm. To do it, fuzzy system training is converted to an optimization issue which will be solved using PSO algorithm. There should be a fuzzy system first. Here we applied Sognou for fuzzy systems. In this paper we use PSO algorithm to train the system and achieve better results. To optimize we use error criterion system. Using this algorithm, the errors increases/ decreases and obtain the optimal parameters. By using optimal values offered by PSO, better results are obtained.

6. Analyses

Table 1 represents dataset details. The data is one-month university file log, including 18742 sessions among which 6395 cases include web crawlers and the rest, humans.

Table 1: The details of the used dataset in this study

Total Number of Sessions	*Number of Crawler Sessions*	*Number of Human Sessions*
18742	6395	12347

2224 records have been used in this paper. The overall variables include: Request percentages consisting blank citations, maximum click rate, session length, image, web and document, 4XX error percentage, "robot.txt." request, HTTP request of HEAD type, sequential rate request, error response, standard deviation of the requested page depth, css file requests, percentage of other requested pages, percentage of cookies and requested data volume which were explained in section 2.5. Of the output features is robot or non-robot which determines human being robot or non-robot. Here the feature 1 indicates robots and zero being human.

We have considered 84% of the entire dataset (1938 records) as the educational dataset and the rest (484) as the test dataset. The applied method of the paper, as described in proposed methodology, we classify according to hybrid fuzzy system and particle optimization algorithm. Such that PSO algorithm is applied to train the fuzzy system. The Sougno phase is used as Fis by the two rules as:

$$\text{Rule 1: If } x \text{ is } A_1 \text{ and } y \text{ is } B_1 \text{ then } f_1 = p_1 x + q_1 y + r_1$$

$$\text{Rule 2: If } x \text{ is } A_2 \text{ and } y \text{ is } B_2 \text{ then } f_2 = p_2 x + q_2 y + r_2$$

In which A, B are entity membership functions, $p_1, q_1, r_1, p_2, q_2, r_2$ are the output parameters.

The initial values of p_1, q_1, r_1, p_2, q_2 & r_2 parameters are randomly adjusted within the initial repetition of fuzzy system and PSO parameters are selected and adjusted as well. The combined model used here, the parameters are considered as coefficient of their initial values. Based on 5, the optimal values of coefficients are obtained by PSO values.

$$p_j^{opt} = \alpha * p_j^0 \quad ; \quad \alpha \epsilon [10^{-\beta}, 10^{\beta}] \quad ; \quad 0 < \beta < 1 \tag{5}$$

In which p_j^{opt} is the optimal parameter value and p_j^0 is the initial value of the parameter α is the fixed coefficient at a distance $[10^{-\beta}, 10^{\beta}]$ And β is the actual number within 0, 1 distances.

Here based on trial and error results, β is considered as 1. But α in a random range $[10^{-\beta}, 10^{\beta}]$ is selected based on the algorithm which is updated during each repetitive PSO algorithm.

Target function here is RMSE and error rates can obtain the minimum relative value through the optimum initial population size and the repetition numbers within hybrid algorithm of fuzzy systems and PSO algorithm. For the initial population size and repetition numbers based on trial / error method we concluded that in the research application, the amounts 100, 500 are respectively optimum. So the values were selected for the study.

Due to different results of each application, the algorithm was applied 10 times and the average results were reported as the final yield.

The evaluation criteria used based on the algorithm efficiency, are:

Precision: A general metric used to measure the usefulness of the proposed criteria. The formula of which is shown in equation 6.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{6}$$

Recall: A general metric to measure the usefulness of the proposed criteria.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{7}$$

F-Measure: Another investigation metric obtained through Recall and Precision.

$$F - \text{Measure} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

Also two are measure are accuracy and specificity which are respectively represented in the equations 9, 10.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \tag{9}$$

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{10}$$

TP equals the sample numbers which belong to a non-robot class and the algorithm proposed "non-robot" for them. FP the amount of samples belongs to "non-robot" class and the algorithm proposed "robot" class for them. TN sample numbers which belong to "robot" class and the algorithm predicted "robot" class for them. FN sample numbers belong to "robot" class and the algorithm proposed "non-robots" for them. Due to the differences each time, the algorithm was applied 10 times and the mean was considered as the final value.

Table 2 shows the proposed values for the mentioned criteria which are presented for the proposed method. Values are in percentage.

Table 2: Results obtained through the proposed methods over dataset of this research

	Proposed Method
Precision	99.58
Recall	91.39
F-Measure	96.10
Accuracy	96.92
Specificity	99.81

Authors and A Here to compare the results, the base fuzzy system and ANFIS neuronal fuzzy network system are applied. Table 3 shows the comparison between the proposed and comparison method for the considered criteria of the research.

Table 3: Comparison of the proposed method and algorithm for comparison in this research

	FUZZY	**ANFIS**	*Proposed Method*
Precision	97.43	98.23	99.58
Recall	85.09	89.15	91.39
F-Measure	93.67	94.59	96.10
Accuracy	94.50	94.31	96.92
Specificity	98.93	99.02	99.81

In the charts 3-8, the algorithms are proposed in terms of accuracy, Precision, F-measure, Recall and Specificity.

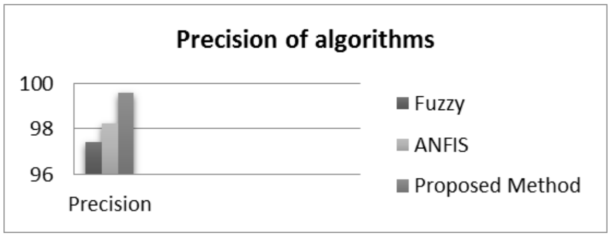

Fig. 3 Comparison of algorithms from the point view of precision (in percentage)

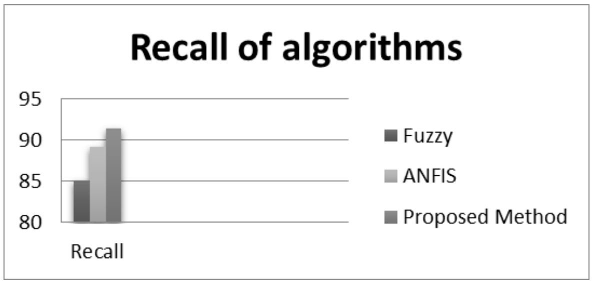

Fig. 4 Comparison of algorithms from the point view of Recall (in percentage)

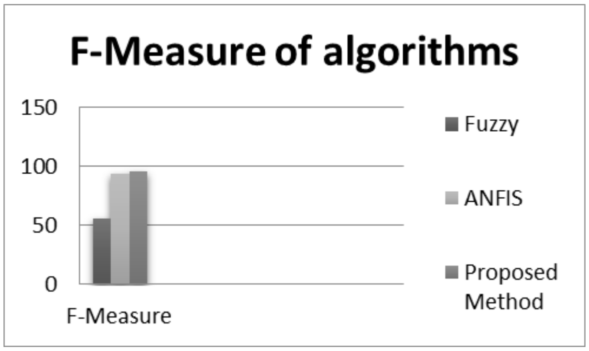

Fig. 5Comparison of algorithms from the point view of F- Measure (in percentage)

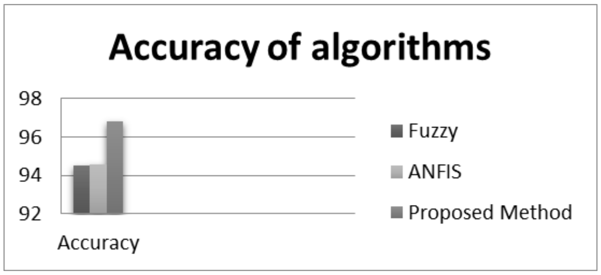

Fig. 6 Comparison of algorithms from the point view of Accuracy (in percentage)

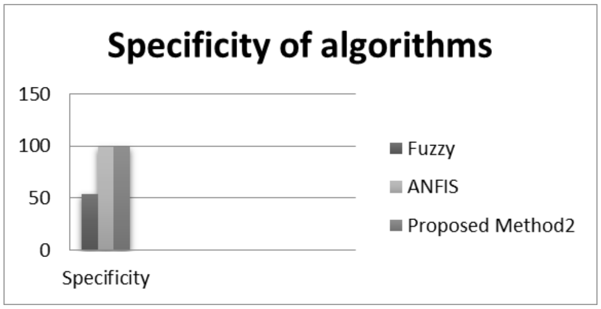

Fig. 7 Comparison of algorithms from the point view of Specificity (in percentage)

As we see in table 3 and Fig 3-8, the proposed values are higher than the compared methods. The results indicate that the proposed approach here, works better than comparable approaches.

7. Conclusions

This paper aimed at providing a hybrid method to improve web robots accuracy. Of the features of proposed method than similar works we consider:

- Providing a novel method to detect Web robots influence
- Improve efficiency detecting web robots

The paper investigated through MATLAB software, the results were compared using algorithms. To test the performance of the proposed method some criteria including Precision, Accuracy, F-Measure, Recall and Specificity were considered. Conducting experiments yielded that the proposed algorithm was better than the other ones.

About the research area, integrated and unit data are used in all scientific articles but the content is not always available which is considered as limitation for our research. Based on the results the most important proposals which can be considered as the framework of future research are:

- *Applying a fuzzy decision tree to detect web robots instead of the proposed method of the research*
- *Applying other evolutionary algorithms than PSO to train basic fuzzy systems and using them in web robot detection operations*
- *Use a feature selection algorithm before data classification*
- *Consider other criteria except the ones used in this research*

References

[1] D. Doran, S.S. Gokhale, "A classification framework for web robots", Journal of the Association for Information Science and Technology, 63(12), 2549-2554, 2012.

[2] G. Klir, B. Yuan, "Fuzzy sets and fuzzy logic", (Vol. 4), New Jersey: Prentice hall, 1995.

[3] B. Han, X. Bian, "A hybrid PSO-SVM-based model for determination of oil recovery factor in the low-permeability reservoir", Petroleum, 2017.

[4] M. Hasanipanah, A. Shahnazar, H. B. Amnieh, D.J. Armaghani, "Prediction of air-overpressure caused by mine blasting using a new hybrid PSO–SVR model", Engineering with Computers, 33(1), 23-31, 2017.

[5] R. Dong, J. Xu, B. Lin, "ROI-based study on impact factors of distributed PV projects by LSSVM-PSO", Energy, 124, 336-349, 2017.

[6] D. Stevanovic, A. An, N. Vlajic, "Feature evaluation for web crawler detection with data mining techniques", Expert Systems with Applications, 39(10), 8707-8717, 2012.

[7] D. Doran, S. S. Gokhale, "Web robot detection techniques: overview and limitations", Data Mining and Knowledge Discovery, 22(1-2), 183-210, 2011.

[8] A. Kalantari, A. Kamsin, S. Shamshirband, A. Gani, H. Alinejad-Rokny, A. T. Chronopoulos, "Computational intelligence approaches for classification of medical data: State-of-the-art, future challenges and research directions", Neurocomputing, 2017.