Procedia
Environmental Science, Engineering and Management
http://www.procedia-esem.eu

Procedia Environmental Science, Engineering and Management **7** (2020) (4) 621-628

International Conference on Agriculture, Environment and Allied Sciences (AEAS), December 24th-25th, 2020, Istanbul, Turkey

CREATION OF A FUZZY NEURAL NETWORKS TO ASSESS ENVIRONMENTAL SAFETY*

Oleg Yuryevich Panischev[1], Ekaterina kolaevna Ahmedshina[1], Dina Vladimirovna Kataseva[2], Alexey Sergeevich Katasev[2], Amir Muratovich Akhmetvaleev[2]**

[1]Kazan Federal University, 18 Kremlyovskaya Street, Kazan, Republic of Tatarstan, Russia
[2]Kazan National Research Technical University named after A.N. Tupolev, Ulitsa Karla Marksa, 10, Kazan, Republic of Tatarstan, Russia

Abstract

The article focuses on the problem of assessing the level of the fuzzy phenomena as environmental safety territorial entities. Application of neural networks allows to overcome the lack of available information on the input and to carry out the correct assessment of the environmental safety of territorial entities. The expediency of solving the problem on the basis of machine learning methods - fuzzy neural networks - is substantiated. A set of initial data used to construct neuro-fuzzy models is described. Total 20 input parameters and one output parameter are used in the dataset. The input set of parameters was reduced to 8, and the amount of data for training was 1733 records after performing the appropriate data pre-processing procedures related to the exclusion of insignificant input parameters and after the reduction of the input feature space based on the correlation analysis results, as well as the exclusion of outliers in the data. The Rapid Miner Studio analytical platform was used to prepare the initial data. A specially developed software package was used as a tool for analysing the prepared data and forming a fuzzy model for assessing environmental safety; the software implements the process of training fuzzy neural networks forming a model of a collective of fuzzy neural networks and a fuzzy knowledge production base for classification. As a result of training, a fuzzy model with a knowledge base containing 13122 fuzzy production rules was formed. The results of testing the knowledge base showed its adequacy and the achieved classification accuracy of 95.33%. The achieved accuracy exceeds the accuracy of other classification models based on the same input data. Thus, the constructed fuzzy model can be effectively used to identify environmental issues.

*Selection and peer-review under responsibility of the AEAS Conference
**Author to whom all correspondence should be addressed: opanischev@mail.ru

Keywords: environmental issues, environmental safety, fuzzy model, fuzzy neural network, knowledge base.

1. Introduction

Currently, web browsers have become the main desktop applications and, at the same time, the main point of entry for many computer attacks aimed at intercepting personal data and manipulating users to obtain confidential information (Zamfira et al., 2019). Browsers collect information such as favourite sites, cache files, cookies, browsing history, form filling data, and passwords. If an unsuspecting Internet user visits a malicious site (Kent and Liebrock, 2013), its scripts are usually immediately launched to install rogue programs, steal personal data, or even use the user's machine as part of a botnet to carry out future attacks.

In this regard, an urgent task is the early detection of malicious sites (Hirose and Suzuki, 2005). To solve this problem, services of the "black list" type (Ma et al., 2009) are usually used, which are embedded in browsers and search systems. Databases of malicious sites for such services are generated by manually checking sites by multiple users, using web crawlers or honeypots (Vishal et al., 2017). These services can offer mean accuracy in detecting malicious sites, as any database is limited and new malicious sites appear every day. It is also possible to identify non-dangerous sites as malicious.

Malicious sites can vary in the way they attack users. In general, they can be divided into two main categories (Chuchuen and Chanvarasuth, 2015; Vishal et al., 2017): sites with malicious software and phishing sites. The first type of sites carries out attacks by downloading virus code to users' computers by secretly downloading files, exploiting vulnerabilities in browsers, or through malicious JavaScript code. Phishing is based on social engineering, so users willingly pass their information on to an attacker. The main attack vector of such sites are users themselves. Phishing is designed to convince users to perform certain actions: enter their personal data, click on a specific link, etc. This is often achieved by making phishing sites look like copies of legitimate ones.

Therefore, it is necessary to ensure that those websites are checked before their visiting for malware automatically and invisibly to users. For this, the development of effective methods, models and algorithms (Dagaeva et al., 2019; Perfilieva et al., 2016; Emaletdinova and Kabirova, 2019; Wheatcroft and Walklate, 2014), as well as their practical implementation and use in web browsers, is relevant.

2. Methods

There are many methods for detecting malicious sites (Rajitha and VijayaLakshmi, 2016). One of the first is the method based on the use of a "black list" (Rao and Pais, 2017). A blacklist is a list containing information about the IP address, website names, or URLs of known malicious websites. Examples of blacklisted sites are phishtank.com and vxvault. These sites provide a reliable check on whether a site is malicious as the information is based on user reviews. Although such lists are highly reliable, the speed at which they are updated is slow. In general, it takes a long period of time to search, check and blacklist a site.

In addition to blacklisting, there are also proactive methods for detecting malicious sites: using honeypot clients, machine learning (Ismagilov et al., 2019; Kawaguchi and Ozawa, 2019; Kosuri and Nagasri, 2020; Singh and Ashraf, 2019; Zhang et al., 2018), and page content analysis (Bannur et al., 2011). In general, the existing methods can be divided into two categories (Kim and Hawkins, 2013; Uitto et al., 2017):

1) static (detection of malicious sites by analysing their URL);

2) dynamic (detection of malicious sites by analysing their behaviour).

Let's consider the features and disadvantages of traditional methods for detecting malicious sites (Table 1).

Comparing the considered methods, we can conclude that at present it is most relevant to use machine learning methods to solve the problem (Akhmetvaleev and Katasev, 2018; Satapathy et al., 2019). However, it should be noted that it is advisable to choose those among this group of methods that, in addition to determining the site for malware, are able to explain the result obtained, that is, to make the solution of the problem transparent to the user. This requirement is largely satisfied by fuzzy neural networks (ALmomani et al., 2012; Katasev, 2019) and fuzzy models built on their basis (Chupin et al., 2019) for checking sites for malware.

Table 1. Comparative analysis of methods for detecting malicious sites

Method	***Features of the method***	***Disadvantages of the method***
"Black list"	- uses a pre-compiled list of known malicious sites; - the accuracy and reliability of the definition are high and based on feedback from many users.	- limited resources and the ability to add sites to lists that require periodic updates; - ease of bypassing "blacklists" by making changes to the original URL.
Honeypot client	- scans the Internet and detects malicious sites in low or high interaction mode.	- can be easily detected by the owners of malicious sites.
Machine learning	- uses existing information from the URL and develops an adaptive model for checking sites for malware.	- difficulties in finding high-quality initial data for training.
Analysing web page content	- checks the content of the page and performs calculations to compare against legitimate pages and a set of rules.	- takes a long time to check.

In this study, the collection and preparation of initial data was performed for fuzzy neural networks training. The dataset retrieved from the site Kaggle (Hu et al., 2019) contained application layer characteristics and network characteristics of 1.781 legitimate and malicious sites. Total dataset used 20 input parameters and one Type output parameter, which defines the object class.

Input parameters are as follows:

1) URL: anonymized representation of the parsed URL;

2) URL_Length: the number of characters in the URL;

3) Number_Special_Characters: the number of special characters (/, \%, \ #, etc.) in the URL;

4) Charset: character encoding of the content;

5) Server: The operating system on which the site runs;

6) Content_Length: the size of the HTTP header content;

7) Whois_Country: country where the website is located based on the Whois API;

8) Whois_Statepro: country from which web site responses came;

9) Whois_Regdate: server registration date;

10) Whois_Updated_Data: last server update;

11) TCP_Conversation_Exchange: the number of TCP packets exchanged between the server and the honeypot client;

12) Dist_Remote_TCP_Port: the number of detected dedicated ports;

13) Remote_IPS: the total number of IP addresses connected to the honeypot;

14) App_Bytes: the number of bytes transferred;

15) Source_App_Packets: the number of packets sent from client to server;

16) Remote_App_Packets: the number of packets received from the server;

17) Source_App_Bytes: the number of bytes in the sent packets;

18) Remote_App_Bytes: number of bytes in received packets;

19) App_Packets: the total number of IP packets generated between the honeypot client and server;

20) DNS_Query_Times: Number of generated DNS packets.

The Type output parameter determines the class of the site: 1 - malicious, 0 - legitimate.

To prepare the initial data, the analytical platform Rapid Miner Studio (Devipriya, 2019) was used. Several columns were removed from the original dataset in the process of preparing data for analysis: "URL", "Whois_Regdate", "Whois_Updated_Date", "Whois_County", "Whois_Statepro", "Content_length". The URL column was removed because it contained unique data to anonymize URL addresses in those data. The "Content_Length" column was removed because its 812 values were empty. The rest of the columns containing information about the server were also removed, since each of them had many unique named values (about 200 or more in each). 13 input parameters remained in the table after removing these columns.

Further preprocessing of the remaining data was associated with conducting a correlation analysis of the input parameters against the output to assess their information content and reduce the dimension of the input feature space (Jebarathinam et al., 2020). Fig. 1 shows the results of the correlation analysis.

Attributes	URL_LE...	NUMBE...	TCP_C...	DIST_R...	REMOT...	APP_BY...	SOURC...	REMOT...	SOURC...	REMOT...	APP_P...	DNS_Q...	Type
URL_LENGTH	1	0.914	-0.044	-0.039	-0.057	-0.027	-0.048	-0.039	-0.019	-0.028	-0.048	-0.074	-0.173
NUMBER_SPECIAL_CHARACTERS	0.914	1	-0.042	-0.042	-0.058	-0.025	-0.045	-0.035	-0.019	-0.025	-0.045	-0.055	-0.292
TCP_CONVERSATION_EXCHANGE	-0.044	-0.042	1	0.556	0.331	0.458	0.998	0.991	0.865	0.459	0.998	0.349	0.040
DIST_REMOTE_TCP_PORT	-0.039	-0.042	0.556	1	0.211	0.781	0.559	0.592	0.314	0.782	0.559	0.259	0.083
REMOTE_IPS	-0.057	-0.058	0.331	0.211	1	0.023	0.361	0.304	0.172	0.025	0.361	0.548	0.081
APP_BYTES	-0.027	-0.025	0.458	0.781	0.023	1	0.446	0.469	0.074	1.000	0.446	0.012	0.011
SOURCE_APP_PACKETS	-0.048	-0.045	0.998	0.559	0.361	0.446	1	0.989	0.857	0.448	1	0.410	0.034
REMOTE_APP_PACKETS	-0.039	-0.035	0.991	0.592	0.304	0.469	0.989	1	0.880	0.471	0.989	0.355	0.032
SOURCE_APP_BYTES	-0.019	-0.019	0.865	0.314	0.172	0.074	0.857	0.880	1	0.075	0.857	0.215	0.043
REMOTE_APP_BYTES	-0.028	-0.025	0.459	0.782	0.025	1.000	0.448	0.471	0.075	1	0.448	0.016	0.011
APP_PACKETS	-0.048	-0.045	0.998	0.559	0.361	0.446	1	0.989	0.857	0.448	1	0.410	0.034
DNS_QUERY_TIMES	-0.074	-0.055	0.349	0.259	0.548	0.012	0.410	0.355	0.215	0.016	0.410	1	-0.069
Type	-0.173	-0.292	0.040	0.083	0.081	0.011	0.034	0.032	0.043	0.011	0.034	-0.069	1

Fig. 1. Correlation matrix in the Rapid Miner program

It can be seen that the NUMBER_SPECIAL_CHARACTERS and URL_LENGTH input parameters have the greatest correlation with the output parameters. It is also seen that many of the input parameters are strongly correlated with each other. This means the redundancy of the input feature space. After reducing it, there are 8 input parameters left: URL_Length, Number_Special_Characters, TCP_Conversation_Exchange, Dist_Remote_TCP_Port, Remote_IPS, App_Bytes, App_Packets, DNS_Query_Times.

Also, outliers are searched for and eliminated in the initial data. Outliers were determined based on the distance from a point to its nearest neighbour *k*. Each point was ranked based on its distance to the *k*-th nearest neighbour, and the top n points in this ranking were declared outliers.

After deleting rows that were marked as outliers, 1.733 objects remained in the data table, 1.528 of which were marked as legitimate, and 205 as malicious. The undersampling technique was used to correct data imbalances. To do this, so many objects were randomly removed from class "0" that the resulting dataset had contained an equal number of examples from both classes.

A specially developed software package (Katasev, 2019) was used as a tool for analysing the prepared data and forming a fuzzy model for checking sites for malware; the package implemented the process of training fuzzy neural networks forming a model of a collective of fuzzy neural networks and a fuzzy production knowledge base for classification.

To determine the structure of fuzzy neural networks in the software package, the following parameters were set:

- number of gradations of input neurons: 3;
- membership function: triangular;
- granulation method: k-mean values.

The process of fuzzy neural networks training was carried out using a genetic algorithm (Katasev, 2019) with the following characteristics:

- size of the initial population of chromosomes: 100;
- type of crossing over: two-point with floating points;
- probability of mutation: 2%.

The bootstrap error (Efron and Tibshirani, 1997) was used as a criterion of neural networks training efficiency; which is a linear convolution consisting of training and testing errors of the constructed models.

3. Results and discussion

Training of fuzzy neural networks ended with the following result:

- classification error on the training sample: 0.02;
- classification error on the testing sample: 0.03;
- model's bootstrap error: 0.0266;
- training time: 11:08:07.

In addition, 75.004 cycles of the genetic algorithm were implemented during the training of fuzzy neural networks. As a result of training, a fuzzy model was formed with a knowledge base containing 13122 fuzzy production rules. A fragment of the generated knowledge base is shown in Fig. 2.

	APP_PACKETS	DIST_REMOTE	REMOTE_IPS	DNS_QUERY_T	TCP_CONVERS	Type
▸	1(w=0.515)	1(w=0.818)	1(w=0.495)	1(w=0.424)	1(w=0.606)	0(CF=0.239)
	1(w=0.515)	1(w=0.818)	1(w=0.495)	1(w=0.424)	1(w=0.606)	1(CF=0.019)
	1(w=0.515)	1(w=0.818)	1(w=0.495)	2(w=0.323)	1(w=0.606)	1(CF=0.151)
	1(w=0.515)	1(w=0.818)	2(w=0.374)	1(w=0.424)	1(w=0.606)	0(CF=0.043)
	1(w=0.515)	1(w=0.818)	2(w=0.374)	2(w=0.323)	1(w=0.606)	1(CF=0.019)
	1(w=0.515)	2(w=0.152)	2(w=0.374)	1(w=0.424)	1(w=0.606)	0(CF=0.087)
	1(w=0.515)	2(w=0.152)	3(w=0.131)	1(w=0.424)	1(w=0.606)	0(CF=0.022)
	1(w=0.515)	2(w=0.152)	3(w=0.131)	1(w=0.424)	2(w=0.313)	0(CF=0.022)
	2(w=0.404)	1(w=0.818)	2(w=0.374)	2(w=0.323)	1(w=0.606)	0(CF=0.022)
	2(w=0.404)	1(w=0.818)	2(w=0.374)	2(w=0.323)	1(w=0.606)	1(CF=0.075)
	2(w=0.404)	1(w=0.818)	2(w=0.374)	3(w=0.253)	1(w=0.606)	0(CF=0.065)
	1(w=0.515)	1(w=0.818)	1(w=0.495)	2(w=0.323)	1(w=0.606)	0(CF=0.022)
	2(w=0.404)	1(w=0.818)	1(w=0.495)	2(w=0.323)	2(w=0.313)	0(CF=0.022)
	2(w=0.404)	1(w=0.818)	1(w=0.495)	3(w=0.253)	2(w=0.313)	1(CF=0.038)

Fig. 2. Fragment of the generated knowledge base

Knowledge base rules are a set of conditions corresponding to each of the 8 input parameters, and a conclusion corresponding to the output parameter. To assess the adequacy of the constructed fuzzy model, it is necessary to investigate the knowledge base formed as a

result of fuzzy neural networks training. A testing data sample consisted of 300 records was used for this purpose.

Having checked the trained model on a testing sample, it is possible to assess whether it has acquired the ability to predict the class of an object according to the previously specified characteristics. If the classification error on the testing sample is not large, it indicates that the model has acquired the ability to generalize, that is, it effectively solves the problem. The Table 2 shows the results of testing the model.

Table 2. Matrix of errors when testing the fuzzy model

In fact	***Defined by the model***	
	0	***1***
0	139	11
1	3	147

Based on the generated knowledge, only 286 of 300 objects were correctly classified. The testing sample included 150 objects from each class: "legitimate site" and "malicious site". The table on the left shows the actual class label and on the right there is the model-defined one. Class "Legitimate" has been assigned the number "0", and the class "Malicious" has been assigned the number "1".

Let's calculate the accuracy of the generated model using the formula (1) (Mustafin et al., 2018):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (1)$$

Where, TP is the number of true positive classification results, TN is the number of true negative classification results, FP is the number of false positive classification results, and FN is the number of false negative classification results.

The resulting assessment of the classification accuracy was 95.33%, which is a fairly high result. To evaluate the obtained result, let us compare the accuracy of the generated fuzzy model with the accuracy of other classification models. For example, (Chan, 2020) presents simulation results. The author used 4 models. The research was carried out on the same data that were used in this work.

Here are the results of evaluating the accuracy of classification models:

- multilayer perceptron - 83%;
- decision tree - 87.8%;
- “k-nearest neighbours’ method” - 91.47%;
- “random forest” - 95%.

Thus, the classification accuracy obtained on the basis of a fuzzy knowledge base exceeds the accuracy of other known classification models based on the same input data. Consequently, the fuzzy model constructed as a result of fuzzy neural networks training is adequate and can be effectively used to check sites for malware on the Internet. In addition, the generated knowledge base is transparent to the user and makes it easy to interpret the output result of site classification.

4. Conclusions

As the study showed, it is advisable to use modern machine learning methods such as fuzzy neural networks to check sites for malware. Training from the initial data, they are able to form a fuzzy knowledge base for classification, which allows to interpret the output result. The results of the conducted studies indicate that the constructed fuzzy model is an effective

tool for checking sites for malware. It allows us to solve the problem with a high degree of accuracy. Thus, the work has solved the problem of mathematical modelling to construct a fuzzy model for checking sites for harmfulness based on training fuzzy neural networks. The results of the experiments have shown the effectiveness of the proposed approach to solving the problem.

The constructed model has shown high recognition accuracy. This indicates its effectiveness and the possibility of practical use for checking sites for malware in the Internet.

Acknowledgements
The work is performed according to the Russian Government Program of Competitive Growth of Kazan Federal University.

References

Akhmetvaleev A.M., Katasev A.S., (2018), Neural network model of human intoxication functional state determining in some problems of transport safety solution, *Computer Research and Modelling*, **10**, 285-293.

Almomani A., Wan T.-C., Altaher A., ALomari E., Ramadass S., (2012), Evolving fuzzy neural network for phishing emails detection, *Journal of Computer Science*, **8**, 1099-1107.

Bannur S.N., Saul L.K., Savage S., (2011), *Judging a Site by Its Content: Learning the Textual, Structural, and Visual Features of Malicious Web Pages*, Proc. of the ACM Conf. on Computer and Communications Security, 1-9.

Chan M., (2020), Classifying malicious and benign websites based on application and network features, *John Jay's Finest*, **35**, 63-72.

Chuchuen C., Chanvarasuth P., (2015), Relationship between phishing techniques and user personality model of Bangkok internet users, *Kasetsart Journal - Social Sciences*, **36**, 322-334.

Chupin M.M., Katasev A.S., Akhmetvaleev A.M., Kataseva D.V., (2019), Neuro-fuzzy model in supply chain management for objects state assessing, *International Journal of Supply Chain Management*, **8**, 201-208.

Dagaeva M., Garaeva A., Anikin I., Makhmutova A., Minnikhanov R., (2019), Big spatiotemporal data mining for emergency management information systems, *IET Intelligent Transport Systems*, **13**, 1649-1657.

Devipriya B., Kalpana Y., (2019), Evaluation of sentiment data using classifier model in rapid miner tool, *International Journal of Engineering and Advanced Technology*, **9**, 2966-2972.

Efron B., Tibshirani R., (1997), Improvements on cross-validation: The 632 bootstrap method, *Journal of the American Statistical Association*, **92**, 548-560.

Emaletdinova L.Y., Kabirova A.N., (2019), Methods of constructing the neural network models of regulators for controlling a dynamic object with smooth monotonous behavior, *Russian Aeronautics*, **62**, 213-221.

Wheatcroft J.M., Walklate S., (2014), Thinking differently about 'False Allegations' in cases of rape: The search for truth, *International Journal of Criminology and Sociology*, **3**, 239-248.

Hirose N., Suzuki E., (2005), Engineering web log for detecting malicious sessions to a web site by visual inspection, *WSEAS Transactions on Computers*, **4**, 1249-1258.

Hu Y.-H.F., Ali A., Hsieh C.-C.G., Williams A., (2019), *Machine Learning Techniques for Classifying Malicious API Calls and N-Grams in Kaggle Data-set*, Conf. Proc.-IEEE SOUTHEASTCON, 9020353, 11-14 April, Huntsville, AL, USA, http://doi.org/10.1109/SoutheastCon42311.2019.9020353.

Ismagilov I.I., Molotov L.A., Katasev A.S., Kataseva D.V., (2019), Construction and efficiency analysis of neural network models for assessing the financial condition of enterprises, *Journal of Advanced Research in Dynamical and Control Systems*, **11**, 1842-1847.

Jebarathinam C., Home D., Sinha U., (2020), Pearson correlation coefficient as a measure for certifying and quantifying high-dimensional entanglement, *Physical Review A*, **101**, 022112, https://doi.org/10.1103/PhysRevA.101.022112.

Katasev A.S., (2019), Neuro-fuzzy model of fuzzy rules formation for objects state evaluation in conditions of uncertainty, *Computer Research and Modeling*, 11, 477-492.

Kawaguchi Y., Ozawa S., (2019), *Exploring and identifying malicious sites in dark web using machine learning*, In: *Lecture Notes in Computer Science* (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Springer, 319-327.

Kent A.D., Liebrock L.M., (2013), *Statistical detection of malicious web sites through time proximity to existing detection events*, Proc.-2013 6th Int. Symp. on Resilient Control Systems, ISRCS P. 192-197.

Kim B., Hawkins P.M., (2013), Who's getting cited: Representation of women and non-white scholars in major American criminology and criminal justice journals between 1986-2005, *International Journal of Criminology and Sociology*, **2**, 306-321.

Kosuri N.K., Nagasri B., (2020), Phishing web sites features classification based on extreme learning machine, *Test Engineering and Management*, **83**, 1222-1225.

Ma J., Saul L.K., Savage S., Voelker G.M., (2009), *Beyond blacklists: Learning to detect malicious web sites from suspicious URLs*, Proc. of the ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, 1245-1253.

Mustafin A.N., Katasev A.S., Akhmetvaleev A.M., Petrosyants D.G., (2018), Using Models of Collective Neural Networks for Classification of the Input Data Applying Simple Voting, *Journal of Social Sciences Research*, **2018**, 333-339.

Perfilieva I.G., Yarushkina N.G., Afanasieva T.V., Romanov A.A., (2016), Web-based system for enterprise performance analysis on the basis of time series data mining, *Advances in Intelligent Systems and Computing*, **450**, 75-86.

Rajitha K., VijayaLakshmi D., (2016), Oppositional Cuckoo Search Based Weighted Fuzzy Rule System in Malicious Web Sites Detection from Suspicious URLs, *International Journal of Intelligent Engineering and Systems*, **9**, 116-125.

Rao R.S., Pais A.R., (2017), *An enhanced blacklist method to detect phishing websites*, In: *Lecture Notes in Computer Science* (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Springer, 323-333.

Satapathy S.K., Mishra S., Mallick P.K., Gudur R.R., Guttha S.C., (2019), Classification of Features for detecting Phishing Web Sites based on Machine Learning Techniques, *International Journal of Innovative Technology and Exploring Engineering*, **8**, 425-430.

Singh D.K., Ashraf M., (2019), Detect the phishing websites in the contex of internet security by using machine learning approach, *International Journal of Advanced Science and Technology*, **27**, 104-111.

Uitto J., Rauti S., Lauren S., Leppanen V., (2017), A survey on anti-honeypot and anti-introspection methods, *Advances in Intelligent Systems and Computing*, **570**, 125-134.

Vishal K.S., Chauhan S., Prakasha K.K., (2017), An implementation of honeypots in a cloud environment for analyzing attacks on websites, *Journal of Engineering and Applied Sciences*, **12**, 6208-6214.

Zamfira A., Fat R., Cenan C., (2019), Applying semantic web technologies to discover an ontology of computer attacks, *Scalable Computing*, **20**, 699-707.

Zhang S., Tang A., Jiang Z., Sethumadhavan S., Seok M., (2018), *Blacklist core: Machine-learning based dynamic operating-performance-point blacklisting for mitigating power-management security attacks*, Proc. of the Int. Symp. on Low Power Electronics and Design, No. a5.