sciendo

JAISCR, 2021, Vol. 11, No. 4, pp. 331 – 339
10.2478/jaiscr-2021-0020

DECISION MAKING SUPPORT SYSTEM FOR MANAGING ADVERTISERS BY AD FRAUD DETECTION

Marcin Gabryel[1,*], Magdalena M. Scherer[2], Łukasz Sułkowski[3,4,5], Robertas Damaševičius[6]

[1]*Faculty of Mechanical Engineering and Computer Science, Czestochowa University of Technology, al. Armii Krajowej 36, 42-200 Częstochowa, Poland*

[2]*Faculty of Management, Czestochowa University of Technology, Poland*

[3]*Faculty of Management and Social Sciences, Jagiellonian University, Cracow, Poland*

[4]*Management Department, University of Social Sciences, 90 - 113 Lodz, Poland*

[5]*Clark University, Worcester, MA 01610, USA*

[6]*Department of Applied Informatics, Vytautas Magnus University, Kaunas 44404, Lithuania*

[*]*E-mail: marcin.gabryel@pcz.pl*

Submitted: November 2020; Accepted: 22nd September 2021

Abstract

Efficient lead management allows substantially enhancing online channel marketing programs. In the paper, we classify website traffic into human- and bot-origin ones. We use feedforward neural networks with embedding layers. Moreover, we use one-hot encoding for categorical data. The data of mouse clicks come from seven large retail stores and the data of lead classification from three financial institutions. The data are collected by a JavaScript code embedded into HTML pages. The three proposed models achieved relatively high accuracy in detecting artificially generated traffic.

Keywords: lead management, feedforward neural networks, embedding, online marketing

1 Introduction

The Internet advertising system consists of: (1) publishers (provide resources for advertising traffic), (2) advertisers buying traffic to deliver advertisements to recipients, (3) intermediaries (affiliate networks, media houses, programmatic platforms). Publishers both provide and generate the traffic. In other words, a user's visit to a publisher's website enables one or more advertisements to be displayed to that user. Thus, the publisher may sell this possibility to those advertisers who are interested in displaying their ads. Most often this is done through an intermediary in the form of an affiliate network. Unfortunately, to increase revenues, the parties may use unethical activities. For example, some advertisers may exhaust their competitors' marketing budgets through a carefully conducted abuse. Similarly, some publishers generate various traps which are meant to lure users to browse or click ads about products that users are not actually interested in. Affiliate networks, on the other hand, do not react to irregular clicking behaviour hoping to earn commissions. In some circumstances, malicious advertisers as well as publishers become fraudsters. In the commonly used different pricing models, despite different security measures, fraud-

sters are constantly trying to introduce new types of fraud using display, click or action-based scams.

One of the main types of crime and frauds online (adfrauds, frauds) has become in the last few years the so-called traffic scams – page views or click fraud, which consist in increasing revenues from online advertising by automatically generating artificial page views, clicks or web forms. This practice brings real (and relatively easy to obtain) income or profit as a result of losses of competing companies. On a global scale, adfrauds are causing multi-billion losses in the advertising industry. Adfraud activities may be carried out by:

1. Publishers billed in the model for the effect. This model can take, for example, a form of "cost per click" (CPC), "cost per thousand" (CPM), or "cost per lead" (CPL). A lead is a potential customer data. The most common frauds in online advertising are related to performance advertising, i.e. billed for the volume of clicks on the advertisement. In addition to "clicking ads", the effects of dishonest publishers may also be fraud leads, i.e. completing web forms with incorrect or false data, and in extreme cases, even fictitious sales resulting in the payment of an unjustified commission and loss of time for the advertiser to handle the order.

2. Intermediaries, offering support for marketing activities for advertisers, e.g. affiliate partners, owners of websites and comparison websites. Fraudulent activities include artificial generation of clicks or leads, generating worthless clicks or leads, "clicking" ads, entering false data in web forms, automatic completion of applications without the knowledge of the person concerned, creating websites generating additional clicks, and even creating special algorithms and bots.

3. Competition – striving to build a position for its ads or lower the cost per click by clicking on its competitor's ads (activities consuming the advertising budget, not bringing the expected sales results, as a result limiting the scale of the competitors' advertising campaign).

According to [4], the global costs of fraud activities are estimated between USD 6.5 and 19 billion (it is difficult to determine the exact scale of the phenomenon due to the lack of precise data). These types of unfair practices result in ineffective budget allocation for online campaigns. However, research predicts that artificial intelligence, machine learning and big data technologies will be key solutions in minimizing losses caused by fraud, thanks to the ability to analyze the huge amounts of data generated from advertising activities. AI-powered platforms will account for 74% of total online and mobile advertising spending by 2022. However, as artificial intelligence becomes saturated, only platforms with the most effective algorithms will be able to guarantee effective ad protection. According to research, these platforms will have to focus on new data sources to improve the proficiency of artificial intelligence algorithms [10]. The financial sector is one of the most attacked – every fourth advertiser is a victim of ad fraud – right behind the gambling industry and airlines [8].

The work presents two deep neural network-based classifiers with the aim to:

– evaluation of internet traffic and making a decision about the presence of a human or bot on the website for the CPC and CPM billing models, i.e. so-called monitoring of clicks. This traffic occurs when the user moves from clicking on an advertisement on the publisher's website to the advertiser's website.

– assessment of human or bot behaviour on the website in order to secure the website forms in the CPS and CPA billing model that is so-called lead monitoring. In this case, traffic goes from the publisher to the advertiser's website in order to fill out the application form (i.e. acquire a lead).

We use data downloaded from the client's website, generated after the user or bot access it. Data such as browser and device parameters, mouse pointer behaviour, keystrokes, page scrolling, screen touch position, etc., are collected using a JavaScript script included in the website's HTML code. The code is designed to retrieve information about the browser and device, leave a trace for later identification and track user behaviour. The task of the classifier is to return the probability value that will allow managing the obtained lead in order to better queue tasks related to its service (trans-

fer to the call centre for verification, transfer to the sales department to handle the service request, etc.).

This paper is organized as follows. Section 2 presents the current state of the art on internet frauds. Section 3 describes the classifier models used during in the paper. The course of experimental research can be found in Section 4. Section 5 concludes the paper and offers suggestions for future work.

2 Related works

Detecting fraud in the Internet advertising industry is one of the challenges that can be successfully solved using machine learning. The literature, however, almost lacks practical works on detecting computer programs (bots) that perform abuses related to generating artificial Internet traffic, relying only on data downloaded from a website. One of the most interesting comprehensive descriptions of these issues can be found in [14]. However, this study is limited to cataloguing threats, not mentioning the methods of counteracting them. Nevertheless, solutions that detect various bots by analyzing the webserver event log (e.g. [6]), requiring modification of the application or redirection of all traffic, are quite popular. Despite the declared effectiveness of these methods, the possibility of their practical application is negligible. Access to the information required by these solutions and modifications to the computer systems of website owners are challenging to carry out or require time and additional financial outlays, which usually discourages the use of such solutions. Therefore, the scope of research was assumed to be limited only to monitoring user behaviour on the website on the browser side. In [7] the behaviour of bots clicking on advertisements on web pages is analysed. The data comes from advertising systems offered by Google, Facebook, LinkedIn and Yahoo. In [11] botnets or DDoS attacks are identified by analyzing log files from WWW servers. Botnets are interconnected compromised machines that are used to perform attacks or filling web forms [12]. It is believed that even 16-25% machines are part of botnets [1]. One of the methods for detecting bots is presented in citesoniya2016detection where randomized bot command and control traffic is detected at an early stage. In [3] botnet-related traffic is identified by using similarity measures and periodic characteristics of botnets.

Classification is one of the functions successfully performed by feedforward neural networks (perceptron networks). Perceptron networks are the most effective model for analyzing large amounts of data stored in a tabular form. However, the most challenging task is the selection of the network structure and the selection of the network hyperparameter values. Another difficulty is that in our data, there are no labels identifying the sample as “bot” / “non-human” or “human”. Therefore, we used the data with information on sales effectiveness and/or information obtained from external sources (Google Captcha v3).

3 Decision-making Support system

As part of the research, we developed two models to classify data obtained from a website for the purpose of identifying human or program (bot) generated traffic. The models will be slightly different from each other because they relate to two different methods of customer financial billing: 1) a model for monitoring clicks and 2) a model for monitoring leads. In the first case, the emphasis of the analysis of human-bot behaviour is put on the very process of opening the page, the time spent on the page and the behaviour on the page (mainly the behaviour of the mouse pointer). In the second case, the process of filling in a web form takes place and, first of all, this process is examined (examination of the method of filling in the form by the user, including the analysis of the sequence of pressed keys and their combinations).

The method analyzes data collected during clicking on advertisements and price comparison engines. Its outcome says if the behaviour on the website is natural, generated by a human or generated artificially by a computer program (bot). A data flow diagram is presented in Figure 1. The service and database belong to the system monitoring and data collecting module. The monitoring also provides a dedicated Javascript code for the advertiser, which allows data to be collected on its website.

After clicking an advert on a publisher’s website, the advertiser’s website opens, and a script

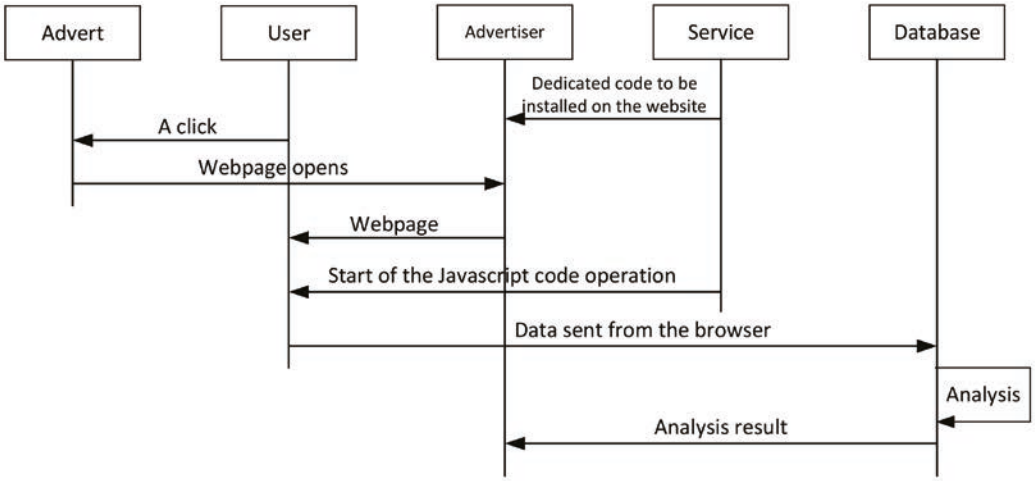

Figure 1. Data flow diagram for fraud click/leads analysis.

monitoring the client's behaviour is launched. The behaviour-related data are stored in a database and used in the paper.

The data collected during the user's visit to the monitored website are of two types: floating-point numbers and categorical data. The neural network is naturally adapted to work with floating-point numbers, so the problem arises in the case of categorical data. In such cases, the simplest method is to use one-hot encoding, where the neural network for each parameter has as many inputs as there are available categories. In other words, each category has its own input. Zeros are given for inputs, except for the input corresponding to a given category, then one is given there. This coding method was used in the first network model presented in Figure 2. It is a multilayer perceptron model consisting of two hidden layers, and the output is calculated by the softmax function. The learning set is $X = \{x_1, x_2, \ldots, x_N\} \in R^m$ where x_i is the m-dimensional feature vector, and the set of categories is $D = \{d_1, d_2, \ldots, d_N\} \in \Omega$, set of labels $\Omega = \{0, 1, 2\}$, N – the number of samples. We use multilayer perceptron which can be presented as a function

$$y = f^*(x, \Theta) \tag{1}$$

where the input vector x is mapped to the output vector y, and Θ is the set of parameters that best approximates the function. A single k multilayer perceptron layer takes the following form

$$f_k(x, \Theta) = s(\mathbf{W}x + \mathbf{b}) \tag{2}$$

where Θ is set $\{\mathbf{W}, \mathbf{b}\}$, and $s(\cdot)$ is the adopted activation function. In our model, $s(\cdot)$ is the ReLU [5] function

$$s(t) = \max\{1, 2\} \tag{3}$$

The described model consists of two layers, so the function (1) takes the form:

$$y = f^*(x, \Theta) = f^2(f^1(x, \Theta_1), \Theta_2) \tag{4}$$

where Θ_1 and Θ_2 are parameters of, respectively, the first $\{\mathbf{W}_1, \mathbf{b}_1\}$ and the second layer $\{\mathbf{W}_2, \mathbf{b}_2\}$, $\mathbf{W}_1 \in R^{m \times n_1}$, $\mathbf{b}_1 \in R^{n_1}$, $\mathbf{W}_2 \in R^{n_1 \times n_2}$, $\mathbf{b}_2 \in R^{n_2}$, n_1 and n_2 is the number of neurons in the first and second layer, respectively. The goal of the multilayer perceptron training is to minimize the difference between the obtained output of the entire y_i network and the expected output of d_i. For this purpose, the loss function is calculated according to the following formula

$$L(d_i, y_i) = \|d_i - y_i\|^2 \tag{5}$$

The aim of training is to find the optimal values of Θ_1 and Θ_2 parameters that minimize the error between the obtained output and the obtained value for the entire training set

$$\Theta_1, \Theta_2 = \underset{\Theta_1, \Theta_2}{\operatorname{argmin}} L(d, y) \tag{6}$$

There are three outputs from the network, each responsible for one of the Ω classes. For the considered category, value one should appear at the network output. Therefore, at the network output, the softmax [2] function was used, which takes the following form for each of the k outputs

$$s(t)_k = \frac{e^{t_k}}{\sum_{j=1}^{m} e^{t_j}} \tag{7}$$

where $t \in R^3$, and t_j is jth element of vector t. For softmax, the loss function $L(d_i, y_i)$ is computed by the categorical cross-entropy

$$L(d_i, y_i) = -\sum_{j=1}^{m} d_{ij} \log(y_{ij}) \tag{8}$$

where m is the number of outputs (categories).

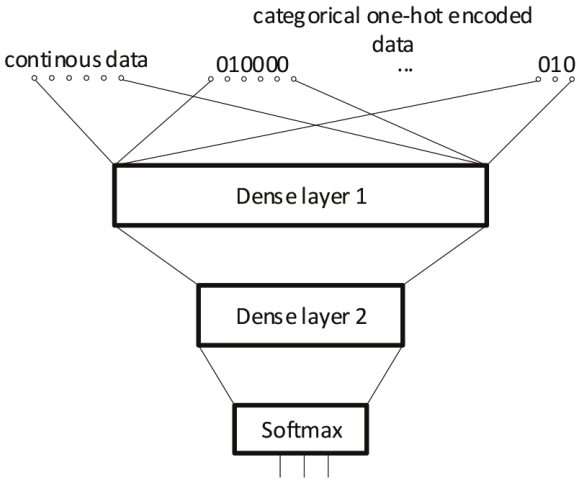

Figure 2. Multilayer perceptron network with one-hot encoded input data.

Model 2 uses embedding layers. Embedding is a mapping of separate categories (such as words) into vectors of real numbers. The rationale behind embedding is that machine learning using general patterns of locations and distances between vectors will find certain relationships between the categories. The network diagram is presented in Figure 3. The network approximates the function

$$f^*(x_r, x_c, \Theta) = f^3(f^2(\{f^1(x_r, \Theta_1), e^1(x_{c1}, \Theta_{c1}), \ldots, e^{n_c}(x_{cn_c}, \Theta_{cn_c})\}, \Theta_2), \Theta_3) \tag{9}$$

where x_r are continuous parameters (floating point) fed to the network input, $x_c \in \{x_{c1}, \ldots x_{cn_c}\}$ are categorical variables, n_c is the number of categorical variables, $e^k(x_{ck}, \Theta_{ck})$ the result of embedding layer for kth variable x_{ck}, Θ_{ck} is the set of weights $\{\mathbf{W}_{ek}\}$ for categorical variable k, $\mathbf{W}_{ek} \in R^{n_e}$, n_e is the number of neurons for every categorical variable k. Parameters Θ_1, Θ_2 and Θ_3 are sets of weights for successive layers of the neural network. The results are returned similarly to the previous model by the softmax function.

Model 3 in its structure is identical to Model 2, except that the process of training the neural network is performed differently. First, all embedding layers are trained. The neural network model is shown in Figure 4. Only the x_c categorical variables are fed into the network. The network consists of one layer and its purpose is to approximate the function

$$f^*(x_c, \Theta) = f^1(\{e^1(x_{c1}, \Theta_{c1}), \ldots, e^{n_c}(x_{cn_c}, \Theta_{cn_c})\}, \Theta_1) \tag{10}$$

where successive variables have the same notation as in Model 2. In the next training step, the embedding layers are transferred to a network with a structure identical to Model 2. The whole model is trained again, but the parameters (weights) of the embedding layers are fixed and no longer trained.

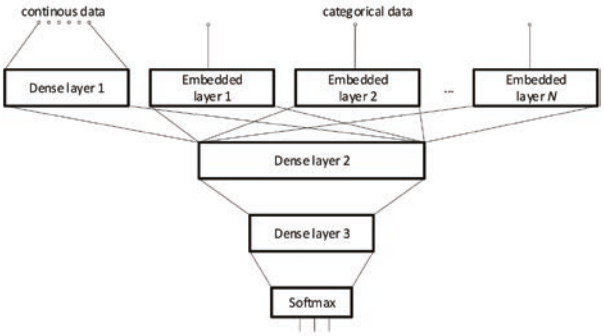

Figure 3. A multilayer perceptron network with one-hot encoded input data.

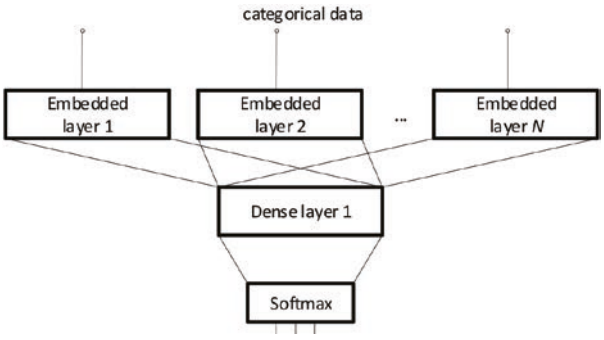

Figure 4. Embedding layer training.

4 Experiments

The experiments were carried out on two data sets:

– Monitoring of mouse clicks – the dataset contains data of 300,000 individual visits to the websites of seven online stores. The data was

obtained from the monitored website after clicking on the advertisement. The data has three labels: 'ok' - the input is correct and was made by a human, 'fake' - the input was most likely made by a bot program or unfair competition, and 'undecided' - the data is most often incomplete. Incomplete values are substituted with a default value and new input is created for a one-hot encoded set of inputs. The labels were obtained on the basis of sales and additionally assessed by an expert.

– Lead monitoring – the dataset contains data from three financial institutions from 263,000 individual visits to the website with the web form in order to fill in the data necessary to obtain a lead (interest in a loan, credit, etc.). The data has three labels: 'ok', 'fake' – the labels have a similar meaning as in monitoring clicks and 'copy' – the data is most likely copied from another database and automatically pasted into the form. The assessment was made on the basis of the sales obtained and additionally assessed by an expert.

The data is collected by a JavaScript code installed on the advertiser's website. All publicly available information that can be obtained from the browser is collected. The most important downloaded data include the following:

– For monitoring clicks: date and time of entry, source (where the traffic came from, which advertisements came from), identifier of the monitored website, type of the operating system and the browser, information whether the operating system, browser, or set languages were not counterfeited, whether mouse movements or screen scrolling were performed, information about the Internet provider, whether the connection was made through a proxy, the type of IP number (hosting, proxy, VPN, Tor network, botnet – information from an external source), whether there was a sale, whether the parameters provided by JavaScript are correct for a given browser type and are not counterfeited. In addition, information about the user's behavior is also taken into account: the number of visits to the page, the area where the mouse moved, the number of page scrolls, the number of unique mouse points indicated, the number of pressed keys.

– For monitoring leads: date and time of entry, source (where the traffic came from, what advertisements came from), identifier of the monitored website, type of operating system and browser, information whether the operating system, browser, set languages were not counterfeited, whether mouse movements or screen scrolling were performed, information about the Internet provider, whether the connection was made through a proxy, the type of IP number (hosting, proxy, VPN, Tor network, botnet), whether there was a sale, whether the parameters returned by JavaScript are correct for a given browser type and the information about the browser has not been counterfeited.

In addition, information about the website user's behaviour is also taken into account: the number of visits to the website, the area in which the mouse moved, the number of page scrolls, the number of unique mouse points indicated, the number of pressed keys, the number of clicks with the left, middle and right mouse buttons, the number of pages remembered in the history, the number of text fields on the page, the number of pasted words from the clipboard, the number of text fields clicked, the number of passes between text fields on the page with the Tab key, the number of all controls in the form necessary to fill, the number of changed values in the controls, number of texts changed during editing, number of fields filled with no key pressed, number of fields filled very quickly (time is measured in ms).

In the experiments, three different models of neural networks described in Section 3 were tested:

– Model 1 – model of a multilayer perceptron network, where one-hot encoded categorical data and other variables in the form of real numbers are provided as inputs. The network has two hidden layers, the ReLU activation function, and the Softmax function at the output. The network diagram is shown in Figure 2. The investigated network structure has the following parameters: $n_1 = 10$ and $n_2 = 5$. Due to one-hot encoding, the number of network inputs is $m = 199$.

– Model 2 – a multi-layer network model, where

categorical data is fed to the embedding layers, and data that are real numbers are fed to a separate hidden layer. Both signals are transferred to two successive hidden layers, similar to Model 1. The network diagram is shown in Figure 3. The parameters of the network under study are as follows: the number of neurons in the subsequent layers of the network: $n_1 = 5$, $n_2 = 10$, $n_3 = 5$, the number of neurons for each category in embedding layers $n_e = 2$, the number of network inputs $m = 6$.

– Model 3 – similar to Model 2, the weights of which are trained in a different way. The weights of embedding layers are trained first. For this purpose, a special neural network with one hidden layer is prepared, shown in Figure 4. After training, the weights are transferred to the Model 2 and the network is trained with these weights values unchanged. The network parameters are the same as in Model 2.

Each network has three outputs corresponding to three data labels: 'ok', 'fake' and 'undecided' for the data from monitoring clicks, and 'ok', 'fake' and 'copy' for the data from monitoring leads. A series of ten training sessions was conducted for each of the models. The models were trained with data derived from monitoring clicks and leads, previously adjusting the appropriate number of network inputs for each case. Accuracy, precision and recall were used to evaluate the effectiveness of the tests [9]. Accuracy is the degree of closeness of measurements of a quantity to that quantity's true value

$$accuracy = \frac{tp+tn}{tp+tn+fp+fn} \tag{11}$$

Precision is the ratio of the number of correctly classified cases to the total number of irrelevant and relevant cases classified

$$precision = \frac{tp}{tp+fp} \tag{12}$$

and recall is the ratio between the number of data that are correctly classified to the total number of positive data

$$recall = \frac{tp}{tp+fn} \tag{13}$$

where tp – true positive, tn – false positive, fp – false positive, fn – false negative and they can be derived from the confusion matrix [9]. The parameter which combines the above two parameters is $F1$ score that it is the harmonic mean of precision and recall

$$F1 = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{14}$$

Tables 1 and 2 show, respectively: average results from ten training trials of three models for click monitoring data, the best results obtained during this training. Tables 3 and 4 present the average results obtained from ten attempts to train systems on the data obtained from lead monitoring for three different models and the best results obtained.

When scrutinizing the best results obtained, it can be noticed that Models 2 and 3 are comparable with each other in terms of learning efficiency. Model 1 clearly gives worse results. It is worth noting; however, that Model 3 showed a much better reproducibility of the results during learning. For practical applications, where a system of this type would be trained regularly with new data, Model 3 would require much less repetition of learning in the case of unsatisfactory results, which would translate into the speed of learning and implementation of the new model into production.

Table 1. Average results obtained for ten attempts to train neural networks for the problem of monitoring clicks.

Model		Acc	Prec	Rec	F1
1	train	0.898	0.89	0.90	0.87
	test	0.897	0.89	0.90	0.87
2	train	0.77	0.66	0.77	0.69
	test	0.78	0.66	0.77	0.69
3	train	0.95	0.93	0.95	0.94
	test	0.95	0.93	0.95	0.94

Table 2. The best results obtained during ten attempts to train neural networks for the problem of monitoring clicks.

Model		Acc	Prec	Recall	F1
1	train	0.975	0.974	0.975	0.974
	test	0.977	0.977	0.977	0.977
2	train	0.987	0.987	0.987	0.987
	test	0.986	0.987	0.986	0.986
3	train	0.987	0.987	0.987	0.987
	test	0.986	0.986	0.986	0.986

Table 3. Average results obtained for ten attempts to train neural networks for the problem of lead monitoring.

Model		Acc	Prec	Recall	F1
1	train	0.879	0.859	0.879	0.860
	test	0.879	0.860	0.879	0.860
2	train	0.822	0.725	0.822	0.760
	test	0.822	0.725	0.822	0.760
3	train	0.956	0.937	0.956	0.943
	test	0.956	0.937	0.956	0.943

Table 4. The best results obtained during ten attempts to train neural networks for the problem of lead monitoring.

Model		Acc	Prec	Recall	F1
1	train	0.974	0.974	0.974	0.974
	test	0.975	0.9754	0.975	0.975
2	train	0.995	0.995	0.995	0.995
	test	0.995	0.995	0.995	0.995
3	train	0.995	0.995	0.995	0.995
	test	0.995	0.995	0.995	0.995

5 Conclusion

In the paper, six neural network models prepared for the classification of network traffic related to Internet advertising were developed, presented and tested. The models were trained with real data obtained from 11 large websites – large Polish e-stores and financial institutions. The presented research may become the basis for conducting intelligent protection against artificially generated internet traffic, in particular against internet bots, virtual machines generating artificial clicks, click farms, fraudsters (fraudsters generating artificial clicks) and abuses related to repeated filling of applications by bots on websites (fraud leads). They can also contribute to the proper management of advertising expenses, be the basis for making complaints, blocking inappropriate visits to the website or queuing tasks related to handling leads. Worse leads, classified as artificially generated (bot), could be moved to the very end of the queue. Then, valuable leads could be handled with priority.

References

[1] AsSadhan, B., Moura, J.M., Lapsley, D., Jones, C., Strayer, W.T.: Detecting botnets using command and control traffic. In: 2009 Eighth IEEE International Symposium on Network Computing and Applications, pp. 156–162. IEEE (2009)

[2] Bengio, Y.: Learning deep architectures for AI. Now Publishers Inc (2009)

[3] Chen, C.M., Lin, H.C.: Detecting botnet by anomalous traffic. journal of information security and applications **21**, 42–51 (2015)

[4] eMarketer: Digital ad fraud 2019. https://www.emarketer.com/content/digital-ad-fraud-2019

[5] Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics, pp. 315–323. JMLR Workshop and Conference Proceedings (2011)

[6] Lagopoulos, A., Tsoumakas, G., Papadopoulos, G.: Web robot detection in academic publishing. arXiv preprint arXiv:1711.05098 (2017)

[7] Neal, A., Kouwenhoven, S., Sa, O.: Quantifying online advertising fraud: Ad-click bots vs humans. In: Tech. Rep. Oxford Bio Chronometrics (2015)

[8] Networks, D.: 2018 bad bot report. https://resources.distilnetworks.com/whitepapers/2018-bad-bot-report

[9] Rajaraman, A., Ullman, J.D.: Mining of massive datasets. Cambridge University Press (2011)

[10] Research, J.: Ad fraud - how ai will rescue your budget. https://news.unilead.net/wp-content/uploads/2017/09/Ad-Fraud-How-AI-will-rescueyour-Budget-whitepaper.pdf

[11] Seyyar, M.B., Çatak, F.Ö., Gül, E.: Detection of attack-targeted scans from the apache http server access logs. Applied computing and informatics **14**(1), 28–36 (2018)

[12] Silva, S.S., Silva, R.M., Pinto, R.C., Salles, R.M.: Botnets: A survey. Computer Networks **57**(2), 378–403 (2013)

[13] Soniya, B., Wilscy, M.: Detection of randomized bot command and control traffic on an end-point host. Alexandria Engineering Journal **55**(3), 2771–2781 (2016)

[14] Zhu, X., Tao, H., Wu, Z., Cao, J., Kalish, K., Kayne, J.: Fraud prevention in online digital advertising. Springer (2017)

Marcin Gabryel earned his Ph.D degree in computer science at Czestochowa University of Technology, Poland, in 2007. He is an assistant professor in the Department of Computer Engineering at Częstochowa University of Technology. His research focuses on developing new methods in computational intelligence and data mining. He has published over 50 research papers. His present research interests include deep learning architectures and their applications in databases and security.

Magdalena Scherer received her M.Sc. degree in computer science from the Czestochowa University of Technology, Poland, in 2008 and her Ph.D. in management in 2016 from the same university. Currently, she is an assistant professor at Czestochowa University of Technology. Her present research interests include machine learning for prediction and classification.

Łukasz Sułkowski holds a professor degree in economics and a doctoral degree in humanities and specializes in issues related to university management. He is the head of the Department of Higher Education Institution Management at the Jagiellonian University as well as a professor at Clark University and at the Academy of Social Sciences. His research interests include organization and management, epistemology and methodology of social sciences and humanities, organizational culture and intercultural management as well as public management and family business management. He is the author of about 400 publications and 10 books closely related to the subject of his interests.

He is a member of the following international organizations and associations: American Academy of Management (AAofM - USA), International Family Entreprises Research Assotiation (IFERA), Reseau Pays du Groupe de Vysegrad (PGV) and the European Academy of Management (EURAM).

Robertas Damaševičius received his Ph.D. in Informatics Engineering from Kaunas University of Technology (Lithuania) in 2005. Currently, he is a Professor at Department of Applied Informatics, Vytautas Magnus University (Lithuania) and Adjunct Professor at Institute of Mathematics, Silesian University of Technology (Poland). His research interests include sustainable software engineering, human–computer interfaces, assisted living, data mining and machine learning. He is the author of over 300 papers as well as a monograph published by Springer. He is also the Editor-in-Chief of the Information Technology and Control journal and has been the Guest Editor of several invited issues of international journals (Biomed Research International, Computational Intelligence and Neuroscience, Journal of Healthcare Engineering, IEEE Access, and Electronics).