UDK 004.852

Ruslan O. Shaporin[1], Candidate of Technical Sciences, the Head of the Computer Intellectual Systems and Networks Department, E-mail: shaporin@opu.ua, Scopus ID: 57204221232, ORCID: 0000-0003-4407-2367
Vladimir O. Shaporin[1], Candidate of Technical Sciences, Associate Professor of the Computer Intellectual Systems and Networks Department, E-mail: shaporin.v.o@opu.ua, Scopus ID: 57189248339, ORCID: 0000-0001-6494-7648
Oleg M. Mikhailov[1], Student of the Computer Intellectual Systems and Networks Department, E-mail: mikhailov.oleg.m@gmail.com, ORCID: 0000-0002-4088-0570
Alexander V. Lysenko[1], Student of the Computer Intellectual Systems and Networks Department, E-mail: lysenko.sasha.v@gmail.com, ORCID: 0000-0002-5025-5891
[1]Odessa National Polytechnic University, Odessa, Shevchenko av., 1, Odessa, Ukraine, 65044

ARTIFICIAL INTELLIGENCE SYSTEM FOR IDENTIFYING ROBOT BEHAVIOR ON A WEB RESOURCE

Annotation. *The architectural implementation of a machine learning system for identifying a robot on a web resource by behavioral factors is considered. The article discusses how to build software architecture for a machine learning system whose task is to determine the behavior of anonymous users. Behavioral factors for identification are a set of factors describing various components, each of which may be characteristic of the behavior of the robot. Weka software provides a mechanism for training on designed data models describing human and robot behavior. The learning algorithm – the "method of nearest neighbours", provides the construction of images based on the largest number of combinations of factors that describe one of the models. Data models for training are stored in a file on the hard disk in the form of matrices of feature descriptions of each of the types of behaviors. The article discusses software and algorithmic solutions that will help solve the problems of combating fraudulent clicks, spam and distributed multi-session attacks on the server, as well as reducing the level of confidence in the website for search engines. The article discusses software and algorithmic solutions that will help solve the problems of fighting click fraud, spam and DDOS attacks, as well as reducing the level of trust of a web site for search engines. Because a large number of illiquid and malicious traffic reduces search positions and reduces the TIC (thematic citation index) and PR (page rank) of the site, which reduces the profitability of the web resource. A large number of illiquid and malicious traffic reduces search positions and reduces the thematic citation index and search ranking of site pages, which leads to a decrease in the profitability of a web resource. The results of this article are the proposed behavior analysis system, a description of the technical implementation shell and a system training model. The statistics for comparing malicious traffic after connecting the system to a web site are also given. The implementation language was selected as Java. Using this system possibly allows cross-platform integration of the system, both on Linux and Windows. Data collection from the site, to determine the role of the user, is carried out using JavaScript modules located on the web resource. All data collection algorithms and user information storage periods are implemented within the framework of the European Data Protection Regulation. The system also provides complete anonymity to the user. Identification is carried out exclusively using fingerprint tags.*

Keywords: *click fraud; robots; antifraud; machine learning; perception; Java; Weka*

Introduction

The ability to identify a robot on a website allows protecting the web-resource from unreliable conversions and visits, which can also bring financial costs to resource owners or lead to loss of copyright data. Any web-resource has the main goals and characteristics: profit, live visits and the likelihood that the user will return to this resource in the future. In general, this forms the profitability and popularity of the web-resource [1]. However, these indicators are strongly distorted by a large number of malicious software - in particular, Internet robots.

Website protection at the network level is not possible if the robot uses dynamically changing IP addresses. Many robots work by simulating user behaviour, but this behaviour is still a rigidly structured logic that can be described by a set of repeating factors. There are various types of malicious robots: click bots, file down robots, imitation robots, scraper robots, spam robots.

One of the most profitable areas of Internet business is web advertising. Sites with a large number of visitors can place ad units on their pages, for example, Google Ads. A conversion in Internet marketing is the ratio of the number of website visitors who have performed any targeted actions on it (hidden or direct instructions from advertisers, sellers, content creators – buying, registering, subscribing, visiting a specific page of a website, clicking on an advertising link), to the total number of site visitors, expressed as a percentage [2]. These ad units, depending on their type (cost per mille (CPM), cost per order (CPO), cost per click (CPC),

© Shaporin, R. O., Shaporin, B. O., Mikhailov, O. M., Lysenko, A. V., 2019
This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/deed.uk)

DOI: 10.15276/hait.04.2019.5
ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)

cost per action (CPA)), charge the advertiser, the owner of the advertisement and the owner of the site where the advertisement is placed. The cost of each ad unit depends on the performance of the current website, namely: unique visitors and the price of a unique visitor [3]. For example, clicker robots bring high costs to online advertising, which operate on the CPC model, where payments to the advertiser are made for each click on the advertising banner. However, clicker robots make empty clicks on ad units, thereby reducing the conversion of these ad units, since this click will not lead to any action on the use of the advertised services or goods. Such bursts of blank taps distort further resource statistics.

Literature review

For timely detection, identification and localization of the impact of malicious robots, it is necessary to provide the system with a knowledge base about malicious robots and means of combating them. Then you need to teach the system to work with this knowledge base. The main element of machine learning is the reliable selection of data that will be used to train the system. Any inaccuracy or miscalculation during training can lead to complete inaccuracy of the result of the system. Repeated retraining entails a reset of results and repeated costs for updating the knowledge base of the system.

The behaviour of a machine learning system is described on the basis of constructing a perceptron logic diagram of the structure of the interaction of value judgments [4]. Each assessment should have a number of input data received from the user and methods for processing them that would fit the description of one of the trained behaviour models [5-6].

However, robotic behaviour algorithms are developing every day, so the system must have three main factors [7]:

– training on examples that will be prepared in advance to build a basic understanding of the study area;

– training with reinforcement, which is based on the positive and negative outcomes of the analysis;

– self-education, i.e. the ability to simultaneously and without reboots expand the knowledge base based on reliable negative and positive results of previous analyzes.

Any system for working with user data should be formulated on the basis of generally accepted standards for the storage of user data and personalization methods of the user on the Internet. Many technical tools that are used to write robots have their own nuances and vulnerabilities, which can be used to determine with high probability that the specified user is a robot [8]. Typically, such tools are implemented in order to expand the functionality of a programming language or library, but such vulnerabilities can be used as identification flags, as an ordinary user in the browser does not have such functionality.

Statement of the problem and purpose of the study

The aim of the study is to develop a microservice system for analyzing user behaviour on the site in order to determine the role of this visitor: “robot” or “human”. It is necessary to give a general description of the robot’s behaviour models and methods for identifying them, as well as a description of the structure of the machine learning system based on Java and forecasting tools – Weka, as well as describe behavioural assessment factors and examples of robot behaviour.

The analysis system should be able to identify robots of any type on a web resource with the aim of the possibility of their subsequent identification and blocking access to them on a web resource.

The system must comply with the general requirements:

– lack of influence and changes to the website interface;

– do not create delays when loading the visual component of the site;

– ensure the anonymity of the transmitted data;

– do not store personal data of users;

– the maximum size of the response delay is up to 120 ms;

– cross-platform.

All collected data from the user's web browser while it is on the site must be transmitted via HTTP POST request.

The server implementation of the module must have two Application Program Interface (API) methods. One API method returns the result of the analysis of user data received from the client module, and the second method provides the ability to dynamically expand the knowledge system with loadable data models. For the Weka machine learning algorithm [9], the k-nearest-neighbour algorithm is used. This algorithm allows analysis based on a comparison of two boundary types of behaviour and to evaluate them by different intersections of behaviour factors.

Main part. Each company that provides the ability to place and sell advertising platforms has its own individual mechanism for determining malicious traffic or uses third-party services. However, the use of such services implies additional costs for checking the visitor, and this reduces the company's profit. Therefore, there is a need for an

independent analysis of each visit to a user on a website using an analysis system. Most robots for analyzing web sites are developed according to well-known schemes or a similar behaviour algorithm, therefore, to describe the model of artificial intelligence (AI) used in this task, it is necessary to highlight the main value judgments by which AI can later make a quick verdict about the type of visitor.

The reasons for starting a behaviour analysis may be a sudden increase in visitors to the site followed by clicking on the ad units, but the time spent by the user on the site remains minimal. Such bursts should be filtered by AI systems for traffic analysis. Also, these users usually have the ability to save cookies disabled. This eliminates the possibility of assigning a label to the user for identification in the future. However, as a rule, it is possible to track the IP of each incoming user and if the AI system once gives a "robot" rating for any IP address, then requests from this address will be ignored in building statistics or charging funds for clicking on an ad unit.

You can place invisible blocks (Fig. 1) outside the visible part of the web resource, but with the name of the block containing the words "ad", "ads", "advertiser", etc. Such blocks, as a rule, contain advertising content, and clicker robots are more likely to react to them and produce clicks on blocks with which a regular user cannot interact.

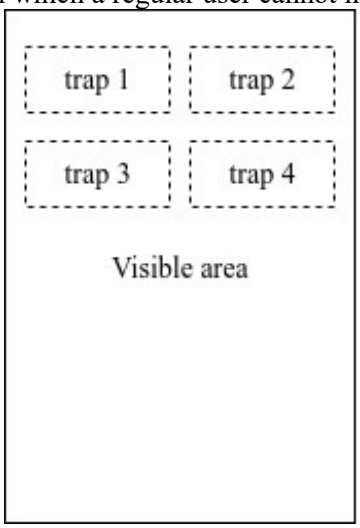


Fig. 1. Visible document area and trap blocks

JavaScript is used to collect data from the client side, thanks to which it is possible to collect data about user actions on the site, for example, to build a map of the cursor moving around the site, time spent on the site, click locations, and frequency of visiting the resource. Since robots are most likely to use many different IP addresses, it is also possible to exclude IP addresses by dialling from a range by grouping them according to similar behaviour. As a rule, the user has scrolling pages, viewing news; there is no direct path for moving the cursor. In turn, robots usually learn content by searching for blocks with specific keywords in their name. The movement of the cursor on the document is tracked on each of the conditional sectors of the website (Fig. 1).

A set of this data collected from the client side is sent along with the request to display ads on the server side, where there are two stages of data processing. The first stage is analysis using the developed system, which provides the entire set of input data and which gives an instant assessment of the similarity with the behaviour patterns of various robots.

At the second stage, advertising content is generated taking into account whether the robot or a live user clicked. Thus, the ratio of clicks of live users to actions after a click is stabilized and each click will be targeted.

For the analysis of user clicks, 5 rules are selected, each of which can have a rating from 0 to 100. The ratings are set by the Weka analysis system based on previously trained data. Each of these rules forms intersections that form the resulting classifiers. Each such classifier is used to determine membership in a particular conclusion.

All input data for training are designed in advance with a different number of intersections according to the following rules (Fig. 2):

A. Visible area.

A.1. Hit of the click in the visible area;

A.2. Click transitions between sectors (1, 2, 3, 4);

A.3. Hit of the click by sectors (1, 2, 3, 4);

A.4. Clicking occurs with the scroll event.

B. The intersection of clicks after building straight lines between the stopping points of the cursor, based on the beginning and end of the cursor moving over the visible area of the site.

C. Pressing time and frequency taking into account events:

a) intra-block context events: MouseDown, PointerDown;

b) the event "Video";

c) event "Sound";

e) event "keyboard".

D. User Activity by Event:

D.1. Clicking takes place in the depth of the website transitions;

D.2. Clicking occurs from a point in the world of the corresponding geo-location of the website (timezone).

E. Availability of settings and cookies:

ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)

E.1. There is a fingerprint or any arbitrary marker placed in the cookie or local storage of the user's browser;

E.2. The Browser Session setting is enabled (it does not work in all browsers) [6].

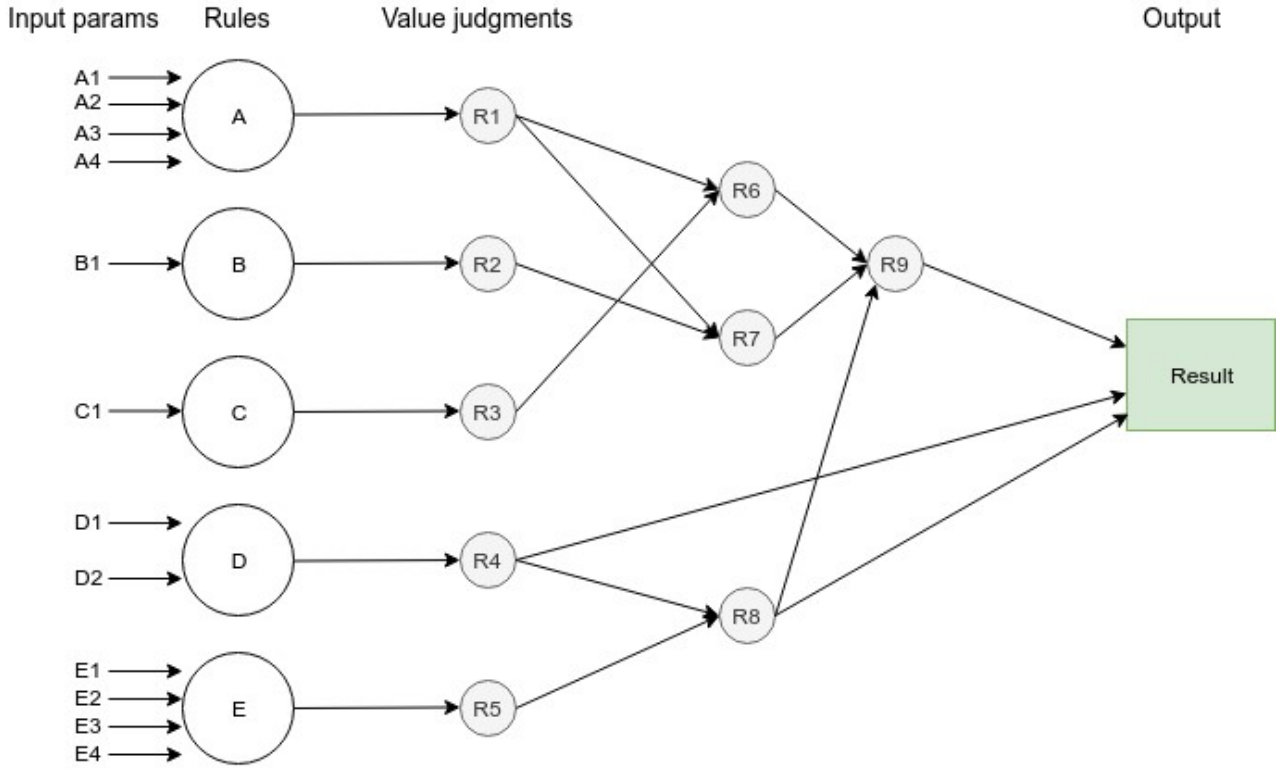


Fig. 2. The structure of the system for determining user behaviour on a web resource

According to the selected rules, it is possible to construct a generalized form of a multilevel perceptron (Fig. 2). The perceptron includes S (sensory), A (associative), R (reactive) elements. The S-element in the diagram is the rules, A is the value judgment, and R is the conclusion constructed by the analysis system.

The mathematical model consists of the sums of all the probabilities of obtaining a positive assessment in one of the blocks R1... R9. If an A-element has n inputs, then n weights and a threshold value Θ must be specified in it. The perceptron generates a logical 1 at the output if the linear form from the inputs with coefficients exceeds Θ and -1 if the condition is not met.

In such systems, it is important to lay the possibility of dynamic expansion and the flexibility of introducing new rules, because the technologies used to create robots are constantly evolving, and the system must be dynamically expandable. Therefore, the formation of the analysis result can be described by the formula [10-11]:

$$f(x) = \frac{\left(\sum_{i_n} W_{ij_n} \ldots \frac{\sum_{i_2} W_{2j_2} \frac{F\left(\sum_{i_1} w_{1j_1} - \Theta_{j_1 1}\right)}{layer\,1} - \Theta_{j_2 2}}{layer\,2} \ldots - \right)}{l\text{layerN}}$$

where: i – the input number;

j – the number of neuron in the layer;

– the weight coefficient of the i-th input of the neuron of the layer j;

– the weight matrix of all neurons of layer j.

Each layer calculates a non-linear transformation of the input parameters from a linear combination of signals from the previous layer. Such a combination forms an arbitrary multidimensional function at the output with an appropriate choice of the number of layers, the range of signals and parameters of neurons. Thus, each of the input parameters has its own significance coefficient in the formation of the subsequent layer. These coefficients are necessary to correlate the effect of the layer on

the next point, for which this conclusion becomes an input parameter (Fig. 3).


```
{"cursor": {
    "click_point": [
      {
        "block_type": "div",
        "class": "button_menu",
        "content": "Menu",
        "style": "format_tmp"
      },{
        "block_type": "div",
        "class": "adv_block",
        "content": "Extra sales!",
        "style": "ad_formatter"
      }
    ],"stopped_coord": [
      {"x": 120,"y": 81},{"x": 13,"y": 245},
      {"x": 94,"y": 124}
    ],
    "speed": 1205,
    "use_scroll": true
  },"user_hash": "asdfasdfa1u2isdkjfasdf",
  "browser": {    "local_storage": true,
    "browser_logging": true,
    "nav_webdriver": false,
    "nav_plugins": [
      "flash"
    ]}}
```


Fig. 3. Example user data "robot" in JSON format

The classification task in machine learning is the task of assigning an object to one of the predefined classes based on its formalized features. Each of the objects in this problem is represented as a vector in an N-dimensional space [12], each dimension in which is a description of one of the features of the object. In this case, it is necessary to classify the user data received from the client module: the speed of moving his cursor, the history of visits from his IP address, the presence of various browser settings, the area of pressing and focus of his cursor, as well as the depth of passage through sections of the site and the duration of the session on each. The training is based on the method of k-nearest neighbours – a metric algorithm for automatic classification of objects or regression [13]. In the case of using the method for classification, the object is assigned to the class that is the most common among k neighbours of this element, the classes of which are already known. In the case of using the method for regression, the object is assigned an average value over the k nearest objects to it, the values of which are already known. The algorithm can be applied to samples with a large number of attributes (multidimensional). To do this, before applying, you need to determine the distance function between each attribute.

Different attributes can have a different range of represented values in the sample (for example, attribute A is presented in the range from 0.1 to 0.5, and attribute B is presented in the range from 1000 to 5000) [18]. Distance values can be highly dependent on attributes with large ranges. Therefore, data is usually subject to normalization. In cluster analysis, minimax normalization is used. Minimax normalization is carried out as follows [16]:

$$x' = (x - \min[X])/(\max[X] - \min[X]),$$

where: x – distance between attributes.

In this case, all values will lie in the range from 0 to 1; discrete binary values are defined as 0 and 1.

Some significant attributes may be more important than others, therefore, for each attribute a certain weight can be set in accordance [14] [15]. Thus, each attribute k will be associated with a weight Zk, so that the attribute value will fall into the range [0, max (k)]. For example, if an attribute is assigned a weight of 2.7, then its normalized-weighted value will lie in the range [0, 2.7] [17].

Since the system has a microservice structure, integration into any existing system is possible. For the user, the design of the website will not have any visible changes. Also, initialization of the client module does not create visible network delays.

Currently, there are many solutions on the market that contain some of the described features, for example, using Google reCAPTCHA is a good algorithm for determining a robot, but due to the need for user interaction with the interface, this can scare away real visitors [19]. There is also the possibility of using Akismet, a system that determines the behaviour of a user by his activity on the site, for example, the number of posting comments or opening links and transitions [20]. The system is invisible to the user, however, it will not protect against parser robots that steal content from a web page, as well as robots that increase the number of downloads or visits.

The developed system was created taking into account all these shortcomings; it is also invisible to the visitor and contains a calculation of the number of user inputs and determination of its uniqueness. The connection between the server and client part occurs when the user goes to the web page where the Javascript module is located, which collects data about the user, tracking his activity on the website and checking the browser data. An AJAX request of the "POST" type is sent from the client to the API method "/check_user". The request contains information about the presence of global variables in the browser, data on the cursor movement, as well as

ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)

a unique fingerprint tag, then it is divided on the server and the structure of perceptron element objects is formed, which is transmitted for identification to the analysis system. If the system determines that this request contained data describing the robot, then a response is sent to the client with the HTTP code “204” and the JavaScript module closes the connection from the servers for this session. The IP address from which the visit was made to the site will be indicated in the system as mistrust. If the user has not been identified as a "robot", the response with the HTTP code "200" will be returned and in this case the site content will be displayed to the user. For example, in the case of the presence of ad units on the site, requests for selection and display of advertising banners will occur.

The client module consists of four main parts:

1) Subscriber Identity Module;

2) The analysis module of the browser and its environment;

3) The module for tracking the speed and behaviour of the cursor;

4) A module for analyzing blocks that a user clicks on.

When the site is first opened, the client module checks the following places in the search for fingerprint tags: cookie, local storage and evercookie (web history, local shared, session, local, global storages). Using evercookie allows you to automatically delete it from the others when you delete a cookie containing an identifier tag in one of these places. The algorithm works even if the user changes the browser (via Local Shared Objects with Flash). This type of identification is not prohibited by the GDPR agreement, since it does not contain any user data in its key.

The browser analysis module checks the following parameters from the browser and environment:

– the presence of global variables;

– User-agent browser;

– access to the LocalStorage browser;

The client module checks for the global variable “navigator webDriver”, this marker may indicate that the current session was initialized by a robot developed using Selenium. The presence of the global variable browser.logging_ evaluate in the browser indicates that this session is initialized using the PhantomJS library.

The headers of a request sent via PantomJS have the following differences:

1) the Host header is the last;

2) the value of the Connection header is sent in upper case for the first letters;

3) the Accept-Encoding header contains only gzip;

4) the User-Agent header contains “PhantomJS”.

The next factor in determining PhantomJS is the lack of access to installing plugins in the browser. Each browser has a set of plugins, a list of which can be obtained by request of the API – “navigator.plugins”. The name of the environment variables contains an array of extensions installed in the browser.

Usually it contains Flash, ActiveX, support for Java-applets or Default Browser Helper which indicates that this browser is installed by default in the client operating system. Such parameters can also be in emulators that are used by robots; however, in order to accelerate the opening of the session and increase productivity, these settings can be turned off by the robot.

Discussion of the results of the study

Based on the results of the study, a system for analyzing user behaviour using machine learning was developed. Using the Weka tool, the system will be able to identify the behaviour of each logged-in user based on previously analyzed traffic, thereby simultaneously performing two functions: determining the user and training their own knowledge bases with each new request. In addition, the system must be supplemented with a complex of block traps on the page. Thus, thanks to this integrated approach, it was possible to reduce the number of robots on the site by about 20 %. This was especially noticeable at the peak of daily traffic, since a large percentage of RPS (Request per Second) was from third-party robots that are not targeted users.

To test the performance of the system, the module was integrated into two large sites of online movie theatres. Such sites usually contain a constant percentage of daily visits, which makes it possible to exclude the possibility that the volume of traffic was less due to any third-party factors. Based on the statistics from Google Analytics, an analysis can be made. There is an average daily turnover of visitors on the site of 3 million people, as can be seen on the graph (Fig. 4).

The upper curve is the number of visits before the analysis system was turned on, black - with the system turned on. For performance measurements, two statistical samples were taken. The first sample for two days of the first week of December, the second two days (with a working analysis system) for the second week of December.

Since advertising banners usually contain targeted clicks and when using the system there is a 12 % increase in the cost of clicking.

To calculate the cost of pressing the formula is used:

$$cost = AE / CC * 100,$$

where: AE – amount of expenses,

CC – number of clicks.

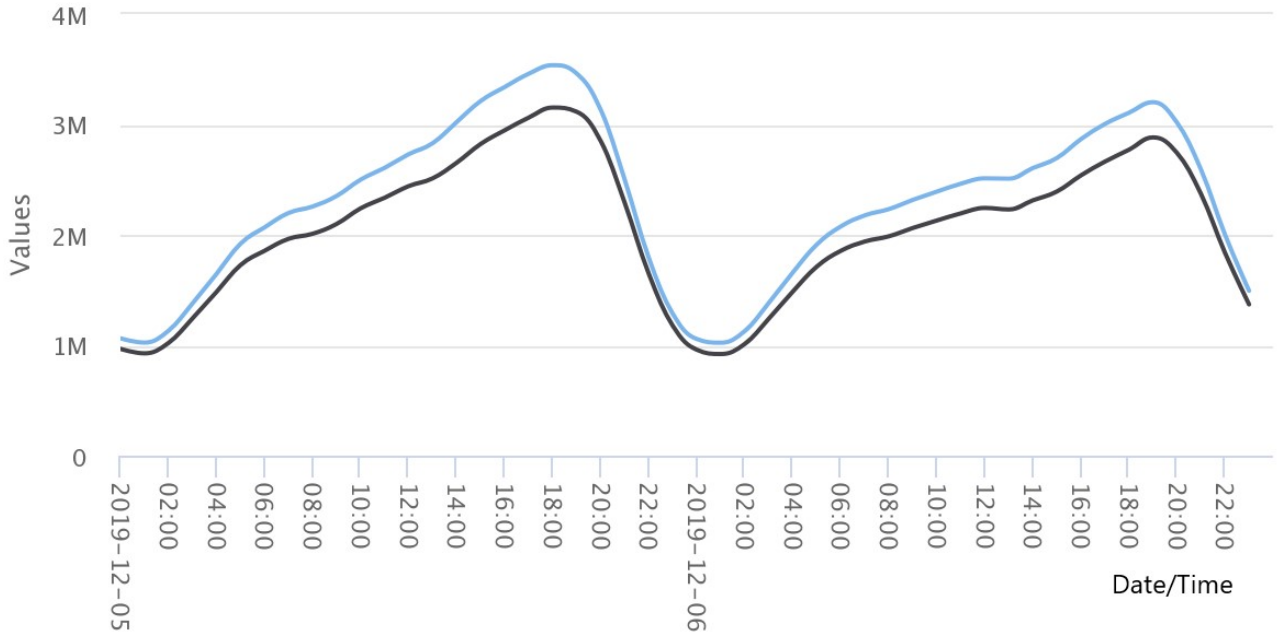


Fig. 4. RPS ratio before and after connecting the robot control system

Thus, the site's performance was increased and a significant percentage of false traffic was lost. This led to an increase in the cost of clicking on the ad unit, and, therefore, the owner of the web resource increased the percentage of monetization of the resource and reduced the load on the server side.

Conclusions

As a result of the study, a machine-learning-based system was developed to analyze user behaviour on a website in order to determine the source of traffic from malicious robots. The implementation of this system in the operation of the web site allowed reducing the number of visits to the site by robots, which caused losses to companies that place advertising content on the site. The developed system integrates into any environment, and is also invisible to the user.

References

1. Goncharov, N. O. & Gorchakov, D. S. (2015). Rassledovanie incidentov, svjazannyh s mobil'nymi bot-setjami i vredonosnym programmnym obespecheniem, [Investigation of Incidents Related to Mobile Botnets and Malware], St. Petersburg, Russian Federation, *Publ. FGAOU,* No. 4, pp. 28-34 (in Russian).

2. Verdoy, A. (2000). "Definition of Conversion Rate" [Electronic resource]. – Available at: URL: https://www.marketingterms.com/dictionary/conversion_rate/ – Active link: 15.11.2019.

3. Avinash, K. (2009). Veb-analitika, analiz informacii o posetiteljah veb-sajtov, [Web Analytics, Analysis of Information about Website Visitors], Trans. from Eng., Moscow, Russian Federation, *Publ. Williams*, pp. 242 (in Russian).

4. Kashi, R. S., Lopresti, D. & Wilfong, G. T. (2002). Ocenka jeffektivnosti algoritmov obrabotki tablic, [Evaluation of the Effectiveness of Table Processing Algorithms] *International journal of analysis and recognition of documents*. – Moscow, Russian Federation, *Publ. ICPR*. N. 3, pp. 140-153 (in Russian).

5. Sil'va, S., Horhe, A. & Torgo, L. (2008). Razrabotka skvoznogo metoda dlja izvlechenija informacii iz tablic, [Develop an end-to-end Method for Extracting Information from Tables], St. Petersburg, Russian Federation, *Publ. ITLab*, pp. 144-171 (in Russian).

6. Miteva, R. (2018). "4 Highlights to look for in a Fraud Detection Solution" [Electronic resource].

ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)

– Available at: URL: https://www.onespan.com/blog/fraud-detection – Active link: 15.11.19.

7. Hajkin, S. (2008). "Neural Networks: Complete Course, 2nd Edition", Trans. from Eng., Moscow, Russian Federation, *Publ. Williams*, pp. 1103 (in Russian).

8. Wilbur, C., I. Zhu. (2015). "Click Fraud Monitoring". New York: USA, *Publ. Marketing Science*, 25 p.

9. Witeen, H. (1999). "Weka: Practical Machine Learning Tools and Techniques with Java Implementations", [Electronic resource]. – Available at: URL: https://researchcommons.waikato.ac.nz/handle/10289/1040 – Active link: 15.11.19.

10. Shekyan, S. & Vinegar, B. (2015). Opredeljaem Phantom-nyh botov, [Define Phantom-them Bots] [Electronic resource] – Available at: URL: https://blog.shapesecurity.com/2015/01/22/detecting-phantomjs-based-visitors/ – Active link: 15.11.19 (in Russian).

11. Uossermen, F. (1992). "Neurocomputer Technology: Theory and Practice". Moscow, Russian Federation, *Publ. Mir*, pp. 184 (in Russian).

12. Shaporin, V. O., Tishin, P. M, Kopytchuk, N. B. & Shaporin, R. O. (2013). Nechetkie lingvisticheskie modeli obespechenija bezopasnosti komp'juternyh setej, [Fuzzy Linguistic Models of Computer Network Security], *Modern information and electronic technologies: 14-th international scientific-practical conference,* Odessa, Ukraine, pp. 155-156 (in Russian).

13. Laros, T. (2004). "Discovering Knowledge in Data: An Introduction to Mining". New Jersy: USA, *Publ. Spring,* 240 p.

14. Gafner, V. (2010). Informacionnaja bezopasnost': uchebnoe posobie, [Information Security: a Tutorial]. Rostov na Donu: Russian Federation, *Publ. Feniks*. 324 p. (in Russian).

15. Shaporin, V. O. Tishin, P. M, Kopytchuk, N. B. & Shaporin R. O. (2014). Razrabotka nechetkih lingvisticheskih modelej atak dlja analiza riskov v raspredelennyh informacionnyh sistemah, [Development of Fuzzy Linguistic Attack Models for Risk Analysis in Distributed Information Systems]. *Modern information and electronic technologies: 15-th international scientific-practical conference,* Odessa, Ukraine, pp. 131-132 (in Russian).

16. Nesterenko, S. A., Tishin, P. M. & Makoveckij, A. S. (2013). Model' ontologii apriornogo podhoda prognozirovanija problemnyh situacij v slozhnyh vychislitel'nyh sistemah, [Ontology Model of the Priori Approach for Predicting Problem Situations in Complex Computing Systems], *Electrotechnic and computer systems.* Kiev: Ukraine, *Publ. Tehnika,* No. 0, pp. 111-119 (in Russian).

17. Kopytchuk, N. B. Tishin, P. M. & Cjurupa, M. V. (2014). Procedura sozdanija nechetkih modelej analiza riskov v slozhnyh vychislitel'nyh sistemah, [The Procedure for Creating Fuzzy risk Analysis Models in Complex Computing Systems]. *Electrotechnic and computer systems,* Kiev: Ukraine, *Publ. Tehnika,* No. 13, pp. 215-222 (in Russian).

18. Ruban, O. (2019). "Volterra Neural Network Construction in the Nonlinear Dynamic Systems Modeling Problem", *Herald of Advanced Information Technology,* Odessa, Ukraine, *Publ. Science and Technical*, Vol. 2, No. 1. pp. 24-28 [Electronic resource]. – Available at: URL: http://dspace.opu.ua/jspui/handle/123456789/8435 – Active link: 15.11.2019.

19. Sivacorn, S., Polakis, D. & Keromitis, D. (2008). "I am not a person: hacking Google reCAPTCHA". New York: USA, *Publ. Columbia University*, 13 p.

20. Thomason, A. (2009). "Blog Spam: Akismet Review". San Francisco: USA, *Publ. Six Apart*, 5 p.

21. Dyball, J. (2009). "Anti-fraud Voter Registration and Voting System using a data Card". New York: USA, *Publ. Abloy*, 14 p.

22. Ge, L. (2007). "Real-time Click Fraud Detecting and Blocking System", *Tennessee*, USA, *Publ. USPTO*. 19 p.

Received. 07.10.2019
Received after revision.20.11.2019
Accepted 02.12.2019

УДК 004.852

[1]**Шапорін, Руслан Олегович,** канд. техн. наук, доцент, завідувач кафедри комп'ютерних інтелектуальних систем і мереж, E-mail: shaporin@opu.ua, Scopus ID: 57204221232,
ORCID: https://orcid.org/0000-0003-4407-2367
[1]**Шапорін, Володимир Олегович,** канд. техн. наук, доцент кафедри комп'ютерних інтелектуальних систем і мереж, E-mail: shaporin.v.o@opu.ua, Scopus ID: 57189248339
ORCID: https://orcid.org/0000-0001-6494-7648
[1]**Михайлов, Олег Михайлович,** студент кафедри комп'ютерних інтелектуальних систем та мереж, E-mail: mikhailov.oleg.m@gmail.com, ORCID: https://orcid.org/0000-0002-4088-0570
[1]**Лисенко, Олександр Володимирович,** студент кафедри комп'ютерних інтелектуальних систем та мереж, E-mail: lysenko.sasha.v@gmail.com, ORCID: https://orcid.org/0000-0002-5025-5891
[1]Одеський національний політехнічний університет, Одеса, проспект Шевченка, 1, Одеса, Украина, 65044

СИСТЕМА ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ІДЕНТИФІКАЦІЇ ПОВЕДІНКИ РОБОТА НА WEB-РЕСУРСІ

Анотація. *Розглянуто архітектурна реалізація системи машинного навчання для ідентифікації робота на web-ресурсі по поведінкових факторів. У статті описано побудову програмної архітектури для системи машинного навчання, завданням якої є визначення поведінку анонімних користувачів. Поведінкові фактори для ідентифікації шкідливих роботів – це сукупність факторів, що описують різні складові, кожен з яких може бути характерним для поведінки робота. Програмне забезпечення Weka забезпечує механізм навчання по спроектованим моделям даних, що описують поводження людини і поведінки робота. Алгоритм навчання – «метод найближчих сусідів», забезпечує побудова образів на основі найбільшого кількість поєднань чинників, що описують одну з моделей. Моделі даних для навчання зберігаються в файлі на жорсткому диску у вигляді матриць ознакових описів кожного з типів поводжень. У статті розглядаються програмні та алгоритмічні рішення, які допоможуть вирішити проблеми боротьби з шахрайськими натисканнями на рекламні блоки, спамом і розподіленими багатосесійність атаками на сервер, а також зниження рівня довіри до web-сайту для пошукових систем. Велике у неліквідного і шкідливого трафіку знижує пошукові позиції і зменшує тематичний індекс цитування та пошуковий рейтинг сторінок сайту, що призводить до зниження прибутковості web-ресурсу. Результатами цієї статті є запропонована система аналізу поведінки, опис технічної оболонки реалізації і модель навчання системи. Також приведена статистика порівняння шкідливого трафіку після підключення системи на web-сайті. Мова реалізації – Java. Використання Java дозволяє кроссплатформенную інтеграцію системи, як на Linux, так і Windows. Збір даних з сайту для визначення ролі користувачів, здійснюється за допомогою JavaScript модулів, розміщених на web-ресурсі. Всі алгоритми збору даних і терміни зберігання інформації реалізовані в рамках загальноєвропейського регламенту щодо захисту даних. Також система забезпечує повну анонімність користувача. Ідентифікація здійснюється виключно за допомогою використання fingerprint-міток.*

Ключові слова: *клікфрод; роботи; антіфрод; машинне навчання; персептрон; Java; Weka*

УДК 004.852

[1]**Шапорин, Руслан Олегович,** канд. техн. наук, заведующий кафедры компьютерных интеллектуальных систем и сетей, E-mail: shaporin@opu.ua, Scopus ID: 57204221232,
ORCID: https://orcid.org/0000-0003-4407-2367
[1]**Шапорин, Владимир Олегович,** канд. техн. наук, доцент кафедры компьютерных интеллектуальных систем и сетей, E-mail: shaporin.v.o@opu.ua, Scopus ID: 57189248339,
ORCID: https://orcid.org/0000-0001-6494-7648
[1]**Михайлов, Олег Михайлович,** студент кафедры компьютерных интеллектуальных систем и сетей, E-mail: mikhailov.oleg.m@gmail.com, ORCID: https://orcid.org/0000-0002-4088-0570
[1]**Лысенко, Александр Владимирович,** студент кафедры компьютерных интеллектуальных систем и сетей, E-mail: lysenko.sasha.v@gmail.com, ORCID: https://orcid.org/0000-0002-5025-5891
[1]Одесский национальный политехнический университет, г. Одесса, проспект Шевченко, 1, Одесса, Украина, 65044

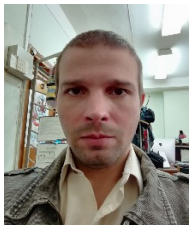
ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)

СИСТЕМА ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ДЛЯ ИДЕНТИФИКАЦИИ ПОВЕДЕНИЯ РОБОТА НА WEB-РЕСУРСЕ

Аннотация. *Рассмотрена архитектурная реализация системы машинного обучения для идентификации робота на web-ресурсе по поведенческим факторам. В статье описано построение программной архитектуры для системы машинного обучения, задачей которой является определение поведения анонимных пользователей. Поведенческие факторы для идентификации вредоносных роботов-это совокупность факторов, описывающих различные составляющие, каждый из которых может быть характерным для поведения робота. Программное обеспечение Weka обеспечивает механизм обучения по спроектированным моделям данных, описывающим поведение человека и поведение робота. Алгоритм обучения – «метод ближайших соседей», обеспечивает построение образов на основе наибольшего количества сочетаний факторов, описывающих одну из моделей. Модели данных для обучения хранятся в файле на жёстком диске в виде матриц признаковых описаний каждого из типов поведений. В статье рассматриваются программные и алгоритмические решения, которые помогут решить проблемы борьбы с мошенническими нажатиями на рекламные блоки, спамом и распределёнными многосессионными атаками на сервер, а также снижения уровня доверия к web-сайту для поисковых систем. Большое количество неликвидного и вредоносного трафика снижает поисковые позиции и уменьшает тематический индекс цитирования и поисковый рейтинг страниц сайта, что приводит к снижению прибыльности web-ресурса. Результатами данной статьи являются предложенная система анализа поведения, описание технической оболочки реализации и модель обучения системы. Также приведена статистика сравнения вредоносного трафика после подключения системы на web-сайте. Язык реализации–Java. Использование Java позволяет осуществить кроссплатформенную интеграцию системы, как на Linux, так и Windows. Сбор данных с сайта для определения роли пользователей, осуществляется при помощи JavaScript модулей, размещенных на web-ресурсе. Все алгоритмы сбора данных и сроки хранения пользовательской информации реализованы в рамках общеевропейского регламента по защите данных. Также система обеспечивает полную анонимность пользователя. Идентификация осуществляется исключительно при помощи использования fingerprint-меток.*

Ключевые слова: *кликфрод; роботы; антифрод; машинное обучение; персептрон; Java; Weka*

Шапорин Руслан Олегович
The main direction of scientific research – computer network design



Шапорин Владимир Олегович
The main direction of scientific research – computer network management automation and information security

Михайлов Олег Михайлович
The main direction of scientific research – intelligent software systems for computer networks

Лисенко Александр Владимирович
The main direction of scientific research – design of complex client-server systems of computer networks