Michał Bernardelli

Collegium of Economic Analysis
Warsaw School of Economics

Cheater detection in Real Time Bidding system – panel approach

Summary

The aim of this paper is to present an econometric model as a key to detect fraud traffic in the Real Time Bidding system. The proposed method was verified by computer simulations. It consists of two different models, one designed for user classification and the second to distinguish actual websites from those specially prepared by cheaters. Presented models depend on each other and together seems to be a quite fast and effective tool to separate online human traffic from artificial one generated by bots.

Keywords: Real Time Bidding, big data, cheaters detection, linear probability model

1. Introduction

Real Time Bidding (RTB) is a method of providing an online advertise in the real time. Due to the strict time limit (usually below 120 ms) this method is reserved only to automated systems using highly-specialised algorithms and powerful computers. Usually profit for the company is related to the action performed by the user after seeing the advertisement on a website. Showing the advertisement without any user interaction means loss for the company. Therefore mathematical models are created for identification of the users that have the highest probability of performing an action.

Showing one advertisement is relatively cheap and in most cases measured in micro dollars[1]. It must be emphasized, that the whole RTB market is worth billions of dollars. As an example consider the AppNexus – currently the biggest independent advertising company. It was created only in 2007, and by the 2012 it was executing transactions worth over 500 millions of dollars. In 2014 the worth of the transactions exceeded two

[1] There are however exclusive part of the internet market in which prices of advertisements are measured in the tens of dollars.

billion dollars. Right now the number of shown advertisements each day is estimated at around 30 billion. No wonder that the number of frauds and people trying to cheat the system is increasing rapidly. The costs related to the artificial traffic could reach tens of thousands of dollars in just a few minutes.

RTB is a model example of what kind of problems are encountered in big data. By definition big data is charactarised by:

- large quantity of data (at the moment at least millions of objects),
- wide variety of data,
- velocity, that is high speed of generation of data.

Therefore it is impossible to distinguish users manually. Dedicated, automatic filters of the traffic are the only solution of the cheaters detection problem. In this paper the method based on panel data is described. Computer simulations show that this approach has greater efficiency than algorithms that use a single time series.

This paper is composed of five sections. The more detailed description of Real Time Bidding system is in section 2. Section 3 presents the method of detecting the cheater's site and user based on the econometric panel model. In section 4 there are results from computer simulations exploring the presented model. This paper ends with the summary in section 5.

2. Real Time Bidding system

The creation of the Real Time Bidding system dates back to 2009, while the year 2011 is considered as the beginning of the RTB in Poland. Since that time can be seen dynamic development of RTB system. Basically in RTB we can distinguish four parts:

- Advertiser – entity that wants to show its advertisement on websites to promote a product / service or its online sales.
- Demand Side Platform (DSP) – entity that allows an Advertiser to buy in an automatic way the place for its advertisement on the website. The specialized models are created for buying the best places for the best users by the lowest price[2].
- Supply Side Platform or Sell Side Platform (SSP) – entity that sells the places to the DSPs, which offers the highest price. SSP is managing the transaction process of selling the space on the website and displaying advertising in it[3].

[2] Examples of DSP companies: Sociomantic labs, Criteo, HCore, Triggit, IgnitionOne, Xaxis.
[3] Examples of SSP companies: AppNexus, OpenX, PubMatic, Rubicon Project, DoubleClick.

- Publisher – entity that owns a website and gets a money for providing a space for online advertisements.

For the sake of clarity let's describe step-by-step the typical transaction within the RTB system. The whole process begins with the user visiting some website with the places for advertisements. This website must of course take a part in the RTB system. There are plenty of websites with advertisements, which are sold directly to an advertiser[4]. Assume that the website has few placements, which are appropriate for the advertisement and the owner of the website is offering those spaces for sale. SSP gets information about this particular entrance to the website and sends it to all DSPs. The set of information send by the SSP to the DSP is called bid request. It can contain many information about the website and a user, like location, language, url, user's ip, size of the space for the advertisement and its location on the website (on top, on the side or on the bottom), what kind of device, operating system and browser user is using, sex, age of the user and many others[5]. Based on those information in the bid request DSPs give their bids for each of the offered space separately (each space is connected with different bid request). Of course if a DSP is not interested in this particular user (or entrance of the user on the website to be more exact), it could send no bid, which is equivalent to the resignation. A bid is a value, that the DSP is willing to pay for the permission to display the advertisement in the specific place (and time) on the website. In real time potentially thousands of DSPs are estimating the value of the particular bid request and sending the answer (bid) to the SSP – and all of it typically within around 100 milliseconds. Bidding in RTB is an auction – the place on the website goes to the highest bidder among DSPs.

The interesting thing in RTB are the rules of the auction system. The mechanism that is used, is called the generalized second-price auction (GSP)[6]. The higher bidder wins the auction, but pays the price bid by the second-highest bidder. The analysis of GSP can be found in the economics and mathematical literature, see for example Edelman et al.[7], Caragiannis et al.[8] or Brendan et al.[9]

---

4 In fact in Poland most of websites, that offers advertisements, are not a part of the RTB system.

5 Type of these information usually depends on the SSP.

6 It is also used for example by Google's AdWords technology.

7 B. Edelman, M. Ostrovsky, M. Schwarz, *Internet Advertising and the Generalized Second-Price Auction: Selling Billions of Dollars Worth of Keywords*, "American Economic Review" 2007, vol. 97(1), pp. 242–259.

8 I. Caragiannis, Ch. Kaklamanis, P. Kannelopolous, M. Kyropolou, B. Lucier, R. Paes Leme, E. Tardos, *Bounding the inefficiency of outcomes in generalized second price auctions*, "Journal of Economic Theory" 2012, vol. 156, pp. 343–388.

9 L. Brendan, R. Paes Leme, E. Tardos, *On Revenue in the Generalized Second Price Auction*, Proceedings of the 21st International World Wide Web Conference (WWW '12), 2012, pp. 361–370.

Advertiser pays DSP for showing its advertisements and increase in sales, DSP is paying SSP for the execution of an auction, infrastructure maintenance and the place, where it can put the advertisement. Finally SSP pays the publisher for using the website for advertising purpose. Schema of the RTB transaction is presented in Figure 1. This process is repeated for each advertisement, billions of time every day, without noticeable delay to the user. Showing the advertisement on the website is typically called an impression and cost of it in most cases is expressed in micro dollars. Only a fraction of percent of impressions are ended with a user's action, like click, scroll or conversion. The rest of impressions (in practice most of them) could be considered as a financial loss. That is the reason for building the better econometric models and complex algorithms, which allow to win with the competition and increase the profit of the DSP.

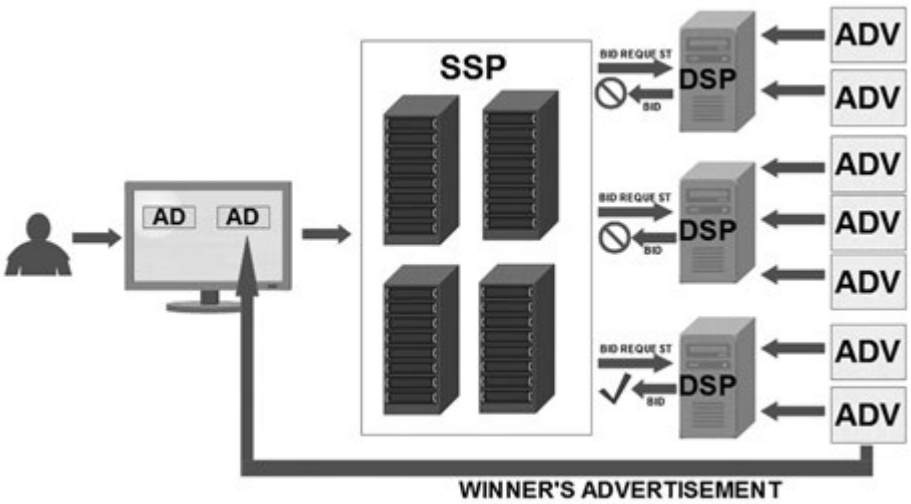

Figure 1. A schema of a Real Time Bidding transaction

Source: own calculations.

Due to the incredible amount of bid requests worldwide, it is easy to win practically any number of auctions and show impressions to millions of users. This involves extremely high costs and little chance of profit. For the DSP it is vital to choose the traffic carefully and participate only in auctions that have chance of being profitable (and of course winning those auctions with minimum financial effort). Unfortunately determining which bid request could be profitable and which definitely will not, is not an easy task. First of all the problem is in the strict time limit. Any sophisticated methods can't be used. The second issue is the big amount of data. The DSP must be able to handle hundreds of thousands of bid requests every second! This volume of data

have to be handle in the real time, but even off-line computations based on historical data is doubtful, because it is almost impossible to store the whole data. We are talking here about terabytes of data. Due to all aspects related to the listed technical difficulties and the growing market with high financial value, there is noticeable increase of activities aimed at swindling money by cheating the DSPs. Basic idea of frauds in the RTB system is to create the fake website and generate an artificial traffic, which directs to them. Software that allow to imitate the browsing the web by human-user, are usually referred to as bots. The time that will elapse before the DSP finds out, identify the fake site or user and block them, maybe long enough to loss tens of thousands of dollars. Some cheaters are generating extraordinary high traffic in short amount of time and some create bots for long-term work with a reasonable frequencies of bids. Both kinds of cheaters are hard to detect especially if we realize, that to the specific DSP goes only a part of the traffic generated by bot – the rest is spread over competing DSPs.

It is worth to emphasize, that there is a lack of scientific articles on the RTB algorithms. There are some research, see Chen et al.[10] or Shuai et al.[11], they are usually done from the SSPs point of view. Any efficient methods for DSPs, if they exists, are kept secret, because of the advantage they give over the competition. In this paper the proposition of an online cheaters detection algorithm based on an econometric model is presented. Its effectiveness is verified by the computer simulations that are described in section 4.

3. Method description

In this section the description of the method of detecting the fraud traffic is presented. By the fraud either the fake user or the cheaters site is assumed. Detecting both situations is equally important. To build econometric model the data are needed. In other fields of science the common problem is the insufficient availability of data. Here too much data is the problem. As was mentioned in the previous section gathering all data (bid requests) is practically impossible. The solution could be to use the sampled data – probably few per mill at most. Another approach, used in this paper, is to resign

[10] Y. Chen, P. Berkhin, B. Anderson, N. Devanur, *Real-time bidding algorithms for performance-based display ad allocation*, Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, New York 2011, pp. 1307–1315.

[11] Y. Shuai, W. Jun, Z. Xiaoxue, *Real-time bidding for online advertising: measurement and analysis*, Proceedings of the Seventh International Workshop on Data Mining for Online Advertising, New York 2013, article no. 3.

from collecting information about each entrance and to collect separately data about users and websites instead. It is still a significant amount of data: millions of websites and much more users. It must be emphasized, that a website is uniquely defined by IP address or its URL[12], while user by so called cookie. It can be assumed that cookie is uniquely assigned to the user, but the reverse implication does not hold. Every user can and usually is recognized by many different cookies[13]. Changing the computer, operating system, opening new browser window or simply deleting the cookie (for example for security reasons) will cause the generation of the new cookie. Nevertheless remembering information about users and websites separately is achievable considering the modern technology. The schemas of the tables with the information about the users and websites are given in tables 1 and 2.

Table 1. The schema of the table with the description of information about users contained in the table

Column name	Description
cookie	cookie assigned to the user
timestamp	time of the last bid request (with milliseconds)
count	number of bid request from the user
num_bad_time	number of bid request with too small time delay comparing to the previous bid request
num_good_time	number of bid request with long enough time delay comparing to the previous bid request
num_bad_site	number of bid request for the websites, which were considered as fraud trials
num_good_site	number of bid request for the websites, which were considered as good ones

Source: own calculations.

Table 2. The schema of the table with the description of information about websites contained in the table

Column name	Description
id	identificator of the website (e.g. based on IP or URL)
count	number of bid request for the website
num_bad_user	number of bid request from the users, which were considered as cheaters
num_good_user	number of bid request from the users, which were considered as good ones

Source: own calculations.

[12] URL – a Uniform Resource Locator, simplifying it is an address of a website, see T. Berners-Lee, R. Fielding, *Uniform Resource Identifier (URI): Generic Syntax*, www.ietf.org. Network Working Group 2005 (retrieved: 2014.09.05).

[13] Usually SSPs give their own cookie for the user.

Notice that columns *num_good_time* and *num_good_site* in the user table are not necessary, because they can be calculated as

$$num\_good\_time = count - num\_bad\_time$$

and

$$num\_good\_site = count - num\_bad\_site.$$

Analogously the column *num_good_user* in the websites table is redundant, because

$$num\_good\_user = count - num\_bad\_user.$$

The columns were introduced for the clarity of description and ease of understanding the algorithm. For generalization of the method about number of performed by the user actions, e.g. clicks, the tables need to be extended with extra columns. In the version of the method described in this paper those columns are not used.

The idea behind the procedure of cheater traffic detection can be described as follows. At a certain time interval, like a day (24 hours) the data are gathered online in the form of the user and website table (see Table 1 and 2). Afterwards in an offline mode the regression models for users and for websites are calculated. For the computations only users and websites with a sufficiently large number of views are taken, e.g. any user with at least 10 bid requests during the considered day and any website with the number of corresponding bid requests greater than 1000. The next day the calculated models are used to predict the probability of being fake user or website. At the same time the data are gathered: information about already seen users and websites are updated, while the data about the new users and new websites are added to the tables. In order to save storage space it is a good idea to remove users and websites, which were not seen for a longer period of time. The next day the whole procedure is repeated but with a new set of data and new models parameters.

Given the requirement of fast model evaluation and parameters computation, we deliberately choose one of the simplest class of models that is linear probability model. Let *num_site_begin_to_decide* be the parameter defining the number of bid requests corresponding to the website. When that number for the particular website is exceeded, then the website is taken into consideration in the model learning. Analogously, let *num_user_begin_to_decide* defines the number of bid requests from the user, exceeding which means, that this user is taken into account in the model calculations. Using the names form tables 1 and 2, the model for the website can be written in the following form

$$S_{i,t} = a_0 + a_1 \frac{num\_bad\_user_{i,t-1}}{count_{i,t-1} - num\_site\_begin\_to\_decide} + \varepsilon^S_{i,t}, \tag{1}$$

where:
$S_{i,t}$ – probability, that the *i*-th website at a moment *t* is a fake website,
$num\_bad\_user_{i,t-1}$ – number of bid request corresponding to the *i*-th website at a moment *t-1* from the users, which were considered as cheaters,
$count_{i,t-1}$ – number of bid request corresponding to the *i*-th website at a moment $t - 1$,
$\varepsilon^S_{i,t}$ – error term corresponding to the *i*-th website at a moment *t*. For the full definition of the linear probability model one thing is missing, that is to define the conditions what probabilities are considered as high enough to say that the website was prepared for the purpose of fraud. We define that limit as a minimum from the predicted by the model probabilities over all websites, which were consider as cheaters:

$$S_{limit} = \min_{i,t} \left\{ S_{i,t} : \begin{array}{c} num\_bad\_user_{i,t} > num\_good\_user_{i,t} \\ \wedge count_{i,t} > num\_site\_begin\_to\_decide \end{array} \right\}. \tag{2}$$

There could be used more sophisticated conditions involving for example occurrence of actions performed by the user. Advanced bots however are able to simulate also actions such as clicks.

Model for user classification is defined as follows

$$U_{j,t} = b_0 + b_1 \frac{num\_bad\_site_{j,t-1}}{count_{j,t-1} - num\_user\_begin\_to\_decide} +$$

$$+ b_2 \frac{num\_bad\_time_{j,t-1}}{count_{j,t-1} - num\_user\_begin\_to\_decide} + \varepsilon^U_{j,t}, \tag{3}$$

where:
$U_{j,t}$ – probability, that the *j*-th user at a moment *t* is a bot,
$num\_bad\_site_{j,t-1}$ – number of bid request corresponding to the *j*-th user at a moment $t - 1$ for the websites, which were considered as cheaters,
$num\_bad\_time_{j,t-1}$ – number of bid request corresponding to the *j*-th user at a moment $t - 1$ with too small time delay comparing to the previous bid request of that user,
$count_{j,t-1}$ – number of bid request corresponding to the *j*-th user at a moment $t - 1$,
$\varepsilon^U_{j,t}$ – error term corresponding to the *j*-th user at a moment *t*.
As in the case of websites so in the case of users, the limit probability, from which the user is interpreted as a cheater is defined as:

$$U_{limit} = \min_{j,t} \left\{ U_{j,t} : \begin{array}{c} num\_bad\_time_{j,t} > num\_good\_time_{j,t} \\ \wedge count_{j,t} > num\_user\_begin\_to\_decide \end{array} \right\}. \quad (4)$$

Calculating parameters in the presented two models could take a significant amount of time, due to the data volume. For the computations methods that are efficient not only because of the time, but also because of memory used, should be considered. Parallelization of calculations is advisable. Fortunately fast methods for calculating parameters of a linear regression model are known[14].

Both models are supposed to be complement to each other. Usage in real time can be presented in two steps. For every bid request (*i*-th website, *j*-th user, moment of time *t*):

1. Calculate value of the website model $S_{i,t}$ and compare it with S_{limit}. If $S_{i,t} \geq S_{limit}$ then the website is considered as a try of cheating. Update information about the website in the websites table (see Table 2).
2. Calculate value of the user model $U_{j,t}$ and compare it with U_{limit}. If $U_{j,t} \geq U_{limit}$ then the user is considered as a cheater. Update information about the user in the users table (see Table 1).

If in either of two steps the likelihood of fraud is high enough, then one resigns from participation in the auction ("no bid" as an answer is send).

4. Computer simulations

To verify the usefulness of the presented in the previous section method, computer simulations were performed. The simulation was divided into two parts. In the first, learning part the data were gathered and parameters of the user and website model were computed. In the second, testing part two models were evaluated for every bid request. In the computer simulation some assumptions were made. As the learning time interval an hour was chosen, while the test part last one minute. The value of the parameter *num_site_begin_to_decide*, as well as the value of the parameter *num_site_begin_to_decide* was set as equal to 2. Finally the time limit between two bid requests from the same user, that was considered as two short was set to 0.1 second. There are of course

[14] Compare M. Bernardelli, *Method of QR decomposition's fast updates for linear regression models*, „Roczniki" KAE, z. 27, Oficyna Wydawnicza SGH, Warszawa 2012, pp. 55–68; J.H. Friedman, *Fast Sparse Regression and Classification*, Technical Report, Standford University 2008.

situations when the human user view two websites in really short interval of time, but it is rather rare compared to the frequency of visiting the websites by the bot.

For the learning part of computer simulations ten thousand good (human) users and one thousand of good (real) websites were considered. For each user the frequency of visiting websites was chosen randomly as a parameter from the interval with at least one visit every 100 second and at most one website every two seconds. When it comes to bad users (cheaters), there were considered two groups of those kind of artificial users. In the first group there were two users, which visits only bad websites assigned to that user. The second group consists of two users that with some probability generates the traffic on created by them websites, but also (to keep up appearances) visits other, good websites. For every bad user randomly at most three bad, different websites were generated. In contrast to good users, the frequency of visiting websites for bad users was chosen randomly from one per second, to a hundred per second.

Computed models are presented in Figure 2 (users model) and Figure 3 (websites model). Every parameter in both models proved to be statistically significant. Calculated values of U_{limit} and S_{limit} were equal respectively 0.6182886 and 0.7827286.

```
                    Estimate Std. Error t value Pr (>|t|)
(Intercept)         -1.048e-02 9.817e-05-106.71 <2e-16 ***
num_bad_site_ratio  9.253e-01 5.650e-03 163.76 <2e-16 ***
num_bad_time_ratio  1.794e-01 3.898e-03 46.04 <2e-16 ***
--
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.006293 on 10001 degrees of freedom
Multiple R-squared: 0.9009,      Adjusted R-squared: 0.9009
F-statistic: 4.547e+04 on 2 and 10001 DF, p-value: < 2.2e-16
```

Figure 2. Coefficients of the users model with basic measures and statistical test

Source: own calculations in R.

```
                    Estimate Std. Error t value Pr (>|t|)
(Intercept)         -0.0256818 0.0002889-88.88 <2e-16 ***
num_bad_user_ratio  1.1575996 0.0037643 307.52 <2e-16 ***
--
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.008528 on 1005 degrees of freedom
Multiple R-squared: 0.9895,      Adjusted R-squared: 0.9895
F-statistic: 9.457e+04 on 1 and 1005 DF, p-value: < 2.2e-16
```

Figure 3. Coefficients of the websites model with basic measures and statistical test

Source: own calculations in R.

In the test part besides 10 004 users generated in the learning part of the simulation, one bad user of each kind was added, a thousand of good users and a hundred of new good websites (besides the entities from the learning part). Results are presented in the form of contingency tables (Figures 4–6), separately for the user and website classification, as well as for the total outcome. Calculating the ratio of correctly classified to all cases, one obtains very promicing results:

- for users model: 0.9995,
- for websites model: 0.9972,
- for the total (users and websites) model: 0.9989.

It means, that 99.89% of cases were classified correctly. It must be emphasized, that both user-oriented and website-oriented models are dependent from each other. Better classification of websites is helping to find suspects users and vice versa without proper user recognition it is hard to determine if the website is not a fraud. That is why the number of detected frauds should be counted together for two models not only for one of them.

Probably it would be possible to improve this result by adding to the procedure variables reflecting the human behavior associated with the actions performed on the websites (like click, scroll, conversion).

	REAL USER (model value)	**FAKE USER (model value)**
REAL USER (true value)	167 260	0
FAKE USER (true value)	91	23 694

Figure 4. Contingency table for the result of the users model on testing data

Source: own calculations in R.

	REAL USER (model value)	**FAKE USER (model value)**
REAL USER (true value)	169 714	0
FAKE USER (true value)	533	20 798

Figure 5. Contingency table for the result of the websites model on testing data

Source: own calculations in R.

	REAL USER (model value)	FAKE USER (model value)
REAL USER (true value)	165 801	0
FAKE USER (true value)	213	25 031

Figure 6. Contingency table for the result of the total (users and websites) model on testing data

Source: own calculations in R.

5. Conclusions

The computer simulations of the described in this paper method of detecting the fraud traffic in Real Time Bidding system confirmed its usefulness. Based on performed simulations it is justified to draw the following conclusions:

1. Each model (users and websites) is characterized in high accuracy of classification good and bad users and websites.
2. The method understood as a combining of two models for user and website classification based on estimates of bid request fraud probability, seems to be an effective (in terms of accuracy and time complexity) approach for fraud traffic detection.
3. All parts of described algorithm are susceptible to parallelization.
4. Presented method is easy to generalize by adding extra parameters associated with the human behavior.

Such a promising results are encouraging for implementing the method in existing Real Time Bidding system. This certainly involves with many difficulties in the performance and technology aspects. However the workload may pay off not only in financial profits, but also in the increase in knowledge about the differences between the human behavior and its automated imitation.

References

Bernardelli M., *Method of QR decomposition's fast updates for linear regression models*, "Roczniki" KAE, z. 27, Oficyna Wydawnicza SGH, Warszawa 2012, pp. 55–68.

Berners-Lee T., Fielding R., *Uniform Resource Identifier (URI): Generic Syntax*, www.ietf.org. Network Working Group 2005 (retrieved: 2014.09.05).

Brendan L., Paes Leme R., Tardos E., *On Revenue in the Generalized Second Price Auction*, Proceedings of the 21st International World Wide Web Conference (WWW '12), 2012, pp. 361–370.

Caragiannis I., Kaklamanis Ch., Kannelopolous P., Kyropolou M., Lucier B., Paes Leme R., Tardos E., *Bounding the inefficiency of outcomes in generalized second price auctions*, "Journal of Economic Theory" 2012, vol. 156, pp. 343–388.

Chen Y., Berkhin P., Anderson B., Devanur N., *Real-time bidding algorithms for performance--based display ad allocation*, Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, New York 2011, pp. 1307–1315.

Edelman B., Ostrovsky M., Schwarz M., *Internet Advertising and the Generalized Second-Price Auction: Selling Billions of Dollars Worth of Keywords*, "American Economic Review" 2007, vol. 97(1), pp. 242–259.

Friedman J.H., *Fast Sparse Regression and Classification*, Technical Report, Standford University 2008.

Shuai Y., Jun W., Xiaoxue Z., *Real-time bidding for online advertising: measurement and analysis*, Proceedings of the Seventh International Workshop on Data Mining for Online Advertising, New York 2013, article no. 3.

* * *

Wykrywanie oszustw w systemie *Real Time Bidding* – podejście panelowe

Streszczenie

Celem artykułu jest przedstawienie zastosowania modelowania ekonometrycznego do wykrywania prób oszustwa w systemie *Real Time Bidding*. Skuteczność proponowanej metody została zweryfikowana przez symulacje komputerowe. Metoda składa się z dwóch różnych modeli – modelu przeznaczonego do klasyfikacji użytkowników oraz modelu przeznaczonego do odróżniania rzeczywistych stron internetowych od tych specjalnie spreparowanych przez oszustów. Prezentowane modele są ze sobą ściśle powiązane i razem wydają się dość szybkim i efektywnym narzędziem do oddzielenia ludzkiego ruchu internetowego od tego generowanego przez boty.

Słowa kluczowe: *Real Time Bidding*, *big data*, wykrywanie oszustw, liniowy model prawdopodobieństwa

JEL: C55, C53, C33