International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 13, Number 12 (2020), pp. 4432-4438
© International Research Publication House. http://www.irphouse.com

Creation of a Fuzzy Model for Verification of Malicious Sites Based on Fuzzy Neural Networks

Oleg Yuryevich Panischev[1], Ekaterina Nikolaevna Ahmedshina[2], Dina Vladimirovna Kataseva[3], Alexey Sergeevich Katasev[4], Amir Muratovich Akhmetvaleev[5]

[1]Researcher, Near Space Research Laboratory, Kazan (Volga Region) Federal University; Russia.
Scopus ID: 8355604900; ORCID: 0000-0001-5490-912X

[2]Candidate of Physical and Mathematical Sciences, Senior Researcher, Near Space Research Laboratory, Kazan (Volga Region) Federal University; Russia. Scopus ID: 55488733700; ORCID: 0000-0002-8681-8131

[3]Senior Lecturer, Department of Information Security Systems, Institute of Computer Technologies and Information Security, Kazan National Research Technical University named after A.N. Tupolev-KAI; Russia.
Scopus ID: 57193401954; ORSID: 0000-0001-6141-8329

[4]Doctor of Engineering Sciences, Professor, Department of Information Security Systems, Institute of Computer Technologies and Information Security, Kazan National Research Technical University named after A.N. Tupolev-KAI; Russia.
Scopus ID: 57193408902; ORSID: 0000-0002-9446-0491

[5]Candidate of Engineering Sciences, Associate Professor, Department of Information Security Systems, Institute of Computer Technologies and Information Security, Kazan National Research Technical University named after A.N. Tupolev-KAI; Russia.
Scopus ID: 57202913457; ORCID: 0000-0003-0384-9539

Abstract

This paper solves the actual problem of checking sites for malware in the Internet. The existing methods for identifying malicious sites are analysed. 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 preprocessing 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 checking sites for malware; 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 malicious sites on the Internet.

Keywords: fuzzy model, knowledge base, checking sites for malware, fuzzy neural network.

I. 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 [1]. 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 [2], 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 [3]. To solve this problem, services of the "black list" type [4] 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 [5]. 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 [6-8]: 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 [9-11], as well as their practical implementation and use in web browsers, is relevant.

II. METHODS

There are many methods for detecting malicious sites [2, 12]. One of the first is the method based on the use of a "black list" [13]. 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 [5], machine learning [14-18], and page content analysis [19]. In general, the existing methods can be divided into two categories [20, 21]:

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 (see table 1).

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.

Comparing the considered methods, we can conclude that at present it is most relevant to use machine learning methods to solve the problem [22, 23]. 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 [24, 25] and fuzzy models built on their basis [26] for checking sites for malware.

In this study, the collection and preparation of initial data was performed for fuzzy neural networks training. The dataset retrieved from the site Kaggle [27] 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;

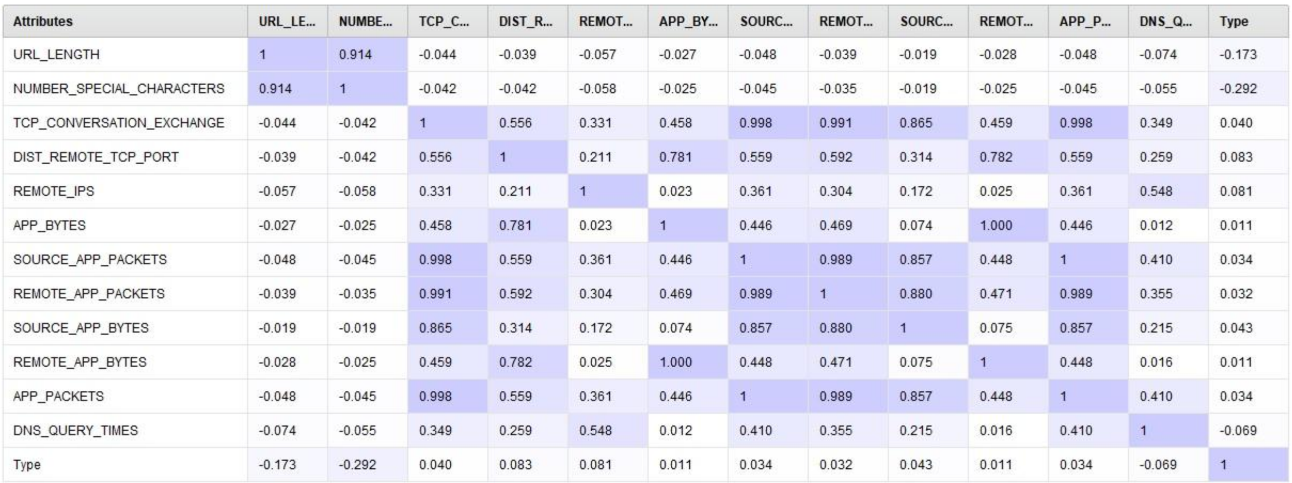
19) App_Packets: the total number of IP packets generated between the honeypot client and server;

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 [28] 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 [29]. Figure 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

Figure 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 [24] 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 [24] 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 [30] 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.

III. RESULTS AND DISCUSSION

Training of fuzzy neural networks ended with the following result:

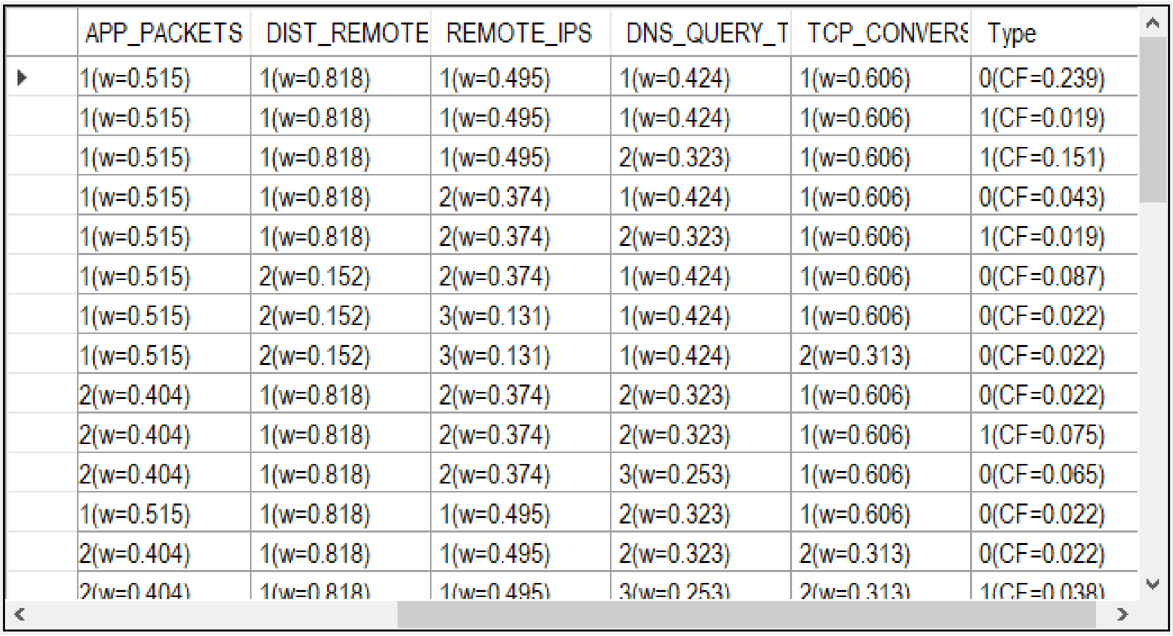
- classification error on the training sample: 0.02;

- 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 [24]. A fragment of the generated knowledge base is shown in Figure 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)

Figure 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 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 [31]:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%,$$

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, [32] 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.

IV. SUMMARY

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.

V. CONCLUSIONS

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

[1] Zamfira A, Fat R, Cenan C. Applying Semantic Web Technologies to Discover an Ontology of Computer Attacks. Scalable Computing: Practice and Experience. 2019 Dec 4;20(4):699-707.

[2] Kent AD, Liebrock LM. Statistical detection of malicious web sites through time proximity to existing detection events. In2013 6th International Symposium on Resilient Control Systems (ISRCS) 2013 Aug 13 (pp. 192-197). IEEE.

[3] Hirose N, Suzuki E. Engineering web log for detecting malicious sessions to a web site by visual inspection. WSEAS Transactions on Computers. 2005 Oct 1;4(10):1249-58.

[4] Ma J, Saul LK, Savage S, Voelker GM. Beyond blacklists: learning to detect malicious web sites from suspicious URLs. InProceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining 2009 Jun 28 (pp. 1245-1254).

[5] Vishal KS, Chauhan S, Prakasha KK. An implementation of honeypots in a cloud environment for analyzing attacks on websites. Journal of Engineering and Applied Sciences. 2017 Jan 1;12(Specialissue2):6208-14.

[6] Lakhita, Yadav S, Bohra B. Pooja A review on recent phishing attacks in Internet // Proceedings of the 2015 International Conference on Green Computing and Internet of Things, ICGCIoT 2015, 7380669: 1312-1315.

[7] Chuchuen C, Chanvarasuth P. Relationship between phishing techniques and user personality model of Bangkok Internet users. Kasetsart Journal Social Sciences. 2015 May;36(2):322-34.

[8] Fu-an Z. Phishing Sites and Prevention Measures. International Journal of Security and Its Applications. 2015 Jan 30;1(1):8.

[9] Dagaeva M, Garaeva A, Anikin I, Makhmutova A, Minnikhanov R. Big spatio-temporal data mining for emergency management information systems. IET Intelligent Transport Systems. 2019 Sep 5;13(11):1649-57.

[10] Perfilieva IG, Yarushkina NG, Afanasieva TV, Romanov AA. Web-based system for enterprise performance analysis on the basis of time series data mining. InProceedings of the First International Scientific Conference “Intelligent Information Technologies for Industry”(IITI’16) 2016 (pp. 75-86). Springer, Cham.

[11] Emaletdinova LY, Kabirova AN. Methods of Constructing the Neural Network Models of Regulators for Controlling a Dynamic Object with Smooth Monotonous Behavior. Russian Aeronautics. 2019 Apr 1;62(2):213-21.

[12] Rajitha K, VijayaLakshmi D. Oppositional Cuckoo Search Based Weighted Fuzzy Rule System in Malicious Web Sites Detection from Suspicious URLs. International Journal of Intelligent Engineering and Systems. 2016;9:116-25.

[13] Rao RS, Pais AR. An enhanced blacklist method to detect phishing websites. InInternational Conference on Information Systems Security 2017 Dec 16 (pp. 323-333). Springer, Cham.

[14] Zhang S, Tang A, Jiang Z, Sethumadhavan S, Seok M. Blacklist core: machine-learning based dynamic operating-performance-point blacklisting for mitigating power-management security attacks. InProceedings of the International Symposium on Low Power Electronics and Design 2018 Jul 23 (pp. 1-6).

[15] Ismagilov I, Molotov L, Katasev A, Kataseva D. Construction and efficiency analysis of neural network models for assessing the financial condition of enterprises. SCOPUS-2019-11-8-SID85073341944. 2019 Jan 1.

[16] Kawaguchi Y, Ozawa S. Exploring and Identifying Malicious Sites in Dark Web Using Machine Learning. InInternational Conference on Neural Information Processing 2019 Dec 12 (pp. 319-327). Springer, Cham.

[17] Singh DK, Ashraf M. Detect the phishing websites in the contex of internet security by using machine learning approach. International Journal of Advanced Science and Technology. 2019;27(1):104-111.

[18] Sönmez Y, Tuncer T, Gökal H, Avcı E. Phishing web sites features classification based on extreme learning machine. In2018 6th International Symposium on Digital Forensic and Security (ISDFS) 2018 Mar 22 (pp. 1-5). IEEE.

[19] Bannur SN, Saul LK, Savage S. Judging a site by its content: learning the textual, structural, and visual features of malicious web pages. InProceedings of the 4th ACM Workshop on Security and Artificial Intelligence 2011 Oct 21 (pp. 1-10).

[20] Uitto J, Rauti S, Laurén S, Leppänen V. A survey on anti-honeypot and anti-introspection methods. InWorld Conference on Information Systems and Technologies 2017 Apr 11 (pp. 125-134). Springer, Cham.

[21] Patil NM, Dias SP, Dcunha AA, Dodti RJ. Hybrid phishing site detection // International Journal of Advanced Science and Technology. 2020; 29(6 Special Issue):2452-2459.

[22] Satapathy SK, Mishra S, Mallick PK, Badiginchala L, Gudur RR, Guttha SC. Classification of Features for detecting Phishing Web Sites based on Machine Learning Techniques. International Journal of Innovative Technology and Exploring Engineering, volume8 (8S2). 2019:425-30.

[23] Akhmetvaleev AM, Katasev AS. Neural network model of human intoxication functional state determining in some problems of transport safety solution. Computer research and modeling. 2018;10(3):285-93.

[24] Katasev AS. Neuro-fuzzy model of fuzzy rules formation for objects state evaluation in conditions of uncertainty. Computer research and modeling. 2019;11(3):477-92.

[25] Almomani A, Wan TC, Altaher A, Manasrah A, ALmomani E, Anbar M, ALomari E, Ramadass S. Evolving fuzzy neural network for phishing emails detection. Journal of Computer Science. 2012 Jul 1;8(7):1099.

[26] Chupin MM, Katasev AS, Akhmetvaleev AM, Kataseva DV. Neuro-Fuzzy Model in Supply Chain Management for Objects State Assessing. Int. J Sup. Chain. Mgt Vol. 2019 Oct;8(5):201.

[27] Hu YH, Ali A, Hsieh CC, Williams A. Machine Learning Techniques for Classifying Malicious API Calls and N-Grams in Kaggle Data-set. In2019 SoutheastCon 2019 Apr 11 (pp. 1-8). IEEE.

[28] Devipriya B, Kalpana Y. Evaluation of sentiment data using classifier model in rapid miner tool // International Journal of Engineering and Advanced Technology. 2019;9(1):2966-2972.

[29] Jebarathinam C, Home D, Sinha U. Pearson correlation coefficient as a measure for certifying and quantifying high-dimensional entanglement. Physical Review A. 2020 Feb 24;101(2):022112.

[30] Efron B, Tibshirani R. Improvements on cross-validation: the 632+ bootstrap method. Journal of the American Statistical Association. 1997 Jun 1;92(438):548-60.

[31] Mustafin AN, Katasev AS, Akhmetvaleev AM, Petrosyants DG. Using Models of Collective Neural Networks for Classification of the Input Data Applying Simple Voting. The Journal of Social Sciences Research. 2018:333-9.

[32] Chan M. Classifying Malicious and Benign Websites Based on Application and Network. Finest. 2020 Apr:63.

About the Authors

Oleg Yurievich Panishchev is a researcher at the Near Space Research Laboratory at the Kazan (Volga Region) Federal University. He graduated from Yelabuga State Pedagogical University with a degree in Physics, Informatics and Computer Science in 2003. His research interests are Time Series Analysis, methods of non-equilibrium distributed systems analysis.

Ekaterina Nikolaevna Akhmedshina is a Candidate of Sciences in Physics and Mathematics, Senior Researcher at the Near Space Research Laboratory at Kazan (Volga Region) Federal University. She graduated from the Tatar State Humanitarian Pedagogical University with a degree in Physics and an additional specialty in computer science. She defended her Candidate thesis at KFU in 2015. Her research interests: logical operations with sets, logical operations with images in optical echo-holography, photon echo.

Dina Vladimirovna Kataseva is a postgraduate student, senior lecturer of the Department of Information Security Systems at the Institute of Computer Technologies and Information Security, Kazan National Research Technical University named after A.N. Tupolev-KAI (KNRTU-KAI). She graduated from Kazan State Institute of Finance and Economics with a degree in Accounting, Analysis and Audit in 2008, as well as graduated from the KNRTU-KAI with a master's degree in the direction of Informatics and Computer Engineering in 2018. Her research interests are intellectual analysis of time series, fuzzy logic, neural networks, and decision support systems.

Aleksey Sergeevich Katasev is a Doctor of Engineering Sciences, Professor of the Department of Information Security Systems at the Institute of Computer Technologies and Information Protection, Kazan National Research Technical University named after A.N. Tupolev-KAI (KNRTU-KAI). He graduated from the Yelabuga State Pedagogical Institute with a degree in Physics, Informatics and Computer Engineering in 2002, as well as graduated from the KNRTU-KAI with a master's degree in the direction of Informatics and Computer Engineering in 2018. He defended his doctoral thesis at KNRTU-KAI in 2019. His scientific interests are data mining technologies, formation of knowledge bases of expert systems, neural network, fuzzy and neuro-fuzzy modelling.

Amir Muratovich Akhmetvaleev is a Candidate of Engineering Sciences, Associate Professor of the Department of Information Security Systems at the Institute of Computer Technologies and Information Security, Kazan National Research Technical University named after A.N. Tupolev-KAI (KNRTU-KAI). He graduated from the Kazan State Technical University named after A.N. Tupolev in the specialty "Information security of telecommunication systems" in 2008, as well as graduated from the KNRTU-KAI with the master's degree in the direction of "Informatics and computer technology" in 2012. He defended his Candidate thesis at KNRTU-KAI in 2018. His research interests are data mining, neural network modelling, assessment of human functional states by pupillary responses to changes in illumination.