http://doi.org/10.35784/iapgos.5376 *received: 26.07.2023 | revised: 19.10.2023 | accepted: 20.11.2023 | available online: 20.12.2023*

ADVERTISING BIDDING OPTIMIZATION BY TARGETING BASED ON SELF-LEARNING DATABASE

Roman Kvyetnyy[1], Yuriy Bunyak[2], Olga Sofina[1], Oleksandr Kaduk[1], Orken Mamyrbayev[3], Vladyslav Baklaiev[4], Bakhyt Yeraliyeva[5]

[1]Vinnytsia National Technical University, Vinnytsia, Ukraine, [2]Spilna Sprava Company, Vinnytsya, Ukraine, [3]Institute of Information and Computational Technologies of the Kazakh National Technical University named after K. I. Satbayev, Almaty, Kazakhstan, [4]Taras Shevchenko National University of Kyiv, Kyiv, Ukraine, [5]M. Kh. Dulaty Taraz Regional University, Taraz, Kazakhstan

Abstract. *The method of targeting advertising on Internet sites based on a structured self-learning database is considered. The database accumulates data on previously accepted requests to display ads from a closed auction, data on participation in the auction and the results of displaying ads – the presence of a click and product installation. The base is structured by streams with features – site, place, price. Each such structural stream has statistical properties that are much simpler compared to the general ad impression stream, which makes it possible to predict the effectiveness of advertising. The selection of bidding requests only promising in terms of the result allows to reduce the cost of displaying advertising.*

Keywords: advertising bidding, targeting, targeted advertising, click prediction

OPTYMALIZACJA OFERT REKLAMOWYCH POPRZEZ UKIERUNKOWANIE W OPARCIU O SAMOUCZĄCĄ SIĘ BAZĘ DANYCH

Streszczenie. *Rozważono metodę ukierunkowywania reklam w serwisach internetowych w oparciu o ustrukturyzowaną samouczącą się bazę danych. W bazie gromadzone są dane o wcześniej zaakceptowanych żądaniach wyświetlenia reklam z zamkniętej aukcji, dane o udziale w aukcji oraz o wynikach wyświetlania reklam – zarejestrowanie kliknięcia i instalacji produktu. Bazę tworzą strumienie z cechami – strona, miejsce, cena. Każdy taki strumień strukturalny ma właściwości statystyczne, które są znacznie prostsze w porównaniu do ogólnego strumienia wyświetleń reklamy, co pozwala przewidywać skuteczność reklamy. Selekcja tylko obiecujących pod względem wyniku zapytań ofertowych pozwala na obniżenie kosztów wyświetlania reklam.*

Słowa kluczowe: licytowanie reklam, ukierunkowywanie, reklama ukierunkowana, przewidywanie kliknięć

Introduction

Internet bidding became very important factor of economics support on condition of the pandemic. The on-line bidding often starts from the moment when consumer saw an interesting product on a banner of some site. The consumer can make click on the banner and he will be redirected to the corresponding shop, where he may buy the product or not.

From technical side of view, this process is looking as bidding of banner places by internet providers for advertisers. The provider controls the moment of a consumer visit to site and sends bid requests (BR) message to advertisers to participate in a closed auction of advertising (AD) places on the site. Since the display of advertising is paid, the advertiser tries to participate in those auctions where his advertising campaign will be successful. Advertiser specifies by targeting methods the BR messages which most closely correspond to purposes of an advertising campaign and have highest probability to receive paid events – consumer's click event (CLICK) and event of buy by consumer of advertised product (INSTALL). In the Real Time Bidding (RTB) this is exchange between Demand Side Platform (DSP) and Supply Side Platform (SSP). The aim of the targeting is to reduce cost of the CLICK and INSTALL.

The first step of the targeting is known as Native Targeting (NT). It is accessible by the way of the definition native parameters, such as country, region, time intervals, AD categories, list of preferable sites, creatives size, language, user gender and year of birth, etc. All known applications for media buying provide the ability to set NT parameters. The aim of the NT is to specify optimal stream of bid request messages and to limit RTB server load. The NT is not responsible to made the targeting effective, because, less than a tenth part of the BR in the received optimized by NT stream are useful in terms of getting CLICK and less than thousandth part give INSTALL.

At a current time, there is only one approach to made AD trades effective, which is widely using by professional companies – this is the CLICK prediction. There are many methods of its implementation. The modern methods are basing on using of Data Management Platform (DMP) in the manner of real time database. The database stores a history of participation of the BR messages parameters values in previous trades and related with them results. When current BR was received the history of the previous results associated with parameters of the BR create its rating and help to make a decision about participation in the trade. The DMP also includes statistical models of events occurring against the background of the BR stream.

The AD auction creates a problem for the SSP associated with a choosing of a minimal price for the lot received in the BR message so to win the auction. The closed bidding is the feature of the auction. Therefore, only the SSP which won the auction lot knows the lot selling price, offered by nearest competitor. So, the RTB needs in price strategy for forecasting a minimal winning price by the help of a dynamic model which reflects price changes on current bidding conditions.

1. Materials and research methods

The mathematical background of the AD utility estimation and forecasting is wide – from Bayesian probability model to matrix and tensor models with implementation using convolutional and neural networks. The models are intended to evaluate and predict auction winning process, AD utility parameters such as Click to Bid Ratio or Click-through-Rate (CTR), Install-through-Rate (ITR) and other. The RTB process is described in detail in [1]. There are two specifications of messages used in bidding transactions [6, 16]. The BR messages defined by these specifications are similar and include some equivalent parameters and some specific parameters which sign user and device location, thematic profiles by differ ways. The Open RTB specification [4, 11] was used in [1, 3]. Authors made review of some approaches to forecasting of the RTB price and AD budget optimization. There are known probabilistic models which taking into account both the behavior of the users the advertisers, based on history of the impression, the time or date of the impression, the presence of social functionality. The aim of many authors efforts is to predict the bid value while acquiring an impression at a lowest cost. Their strategy is based on a win model which predicts the winning price based on a regression model. The authors of [1] have evaluated the feasibility of applying forecasting approach using autoregressive integrated moving average model to predict the bid prices. The accuracy of the predictions was very low due to dynamism of the RTB process. At the next step they developed

a dynamic programming algorithm to bid for the impressions that operates over a set of consecutive time periods. The algorithm follows a model which adjusts its properties for the next bid period based on the prior period behavior. The model adjusts bid price with account of budget strategy and reached wining results. Authors affirm that such approach can adapt the bidding process in the RTB successfully. The performance of the algorithm depends on duration of chosen bid period. The AD utility and its relation with bidding model was not considered.

The idea to take into account the value of the highest competing bid for prediction of the AD utility was considered in [4]. The utility was interpreted as a profit of the bidding. Expected utility is the integral of distribution of the highest competing bid on condition of bid sale price and corresponding them winning and losing events. The integral was evaluated by using a given set of historical events. So, the highest bid price is a factor of AD utility from the point of view of bidding profit.

Bayesian approach to the prediction problem yields the model of Logistic Regression (LR) [6,11] which is widely used for bidding utility prediction with account of data structures. The next value of the utility parameter, for example the CTR, can be evaluated by using the current and previous values vector and weighting vector which is estimating for each step on the condition that logarithm of click event and not click event probabilities relation (the loss function) is equal to scalar product of the vectors. The procedure of the weighting vector evaluation is a nonlinear optimization problem. Different ways of its iterative solution and implementation are discussed [2, 5, 9, 10]. But LR based methods cannot capture higher order interactions between features, which have proved to be important in the CTR prediction [5, 7]. Therefore, the LR is using in combination with data structures which reflect bidding process [14].

The factor model (FM) of the utility prediction was offered in [13]. The model is basing on pairwise interactions of ID sources. The ID are identifiers of advertisers. The pairwise interactions create a matrix. Every model matrix cell has its own set of features for the factorization. The final prediction is the sum of all pairwise feature dot products. The advantage of this approach is that the information from the test samples set is capable for predicting for new IDs.

The problem of CTR prediction by the DSP is to calculate the bid price according to the estimated CTR is presented in [14]. Tensor factorization model of bidding data and click events was offered. The model presents coupled interactions between user, publisher and advertiser as third order tensor. The tensor is sparse and therefore it can be effectively factorized using high order singular value decomposition. As the result, the CTR is estimated in a manner of a linear sum of some factors related with interaction model. The analogues sparse structures and their parameters optimization by clear and latent factors retrieving using the Method of Factorization Machines (MFM) are considered in [8, 12, 15]. The MFM is model class that combines the advantages of Support Vector Machines with factorization models. The model presents the click probability or the loss function as the second order nonlinear function of a feature vector of impressions. The field-aware version of the MFM [8, 17] is presented by released an open source software.

2. Model experiment

As it follows from above short overview, the RTB utility is an object of influence of many factors. The factors can be signed directly, the clear factors as IDs and bid price, or indirectly, the latent factors evaluated using the utility (CTR) probability model. The considered approaches give the integral feature of a bidding process in the manner of the CLICK probability function. The forecasting function value can be evaluated using its previous samples with account of LR model vector or as a sum of latent factors given by factorization of a matrix which cells contain information about bidding events – interactions between users, advertisers and publishers, events of CLICK and INSTALL. Accumulation of such matrix on condition of big data is difficult process.

The other way is to separate full complex BR stream on independent threads of trades which has stable characteristics on cost and sales. It can be assumed that the products of same shop at same price are characterized by same quality in respect to bidding events probability. So, the model of CLICK and INSTALL prediction may be created for each thread that is indicated as *{shop, product, price}*.

AD consumers can be considered as a members of users classes which are characterized by device. The device is characterized by operating system (OS), its version (OSV), device manufacturer (make) and device model. These parameters form the fields vector: *device={os, osv, make, model}*. Device parameters are important because they effect on AD display. Other information about user may be accounted in a pre-targeting stage by NT schemas. The model of CLICK and INSTALL prediction can be created for each user class.

It is assumed that the combination of effective sources and consumers at a moment of high probability of paid event may give resultative advertising.

Database structure. In mobile application site is signed as a *bundle*. AD banners are signed as creatives with identifiers – *creativeId*. Each site has some banners of different size and visibility. A cost of advertising exhibition on a banner may vary. It is pointed by the start price of the auction and is signed as *minPrice*. So, the vector *{shop, product, price} = {bundle, creativeId, minPrice}* characterizes a source of bidding events.

Table 1. Database thread parameters structure

Field	Comment
rating	rating of click and install
bid, win, click, install	numbers of events
winPrice, winPriceVar	win price and its variation
winbidPrice, winbidPriceVar	win bid price and its variation
lostbidPrice, lostbidPriceVar	lost bid price and its variation
ban	ban counter
wdt=[wdt0,…., wdt23] cdt=[cdt0,…cdt23] idt=[idt0,…idt23]	win, click, install daytime distribution
Plast	last click probability
Tclick	last click time
$[\Delta t_0, \ldots, \Delta t_{N-1}]$	vector of time intervals between clicks
$[\Delta n_0, \ldots, \Delta n_{N-1}]$	vector of bids number between clicks
weight	weight coefficients vector

Each *bundle* parameters can be presented as the cell database. The cell contains some threads signed by *{creativeId, minPrice}*. An example of the cell thread data structure is shown in table 1. The cell array can be stored as structured database. The fragment of the cell includes three threads is presented in Appendix in JSON format.

The consumers database thread is like in table 1 with exclusion of some fields.

Rating filtration. The step-by-step algorithm of bids rating learning and selection of bids with high probability of the events is presented in table 2.

Algorithm starts from learning of registered cell's thread behavioral characteristics. The first operation is reading of data vector pointed by three parameters values from the incoming BR: *{bundle, creativeId, minPrice}*. Each new value of *bundle, creativeId, minPrice* causes creation of new cell or cell's thread in learning mode. The initial *bidPrice* = 1.1·*minPrice* and this price is increasing in the bidding process up to the rating of win to bid reached some level, not less 0.05, for an example. When wins number reached W_{min} value (10 or 20) then a selective filtration can be made by the thread performance evaluation using the specific rating of click with account of installs with their weight and *winPrice*. If *rating* is more than RL_{min} value (0.01 or 0.05) the performance is high and the cell is putting into working mode. Otherwise, the cell is banned for a sometime by the ban time counter. The working mode continue up to *rating* become less then RW_{min} value (0.001 for an example).

The decision that cell's thread is not appropriate for bidding is made on this condition. The system controls the averaged *rating*. If the rating falls, then the *bidPrice* increases in order to increase the number of effective wins. Media buyer can determine the financial limit for participation in trades in the form of a cost of the paid event. Then the constraint *winPrice* < *event_cost* ·*rating* can be defined. This constraint means that maximization of the ratio *max***(**rating/*winPrice***)** is desirable. Each cell thread in table 1 contains three vectors of 24 values: **wdt**; **cdt***;* **idt**. Vectors are designed to fix the distribution of the number of events WIN, CLICK and INSTALL by hours of the day. Instead of total parameters in table 2 points 6, 7 can use values related with current time of bidding, t_b, for example, for *win*: $win(t_b) = wdt[mod_{24}(t_b - 1)] + wdt[t_b] + wdt[mod_{24}(t_b + 1)]$.

Table 2. Algorithm of rating filtration

	Operation	Return	Comment
0	*if* time of cell search expired: new cell creation	*bidPrice* = 1.1·*minPrice*	Bidding is in learning mode
1	*if* thread with parameters *creativeId* or *minPric*e is not found – create new thread		
2	*if ban* > 0 *ban* decrement	"no content"	thread is banned
3	*if ban* = 0	*bidPrice* = 1.1·*minPrice*	bidding is in learning mode
4	*if win* < W_{min}: W_{min} – minimal number of wins for decision making about cell thread performance	if *win/bid* < 0.05 *bidPrice* = 1.1·*bidPrice*	rating increase, W_{min} is 10 or 20
5	*if win* >= W_{min} & *click/win* < RL_{min} *ban=BanCount, bid=win=click=install*=0	"no content"	ban mode of the thread, RL_{min} – minimal rating of ckick in learning mode
	if win >= W_{min} & *click/win* >= RL_{min}	*bidPrice*	thread is in working mode
6	*if win* > W_{min} $rate_t = \frac{click + IW \cdot install}{win \cdot winPrice}$		RW_{min} = 0.02, *IW* – install weight in comparison with click: 200/500/1000
	if $rate_t$ > RW_{min}: RW_{min} – minimal rating in working mode	*bidPrice*	bidding
	else ban=BanCount, bid=win=click=install=0	"no content"	ban mode of the cell thread
7	$rate_{new} = \frac{rate_t + win \cdot rate}{win + 1}$ *if* $rating_{new}$ /*rating* < 0.9	*bidPrice* = 1.1· *bidPrice*	raise bid price if *rating* falls Update *rating*
8	*bid* increment *if* win *win* increment *if* click *click* increment *if* (install) *install* increment		Update events counters

The algorithm in table 2 can be defined as soft mode algorithm which includes procedures of learning and selective filtration. The limit state of the soft algorithm is the hard mode algorithm which includes items 6–8 of table 2. It executes only procedures of the selective filtration when all information about event sources and consumers is known.

Click prediction. Main part of expenses is related with frequent won bids which give rare paid events. The most rating sources and consumers do not give streams of paid events. Their paid events pass with some periodicity. Therefore, it is necessary to estimate the periodicity and to choose bids that follow with a similar periodicity. Then number of paid events of installs and corresponding to them events of click wouldn't be decreased significantly. So, the bids selection should be made from the regard point of CLICK prediction. The click events prediction could be made by evaluation of CLICK probability for current bid request by estimation of time intervals and bids number between consecutive clicks. The vectors of time intervals and the vectors of number of bids between click events in the thread database structure are intended for prediction of click. The number of click events relates with number of WIN which depends on the *bidPrice* level. The rating filter regulates the *bidPrice* level so that there are appropriate flows of WIN and CLICK.

The methods of Bayes-Poisson dynamic model, the regression models basing on the method of Support Vector Machines (SVM) with different kernels, the probabilistic methods, such as Hidden Markov Model (HMM), Gaussian Mixture Model (GMM), Cross Validation Model (CVM), etc., can be used for the click events prediction. The methods can be joined into a general schema with weights which reflect their accuracy.

Bayes-Poisson selective filter. The Poisson model of the click events stream can be used for click probability estimation at the current time. The conditional probability of a click event at time of bidding t_b on condition that the previous CLICK occurred at time t_c is the next:

$$P(t_b \mid t_c) = 1 - e^{-\lambda(t_c)\cdot(t_b - t_c)} \tag{1}$$

where $\lambda(t_c)$ is the events intensity at the time t_c. The intensity of the Poisson stream of click events can be evaluated using *N* time intervals Δt_i between consecutive CLICKs as following:

$$\lambda(t_c) = \left(\frac{1}{N} \Sigma i = 0...N-1 \Delta t_i \right)^{-1} \tag{2}$$

The intensity (2) is variable and therefore it should be evaluated for each click. The number of intervals *N* is defined as the order of the filter. The stability and effectivity of the model depends on the order.

The absolute probability of a CLICK at the moment t_b with account the probability $P(t_b \mid t_c)$ and click probability $P(t_b, t_c)$ at the moment t_c is defined by the Bayes formula.

$$P(t_b) = P(t_b \mid t_c) \cdot P(t_b, t_c) \tag{3}$$

where the click probability $P(t_b, t_c) = \frac{N_{click}(t_c)}{N_{win}(t_b)}$, $N_{click}(t_c)$, $N_{win}(t_b)$ – the numbers of events at the specified time. The Bayes-Poisson dynamic model shows that during a short interval after the event the probability of the next one is small and eventually tends to a value $P(t_b, t_c)$ which can change due to a change of the number $N_{win}(t_b)$ of wins.

In the case of consumer CLICK prediction there are several threads of the click events with different intensities, four fields of device parameters, for example. They can be considered as conditionally independent. Then with account (3) the total probability of the event can be represented as

$$P(t_b) \approx \sum_i P_i(t_b \mid t_c^{(i)}) \cdot P_i(t_b, t_c^{(i)}) / \sum_i P_i(t_b, t_c^{(i)}) \tag{4}$$

where the index *i* lists all model parameters values which are presented in the query and are used in the probability model. Expressions (3), (4) are known as the Bayes-Poisson regression (BPR). The probabilities $P_i(t_b \mid t_c^{(i)})$ define the weights with which the conditional probabilities are summed, their magnitudes determine the probabilities of the event at time t_b. This expression has to be normalized in order to know whether a high or low probability level was obtained at the time when the current request was received. Since there are a set of events that have taken place at the same parameters values, it can be estimated the probability $P(t_c)$ of the last of them and use it as conditional "one". The condition of a sufficient level of the current event probability can be set so that it is of the same order that probability of held events.

$$P(t_b) \geq P(t_c) \;\; or \;\; P(t_b) \subset O\big(P(t_c)\big) \tag{5}$$

where $O(\cdot)$ is a value neighbourhood. The application creates the following flag by condition (5) state.

$$flag_{poisson} = \begin{cases} +1, \; (5) \; is \; true; \\ -1, \; (5) \; is \; false; \\ 0, \; (5) \; is't \; defined. \end{cases} \tag{6}$$

The flags (6) are defined for bundle threads and for consumers threads: $flag_{poisson.bundle} \cdot flag_{poisson.cons}$.

Prediction of time interval and bids number between click events.

Regression methods. Let there is the vector of time intervals $\mathbf{v} = [\Delta t_i]_{i=0...N-1}$ between consecutive click events. Then the following time interval between last click and the next click can be found using the forecasting methods. The Python *scikit-learn* library allows to predict next value using a current values vector and a feature matrix by the SVM method with some kernels. The feature matrix is formed as multiplication of the matrices compiled by *2N*-1 time intervals.

$$\mathbf{F} = [F_{i,k}]_{i,k=0...N-1} = \left[\sum\nolimits_{j=0...N-1} \Delta t_{i+j} \cdot \Delta t_{k+j}\right]_{i,k=0...N-1}.$$

The predicted value is the function

$$v = v(\mathbf{F}, \mathbf{v}, kernel) \tag{7}$$

The *kernel* can be "*gaussian*", "*polynom*", "*linear*". It should be chosen the kernel which gives the best quality of the prediction.

Hidden Markov Model (HMM). The elements of the vector (6) can be considered as states of the HMM process of time intervals (or number of bids). The Python library *hmmlearn* implements the algorithm of prediction of the most probable state for each of elements of the vector $\mathbf{v}$. It returns the vector $\mathbf{p} = [p_i]_{i=0...N-1}$ of pointers which point on elements of the vector **v** which are most probable as the next states. If Δt_{N-1} is the last time interval then the next most probable interval is the $\Delta t_{p_{N-1}}$. It is similar for number of bids too.

Gaussian Mixture Model (GMM). The probability distribution function (PDF) of the elements of the vector **v** can be considered as a weighted sum of Gaussian functions with different parameters when PDF is not regular. The Python library *scikit-learn* implements the algorithm of prediction of the most probable value for each of elements of the vector **v** in accordance with estimated PDF. It returns the vector $\mathbf{g} = [g_i]_{i=0...N-1}$ of pointers which point on elements of the vector **v** which are most probable as the next ones, this is $\Delta t_{g_{N-1}}$

Integral statistic schema for events prediction. The Poisson model and regression models (7) with kernels – gaussian, polynomial, linear, give four estimates of the forecast value: Δt_{gaus}, Δt_{poly}, Δt_{line}. The HMM and GMM give two additional estimates – Δt_{hmm}, Δt_{gmm}. There is the problem to define the estimate with highest probability of the click event. It can be formed the vector of predicted time intervals between events as

$$\mathbf{dt} = [\Delta t_{gaus}, \Delta t_{poly}, \Delta t_{line}, \Delta t_{hmm}, \Delta t_{gmm}].$$

The vector **dt** can be considered as states of the random process. The metods HMM and GMM can label the most probable states for each element of the vector **dt** by label vectors **p** and **g**. The element $\Delta t_{\max(i)}$ on which maximal number of the labels point can be chosen as most probable. Also, it is similar for number of bids.

General algorithm of bid selection. The condition

$$t_b \geq t_c + \Delta t_{predict} \approx T_{last} \tag{8}$$

where $\Delta t_{predict}$ is the interval which is predicted by the approaches *A-C* with probability (5) not less than P_{last}, is the condition to select the bid at current time for trading. The flag

$$flag_{time.bundle} = \begin{cases} +1, \ (5)\,\&\,(8) \ is \ true; \\ -1, \ (5)\,\&\,(8) \ is \ false; \\ 0, \ (5)\,\&\,(8) \ is't \ defined. \end{cases} \tag{9}$$

can be defined using conditions (5) and (8).

By the same way the flag $flag_{time.cons}$ is defined for current cell thread of consumer fields.

Analogous prediction of the number of bids between click events gives the flags $flag_{bid.bundle}$ and $flag_{bid.cons}$.

The bids streams of any bundle cell and fields cells are unconditional random processes and therefore statistic of number of bids between click events can serve as feature of the following click event. So, the integral flag can be evaluated using the results of prediction:

$$\begin{aligned} flag = {} & w_{pb} \cdot flag_{poisson.bundle} + w_{pc} \cdot flag_{poisson.cons} + \\ & + w_{tb} \cdot flag_{time.bundle} + w_{tc} \cdot flag_{time.cons} + \\ & w_{bb} \cdot flag_{bid.bundle} + w_{bc} \cdot flag_{bid.cons}. \end{aligned} \tag{10}$$

The weight coefficients $w_{..}$ in (10) are defined as relation between number of true states of flags (6), (9) and number of true events. These coefficients give an advantage to most truthful predictor. With account of the rating filtration and flag (10) value, the condition of bid selection is the following:

$$\begin{aligned} & if \ \ rate_{bundle} \geq R_{\min_{bundle}} \ \ \& \ \ rate_{cons} \geq R_{\min_{cons}} \\ & \& \ \ flag > 0 = true \end{aligned} \tag{11}$$

The bidding will be made on true condition, the value of *bidPrice* is returned by the application in the response message, otherwise the state *"no content"* is returned.

3. Experimental results

The AD trades have different properties in respect to impressions on IOS and Android devices. Therefore, two examples of trades were used for the database learning.

The first one includes protocols of 682171 auctions of AD impression in IOS-devices. Each protocol includes records of messages of bid request, response with *bidPrice* value or "no content" message and bid result with information on occurred events. The bids were selected by NT filtration using only AD categories. The impressions gave 8271 CLICKs and 12 INSTALLs. As the number of INSTALLs is so small it is very difficult to predict CLICK with following INSTALL. Therefore, the same files were used in the tests to investigate the learning process by accumulation of results.

The second one includes protocols of the 9217055 auctions of AD impression in Android-devices. The bids were selected by filtration on categories too. The impressions gave 21495 CLICKs and 232 INSTALLs. There were used some of consecutive tests using these files to investigate the learning process.

Finally, the training process was investigated using protocols of 32458326 auctions, protocols files size is more than 106 Gbyte. The number of defined bundles is 75556 items. The database of cells includes up to 167403 threads which are presented in some JSON files with structure like in Appendix 1. The fields list which characterize consumers contains 10905 threads of statistical data in the case of device parameters using.

Table 3 shows results of the off-line bids selection learning in accordance with general algorithm (11) in the soft and hard modes of rating filtration in the case of IOS devices. There are presented initial data (high line) and selected data (low line). Two cases of *N*=4 and *N*=8 in (2) of the Poisson model (1) and HMM, GMM number of states are considered. There were made 10 cycles of learning and selective filtration in the soft mode. The install price was consecutively reduced step-by-step, averaged values are presented. At the next step followed 10 cycles of hard selection were made. The CTR multiplicity and ITR multiplicity show relative changes of the click and install rating of filtered data in respect to initial ones.

Table 3. Database learning for IOS devices

Number of states	**4**		**8**	
Mode	soft	hard	soft	hard
Learning cycles	0–9	10–19	0–9	10–19
Auctions	6821710			
	2404771	123068	2045975	143552
WIN	933620			
	119728	44573	107659	49826
CLICK	82710			
	14 954	8 573	13 382	9 341
INSTALL	**120**			
	48	**40**	**42**	**32**
CTR	0.0886			
	0.1249	0.1923	0.1243	0.1875
CTR multiplicity	**1.490**	**2.170**	**1.403**	**2.116**
ITR	0.000128			
	0.00040	0.0009	0.00039	0.00064
ITR multiplicity	**3.132**	**7.008**	**3.048**	**5.017**
Click price $	0.12			
	0.07	0.04	0.07	0.04
Install price $	**82.04**			
	21.31	**9.00**	**21.86**	**12.76**

Table 4 shows results of the off-line training in the case of Android devices. It was made one cycle of the learning and selective filtration in the soft mode and one cycle of the hard selection. Note, the doubling of CLICK ratio leads to a fivefold increase of INSTALL ratio. As it can be seen from tables, the results with $N = 4$ in (2) are better than in the case of $N = 8$. This fact reflects nonstationary of the bidding process. Same results were fixed for other values of N. The number of states $N = 4$ is taken as the base.

Table 4. Database learning for Android devices

Number of states	**4**		**8**	
Mode	soft	hard	soft	hard
Learning cycles	0	1	0	1
Auctions	9217055			
	2185100	441903	1742920	439486
WIN	473174			
	91735	32 201	78884	31984
CLICK	21495			
	5796	2862	5007	2928
INSTALL	**232**			
	150	**84**	**111**	**78**
CTR	0.0454			
	0.0632	0.0888	0.0634	0.0915
CTR multiplicity	**1.392**	**1.958**	**1.398**	**2.016**
ITR	0.00049			
	0.00163	0.0026	0.00178	0.00244
ITR multiplicity	**3.337**	**5.324**	**3.647**	**4.977**
Click price $	0.14			
	0.09	0.06	0.09	0.06
Install price $	**12.96**			
	3.57	**2.08**	**4.14**	**2.26**

Table 5. Filter training by all trades

Mode	soft	hard
Auctions	32458326	
	6631253	1677394
WIN	2527980	
	397488	145434
CLICK	112633	
	23737	12202
INSTALL	**405**	
	153	**90**
Win rate	0.0779	
	0.0599	0.0867
CTR	0.0446	
	0.0597	0.0839
CTR multiplicity	**1.3404**	**1.8811**
ITR	0.00016	
	0.00038	0.00061
ITR multiplicity	**2.38**	**3.87**
Click price $	0.11	
	0.07	0.05
Install price $	**29.74**	
	11.04	**7.43**

The learned filter was used in training to select trades across a variety of protocols, the results are shown in table 5. It was not reached double increase of the CTR multiplicity and so averaged INSTALL price was not decreased significantly. The simulation of the total trades had shown that total price has trend to reduce from cycle to cycle. The hard mode allows to obtain appropriate cost of the INSTALL for profitable trade.

Figures 1, 2 show events distribution of initial and selected by the general algorithm bidding events on the plane (*minPrice* × *winPrice*) obtained for all trades. Subfigures 1), 2), 3) show distributions of the events WIN, CLICK and INSTALL.

As it follows from figures, there are two types of the auction start price *minPrice*. The first type is in the range of 1$ per thousand impressions. The second type is in the range of 3–10$ per thousand impressions. The *winPrice* changes in the range from 1$ up to 20$ for the first type. The *winPrice* ~ *minPrice* for the second type of price.

The INSTALL distribution has shown that the main part of these events is shifted relatively to the main part of win and click events towards the increase of the *winPrice*, especially for the first type of *minPrice*. The most effective *winPrice* diapason is 10…16$. The high level of the install events corresponds to low level of the win events. The range of the *winPrice* in the bounds of 3…10$ per thousand impressions is interest too, but it includes least of half of WINs. So, the trades are not effective in term of INSTALL obtaining. The main part of install events for the second type of prices is also in the area of high prices *winPrice*.

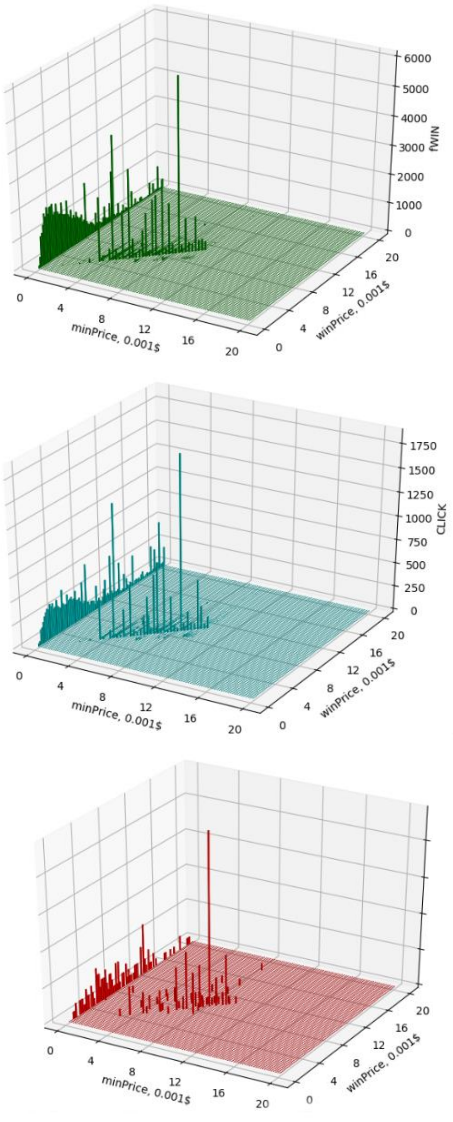

Fig. 1. Price distribution of initial data events

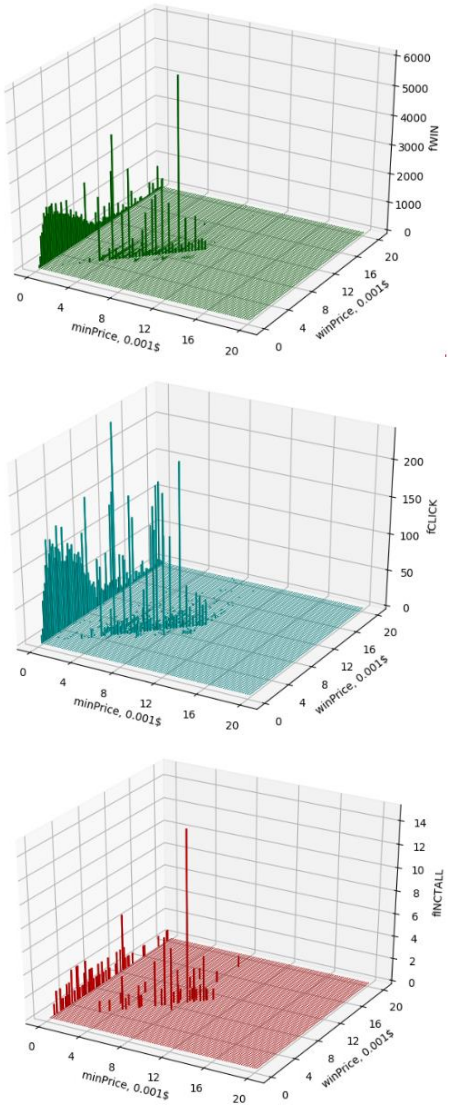

Fig. 2. Price distribution of filtered data events

It can be made the conclusion, that *winPrice* reflects the effectiveness of the trades in terms of obtaining the INSTALL. High level of *winPrice* indicates interest of competitors in trading. So, an estimate of the *winPrice* of current bid is very important characteristic of the bid effectiveness. As statistical analysis has shown, the averaged *winPrice* of each cell thread is relatively stable parameter, for example, its variation is up to 50% when *winPrice* < 4$, 30% when *winPrice* = 4…8$ and less than 10% when price is higher. So, the averaged *winPrice* of previous bids can be used as rating parameter.

The high cost of effective trades highlights the importance of analyzing the likelihood of paid events to trade only when the probability is highest.

As it seen in figures, the most significant peaks of the WIN and CLICK distributions of the selected bids are reduced. This means that intensity of some active sources decreased. These sources are associated with spending a significant part of the funds. The distribution of the INSTALL did not change significantly.

Note. The tracking of the picks in figures 1, 2 has shown, for example, the pick near the left bound on the level of ~15000 wins by the winPrice 13$ per thousand impressions in figure 1.1 corresponds to ~900 clicks in figure 1.2 and 5 installs in figure 1.3. So, the install price is ~39$. The filtration selected the BR which gave ~2500 wins in figure 2.1, ~225 clicks in figure 2.2 and 4 installs in figure 2.3. The install price became ~8$.

4. Conclusion

The main target of the off-line modeling is to determine the dynamic of the bidding process in dependence of system learning. As it follows from the obtained results, the ML model gradually increases the effectiveness of trades in accordance with filling and stabilization of the sources and consumers database.

As statistics has shown, the events cost decreasing is due to the selection of more rating trades and prediction of the events time of occurrence.

The prediction filters play significant role in price decreasing. Mean number of bids between CLICK is up to 300 and it became the same after filtering. But number of WINs decreases up to twice. This allows to reduce the expenses. As statistical investigation has shown, from 70% to 90% of clicks pass through prediction filter with true flags of events.

The hard schema can be used in finish stage of AD session for effectivity improving when all sources are known. For example, when average price of paid events reached some appropriate level. This will mean that the database statistic properties are synchronized with incoming messages property.

Users number which are ready to click and install the proposed product at each moment is bounded. Therefore, it is necessary, at the first, to ensure the fulfillment of the law of large numbers for obtaining a large number of random events of CLICKs and INSTALLs. It is necessary to process about 10–20 millions of bid request messages in each prime-time of one-day auction session, choose among them the most rating ones from the point of view to obtain events and only to make trades on them. The main aim of the statistical analysis is CLICK prediction.

The purpose of targeting advertising is to reduce the price of paid click-install events. The decrease of install price cannot occur at the one-time moment as a result of applying some algorithm of bid selection. The auction bids need to be studied before they will be selected. Therefore, the price of events can be reduced only as a result of the process of studying trades and applying the obtained knowledge to select the most promising advertising bids from the point of view to obtain a CLICK-INSTALL.

Each auction is carried out under certain conditions, which are determined by the composition of competitors, their interests and tactics of bids. Therefore, learning and training should be made at all stages of each session. Historical data should be obtained as averaged over a long period. The transfer of the moment properties of one bidding session to another session may not give positive result.

The first result of the research is that by dividing the total bid request flow into elementary threads as sources of the events and determining their dynamic and general ratings there was created the database in the manner of C++ STL containers or Python lists with supporting access in real time with appropriate delay on decision making. The mean delay time does not exceed 0.5 ms, the maximal time – 1.5 ms correspondingly.

The second result is the classification of AD consumers using parameters of their devices. The statistic characteristics of the parameters values are supported by data containers too.

The third result is application of complex two stage forecasting schema using three parameters and three types of regression based on the SVM method and HMM, GMM for CLICK prediction.

The connection of AD source and consumer with high level of rating in the moment of high level of event probability can give the resultative bidding.

The proposed method implements the AD targeting by the way of paid events prediction. The prediction is basing on evaluation of total and dynamic ratings of paid events along all bidding time with account of historical data and evaluation current event probability in each moment of the bidding. Such approach is usual in professional advertising practice.

The use of a neural network like TensorFlow of a constant structure with overloaded parameters corresponding to the thread of current bid is needed in a long time of decision making on participation in the trades, especially at the learning stage.

References

[1] Adikari S., Dutta K.: Real Time Bidding in Online Digital Advertisement. New Horizons in Design Science 9073, 2015, 19–38.
[2] Avila C. P., Vijaya M. S.. Click Through Rate Prediction for Display Advertisement. International Journal of Computer Applications 136(1), 2016, 18–24.
[3] Bisikalo O., Kharchenko V., Kovtun V., Krak I., Pavlov S.: Parameterization of the Stochastic Model for Evaluating Variable Small Data in the Shannon Entropy Basis. Entropy 2023, 25, 184 [http://doi.org/10.3390/e25020184].
[4] Chapelle O.: Offline Evaluation of Response Prediction in Online Advertising Auctions. IW3C2, Florence, 2015, 943–944.
[5] Chapelle O., Manavoglu E., Rosales R.: Simple and scalable response prediction for display advertising. Transactions on Intelligent Systems and Technology (TIST) 5(4), 2015, Article No. 61, A1–A34.
[6] IAB 2014. OpenRTB API Specification Version 2.2. http://www.iab.net/media/file/
[7] Jahrer M., Töscher A., Lee J.-Y., Deng J., Zhang H., Spoelstra J.: Ensemble of collaborative filtering and feature engineered model for click through rate prediction. Proceedings of KDD Cup 2012 Workshop, Beijing 2012, 1222–1230.
[8] Juan Y., Zhuang Y., Chin W.-S., Lin C.-J.: Field-aware Factorization Machines for CTR Prediction. RecSys'16, Boston, 2016, 43–50.
[9] Kondakindi G., Rana S., Rajkumar A., Ponnekanti S. K., Parakh V.: A Logistic Regression Approach to Ad Click Prediction. Machine Learning Project, 2014, 399–400.
[10] McMahan H. B., Holt G., Sculley D., Young M., Ebner D., Grady J. et. al. Ad Click Prediction: A View from the Trenches. KDD'13, Chicago, 2013, 1222–1230.
[11] Nigam K. L., Afferty J., McCallum A.: Using maximum entropy for text classification. IJCAI-99 1, 1999, 61–67.
[12] Pan Z., Chen E., Liu Q., Xu T., Ma H., Lin H.: Sparse Factorization Machines for Click-through Rate Prediction. IEEE 16th International Conference on Data Mining, 2016, 400–409.
[13] Richardson M., Dominowska E., Ragno R.: Predicting clicks: estimating the click-through rate for new ads. ACM, 2007, 521– 530.
[14] Sree Vani M.: Prediction of Mobile Ad Click Using Supervised Classification Algorithms. International Journal of Computer Science and Information Technologies 7 (2), 2016, 623–625.
[15] Ta A.-P.: Factorization Machines with Follow-The-Regularized-Leader for CTR prediction in Display Advertising. IEEE International Conference on Big Data, 2015, 2889–2891.
[16] The Real-Time Bidding (RTB) Protocol specification, 2016 https://developers.google.com/ad-exchange/rtb
[17] Zhang W., Yuan S., Wang J.: Optimal Real-Time Bidding for Display Advertising. KDD'14, New York, 2014, 1097–1105.

D.Sc. Roman Kvyetnyy
e-mail: rkvetny@sprava.net

Professor of Department of Automation and Intelligent Information Technologies, Vinnytsia National Technical University. Scientific interests include modeling of complex systems and decision-making under conditions of uncertainty (probabilistic and interval methods), modern methods of data processing. The main direction of scientific activity is development of methods and tools for mathematical modeling and information processing in computerized systems of automation and control.

http://orcid.org/0000-0002-9192-9258

Ph.D. Yuriy Bunyak
e-mail: iuriy.buniak@gmail.com

Spilna Sprava Company, Vinnytsya.
The main direction of scientific activity is investigation of the methods of optimization in big data, signals and image processing – denoising, deblurring, object recognition, also using big data.

http://orcid.org/0000-0002-0862-880X

Ph.D. Olga Sofina
e-mail olsofina@gmail.com

Ph.D., senior lecturer of Department of Automation and Intelligent Information Technologies, Vinnytsia National Technical University
The main direction of scientific activity is modern methods of data and image processing, namely methods of filtering textured images and identifying extraneous objects on their background, as well as methods of removing blurring of the image.

http://orcid.org/0000-0003-3774-9819

Ph.D. Oleksandr Kaduk
e-mail: o.kaduk@gmail.com

Associate professor of Computer Engineering Department, Vinnytsia National Technical University,
The main direction of scientific activity is modern methods in development of reliable AC and DC conversions, data and image processing, namely methods of filtering textured images and identifying extraneous objects on their background, as well as methods of removing blurring of the image.

http://orcid.org/0009-0001-2388-9813

Ph.D. Orken Mamyrbayev
e-mail: morkenj@mail.ru

Deputy Deputy General Director in science and Head of the Laboratory of Computer Engineering of Intelligent Systems at the Institute of Information and Computational Technologies of the Kazakh National Technical University named after K. I. Satbayev and associate professor in 2019 at the Institute of Information and Computational Technologies. Main research field: machine learning, deep learning, and speech technologies.

http://orcid.org/0000-0001-8318-3794

M.Sc. Vladyslav Baklaiev
e-mail: vladvlad03072000@gmail.com

Software Engineering, Taras Shevchenko National University of Kyiv, Ukraine
Research area: moderate experience in development of console and web applications. Basic understanding of Spring framework. Able to provide CI/CD of applications based on Maven phases and Docker-Compose. Python: used for implementing academic level scripts, applications and delimiter separated files. JavaScript: Used in combination with framework Vue.js for implementing front-end projects.

http://orcid.org/0009-0008-5767-6964

Ph.D. Bakhyt Yeraliyeva
e-mail: yeraliyevabakhyt81@gmail.com

Senior lecturer of the Information Systems Department, Faculty of Information Technology, M. Kh. Dulaty Taraz Regional University, Taraz, Kazakhstan.
In 2023 completed a full course of Ph.D. program in "Automation, electronics, electrical engineering and space technology" at Lublin University of Technology (Lublin, Poland), and works as a senior lecturer at Taraz Regional University named after M.Kh. Dulati. The direction of the dissertation research was based on the development of a method for measuring spatial strain in composite materials operating under mechanical perturbation.
Research interests: fiber optic technologies, information systems, Internet of Things and blockchain technologies.

http://orcid.org/0000-0002-8680-7694