Reserve price optimization with header bidding and Ad Exchange

Citation for published version (APA):
Refaei Afshar, R., Zhang, Y., Firat, M., Kaymak, U., Izzet Metin, A., Tarakcioglu, G., & Baş, C. (2020). Reserve price optimization with header bidding and Ad Exchange. In *2020 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2020* (pp. 830-835). Article 9283479 Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/SMC42975.2020.9283479

DOI:
10.1109/SMC42975.2020.9283479

Document status and date:
Published: 14/12/2020

Document Version:
Accepted manuscript including changes made at the peer-review stage

Reserve price optimization with header bidding and Ad Exchange

Reza Refaei Afshar
Eindhoven University of Technology
Eindhoven, Netherlands
r.refaei.afshar@tue.nl

Yingqian Zhang
Eindhoven University of Technology
Eindhoven, Netherlands
YQZhang@tue.nl

Murat Firat
Eindhoven University of Technology
Eindhoven, Netherlands
m.firat@tue.nl

Uzay Kaymak
Eindhoven University of Technology
Eindhoven, Netherlands
U.Kaymak@tue.nl

Ali Izzet Metin
Triodor Software
Istanbul, Turkey
izzet.metin@triodor.eu

Gönenç Seçil Tarakçıoğlu
Triodor Software
Istanbul, Turkey
g.tarakcioglu@triodor.eu

Coşku Baş
Triodor Software
Istanbul, Turkey
cosku.bas@triodor.eu

***Abstract*—The extremely high turnover of online advertising makes it one of the most important sources of income for many online ad publishers. Advertising through world wide web is mainly performed by Real Time Bidding in which the advertisers and the publishers participate to online auctions for trading the ad slots. Publishers usually set the reserve prices for their ad slots and any winning buyer in the auctions performed by ad exchanges has to pay at least the value of reserve price. Header bidding is a way of real time bidding and it becomes very popular, but how to use it together with advertising Exchanges (AdX) to achieve good revenue for online publishers is not well studied. In this paper, we propose a method that makes use of the historical auction data from header bidding and AdX to learn and optimize the reserve price for AdX. We propose a method based on supervised learning and survival analysis to increase the reserve price. The method assumes no information about current auctions and the bids of header bidding and AdX response are predicted and used to determine the highest possible reserve price. The experiments with real-world auction data show the promising results of our method in increasing the expected revenue of online publishers.**

***Index Terms*—Online Advertising, Header Bidding, Supervised Learning, Survival Analysis**

I. Introduction

In the past few years, the turnover of online advertising has grown dramatically and its revenue is essential for most of website owners. Thanks to online advertising, the companies offer free services and applications to the users and their expenses are mostly covered by the revenue of online advertising. Display advertising is a type of online advertising in which one or more blocks named *ad slots* are placed in a webpage to show ads mainly via *Programmatic Advertising* [1]. Programmatic advertising is the use of computer programs to automate the process of advertising. Real Time Bidding (RTB) is a programmatic advertising approach in which the ad slots are sold in real time auctions [2]. In RTB, the ad publishers and the advertisers enter the auction as sellers and buyers respectively. The items to be sold are the ad slots and the bidder with the highest bid wins the auction. *Ad Networks* or *Ad Exchanges (AdX)* are the stakeholders responsible to run the auctions [3]. The auctions are usually *second price auctions* [4], in which the ad publisher determines a minimum price called *reserve price* for each ad slot. If the highest bid is greater than the reserve price, the maximum of reserve price and the second highest bid is the paid price for the ad slot. Otherwise the ad slot is not sold in the current auction.

There are many available ad networks for a publisher and depending on its strategy to enter the auctions and chose the ad networks, the RTB can be based on *Waterfall Strategy* or *Header Bidding (HB)* [5], [6]. In waterfall strategy, the ad networks are selected sequentially [7]. When a webpage containing an ad slot is loaded, an *impression* is generated and an *ad request* containing a reserve price is sent to the first ad network. If the auction of this ad network has a winner, the process is finished and the ad slot is filled with the winner's advertisement. Otherwise, another ad request is sent to the second ad network. This process continues until consuming all ad networks or reaching timeout [8]. In header bidding, the ad publisher sends all the ad requests simultaneously to all of the ad networks [9]. The ad networks are called *Header Bidding Partners (HBPs)* and they are connected to separate sets of advertisers [10], [11]. Each HBP provides a bid greater equal than zero and the winner is the bidder with the highest bid. Unlike second price auctions, the publisher receives the highest bid as revenue.

Recently, the publishers use a combined framework of header bidding and AdX. In this framework, the ad requests are sent to HBPs and the highest bid is used as reserve price for AdX. The highest bid may not be an optimized reserve price and the AdX may outbid higher reserve prices. In this paper, we focus on the problem of setting the reserve price for AdX. The proposed method starts with predicting the highest bid of HBPs. Then, if AdX outbids the highest bid of HBPs, the survival curves learned by historical bids are used to increase the reserve price. Our method involves two steps; 1) predicting the response of AdX using predicted highest bid, and 2) increasing the floor price using survival analysis. Like [12], we use survival analysis to determine the first reserve price that AdX fails to outbid. Unlike [12], we model a sequence of components for the publisher to determine the expected revenue prior to enter any auctions. In this way, the publisher can possibly skip HBPs or AdX to reduce the ad loading time and to improve user experience. Furthermore, our method works as a reserve price optimizer, not merely a prediction model to predict the reserve price failure. In sum, our contributions are modeling the process of header bidding as a pipeline of supervised learning components and finding winning probability curve of AdX by employing survival analysis in which the time is modeled by reserve price.

II. Literature Review

Waterfall strategy is the typical approach in ad exchanges. Typically the ordering of ad networks in waterfall strategy is predefined and fixed [13]. However, in [14], a reinforcement learning based approach is proposed to dynamically determine the ordering of the ad networks for each ad slot. Several works focus on setting the reserve price in waterfall fashion including PSO-based algorithm [15], Fuzzy-based approach [16], supervised learning [17] and reinforcement learning based algorithms for bidders [18].

Header bidding is introduced to relieve the limitations of waterfall strategy and it is becoming more and more popular [19]. In [20], the problem is to sequentially optimize the bids of *Supply Side Platforms (SSPs)* in order to maximize their revenue. SSPs are the stakeholders that help publishers to participate in the auctions [20]. The problem is modeled as a stochastic contextual bandit problem where the context is the information of user and ad slot. Then, a variant of Thompson Sampling is used to optimize bids in a sequence of auctions. In [19], a statistical analysis is performed on the systems using header bidding. A mechanism named HB-Detector is designed and implemented to find out the internal behavior, different implementations of HB, effects on user experience, its revenue and the dominating entities.

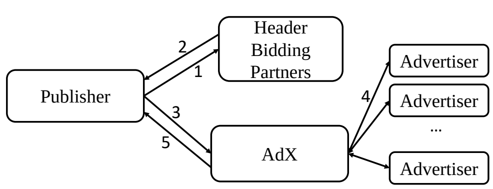

Fig. 1. Overview of HB and AdX systems.

The system containing the combination of HB and AdX is shown in Fig 1. (1) The requests are sent to HBPs and (2) the highest bid is received. (3) This highest bid is considered as the reserve price of AdX and the request is sent to AdX. (4) AdX connects with advertisers and run a second price auction to find the highest bid. (5) AdX send the response determining the winner of the auction. The impression goes to AdX if at least one advertiser participated in AdX's auction outbids this reserve price. Otherwise, the impression goes to the bidder of HBPs with the highest bid and the advertisement is shown accordingly. For simplicity we only mention AdX rather than advertisers connected to AdX in the rest of this paper. As mentioned in [12], the reserve price set in this manner is not optimized. Hence, the authors focus on the problem of reserve price failure where the model determines the probability that no advertiser in the AdX will outbid the reserve price given the information of ad impression. A parametric survival model is used to solve this problem considering the challenge of limited user profile information due to data protection regulation. Based on [6], by using HB, some of the ad slots are sold before going to AdX and this affects on the revenue of AdX. For this reason, the problem of adjusting the internal reserve price of AdX is modeled by a parametric stochastic model. The AdX gets the reserve price of publisher and alter it according to its own performance measure to maximize its revenue.

III. Reserve Price Setting

In this section, the proposed method for setting the reserve price in combined header bidding and AdX systems is presented. As shown in Figure 2, the process starts with predicting the highest bid of the HBPs which is used to predict whether AdX can outbid this value or not. If so, the haghest bid is considered as the minimum reserve price and the increase strategy is employed to set the highest reserve price that AdX can still outbid. The three components are elaborated in the following subsections.

A. Predicting Highest Bid

Upon generating an impression, a set of ad requests are concurrently sent to HBPs. The HBPs provide their bids and the highest bid is sent to AdX as reserve price. This process is straightforward if the publisher knows the bid values in advance. However, these information are not available before sending the ad requests to HBPs. For this reason, predicting the bids for a particular ad slot is the first important step of our proposed method.

The set of information regarding to an ad slot are used to predict the bids. This information are shown in Table I. We assume a limited set of features without special processed data to make the approach applicable on a vast number of online advertising systems based on header bidding. The bids for different ad slots may differ and the data might have outliers that affect the prediction. Before designing the prediction model, the outliers are removed.

This prediction is performed by a supervised learning algorithm that gets a feature vector containing the ad slot information and predicts the highest bid $\hat{f}$ of the HBPs. This bid value is used in the second step to predict whether AdX outbids this value or not. Feature vector used in this step is denoted by χ^{bid} and it is shown in (1). This vector contains the available features before sending ad requests and the features are explained in table I.

$$\chi^{bid} = (\varphi, \Upsilon, \xi, \ell, \tau) \tag{1}$$

TABLE I
GENERAL INFORMATION IN AN AD REQUEST.

Feature	Notation	Description
Slot ID	φ	The unique identifier of an ad slot
Webpage URL	Υ	The address of the webpage containing the ad slot
Auction ID	ρ	The unique identifier of the auction that is taken place in a particular HBP
Location	ℓ	The location of the ad slot in the webpage.
Size	ξ	The size of the ad slot (Width x Height)
Time	τ	Time and date of sending the ad request
HBP_i	h_i	The name or id of i^{th} header bidding partner
bid_i	b_i	The bid value of the i^{th} header bidding partner
Winner	ω	Indicates the entity (one of HBPs or AdX) that takes the impression
Highest Bid	f	Indicates the highest bid of HBPs (target value of bid prediction model)

B. Predicting AdX response

The second step is predicting the response of AdX. The highest bid is used as the reserve price for AdX and the response shows whether an advertiser can be found or not. A separate prediction model using supervised binary classification is used to predict the response of AdX. The binary classifier gets the feature vector including the predicted highest bid and returns the chance of outbidding the highest bid a.k.a. success probability. If the prediction model predicts that AdX outbids the highest bid, there is an opportunity to increase the reserve price and evaluate AdX again. However, it is not possible in real time due to time limitations, i.e. the webpage should be loaded in few milliseconds. For this reason, a prediction model for predicting the response of AdX that provides the output in real time is necessary. The feature vector used in this step is shown in (2).

$$\chi^{win} = (\varphi, \Upsilon, \xi, \ell, \tau, \hat{f}) \tag{2}$$

where, $\hat{f}$ is the predicted value of f using the prediction model explained in III-A.

C. Optimizing the Reserve Price

Normally, the highest bid of HBPs is used as reserve price for AdX and the final winner of the ad slot is determined by comparing the response of AdX and the highest bid. However, increasing the reserve price entails increasing the revenue because AdX runs a second price auction and the revenue is the maximum of reserve price and the second highest bid. In order to increase the reserve price until a threshold that AdX cannot outbid, we incorporate the *Survival Analysis*, since our reserve price and wining probability of AdX can be modeled in the form of Survival Analysis. In this section, first a background on survival analysis is presented and then its modeling in the problem of this paper is discussed.

1) Survival Analysis Background: Survival analysis is a set of statistical tools that answers the questions regarding to the time of the first occurrence of a particular event [21]. For example, the time of death of a patient suffering from cancer, or the time that a user leaves a webpage can be analyzed using survival analysis. For these kinds of problems, a starting point a_s and an end point a_e in time are set for all the samples. The samples could be lifetime of patients or the view time of a webpage. Formally speaking, a random variable T represents the time until a particular event occurs. This random variable has a probability distribution function (PDF) $p(t)$ and a cumulative distribution function (CDF) $P(t)$. The survival function $S(t)$ is defined as the probability that the event has not occurred until time t. The functions $P(t)$ and $S(t)$ are shown in (3) and (4) respectively.

$$P(t) = \int_{a_s}^{a_e} p(t)dt \tag{3}$$

$$S(t) = 1 - P(t) = p(T \geq a_e) = \int_{a_e}^{\infty} p(t)dt \tag{4}$$

Usually, the functions $p(t)$ and $P(t)$ are not known and data is used for finding survival function $S(t)$. A popular non-parametric approach for deriving $S(t)$ is called *Kaplan-Meier (KM) estimator* [22]. Based on KM estimator, the survival function is obtained by (5).

$$S(t) = \prod_{j:t_j \leq t} \frac{n_j - d_j}{n_j} \tag{5}$$

where, t_j is the time when at least one event occurred, n_j is the total number of samples survived prior to time t_j and d_j is the number of events occur at time t_j. In the next subsection, we model the problem of increasing the reserve price by survival analysis.

2) Reserve Price and Survival Analysis: Since we aim to increase the reserve price until AdX cannot outbid it, survival analysis can be helpful in determining that reserve price. We are inspired from [12] and use survival analysis to find the first reserve price value that AdX fails to find an advertiser based on a particular success probability. The *success probability* is the probability that a reserve price is outbid by AdX. In our analysis, the reserve price corresponds to the time in standard survival analysis. In order to provide data for survival analysis, we use the prediction model of section III-B and predict the response of AdX for each ad request and each reserve price. A unit price u is defined and the highest bid of HBPs is considered as the initial reserve price. The initial reserve price is increased by u at each step until the prediction model indicates that AdX fails to outbid. The obtained reserve price is used to develop a survival curve. In other words, for each ad request, the AdX response prediction model provides the highest reserve price that AdX outbids. The reserve price shown in (6) is used with (5) to derive the survival functions.

$$\hat{f}_e = \hat{f} + ku \tag{6}$$

where, $\hat{f}_e$ is the first reserve price that AdX fails to outbid, k denotes the number of increments and u is the price unit. Using the prediction model for deriving success probabilities, for each ad slot a separate survival function is obtained based on (4). For each ad request that AdX outbids based on the prediction model of section III-B, the reserve price is increased until the success probability is 0.5. This value can be adjusted depending on the risk-avoidance characteristics of the publisher. The corresponding reserve price is set as the new reserve price.

The process of increasing the reserve price starts with initializing $\hat{f}$ as the reserve price for a particular ad slot. If AdX outbids $\hat{f}$, $\hat{f}_e$ is set as the first reserve price that AdX fails to outbid using the survival functions. For example, if the reserve price is 0.02 and the success probability is 0.5 at reserve price 0.04, the initial reserve price is increased by 0.02 and the new reserve price is 0.04. The time complexity of predicting success probability is the reason that survival functions are used rather than prediction model. The time complexity of predicting success probability by survival function is $O(1)$ because it can be performed by a lookup table, while the same time complexity for a prediction model such as Random Forest - which works best for our problem - is $O(kn_{features}n_{trees})$ where $n_{features}$ is the number of features and n_{trees} denotes the number of trees [23]. Although $n_{features}$ and n_{trees} are fixed for a particular prediction model, their linear product with k might be large and this affects the loading time of a page. Therefore, the time complexity of predicting by the prediction model is higher than survival analysis. This is crucial in the context of online advertising, because each webpage has to be loaded in few milliseconds. The overview of the proposed method is shown in Fig. 2.

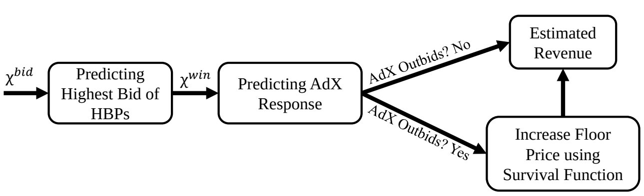

Fig. 2. Overview of the proposed method.

IV. Experiments and Results

The ad auctions data is obtained from *Azerion* that provides advertising platforms based on header bidding and ad exchange. Azerion is a media and technology company that provides service both from publisher and advertiser side. The data is not public. Two sets of ad requests are used to evaluate the method. The first set consists of the ad requests of the days 23, 24 and 25 of February 2020 which the first two days are considered for training and the last day for testing. The second set consists of 6, 7 and 8 of March 2020. Similarly, the model is trained on the first two days and the last day is left for evaluation.

The datasets contain the features shown in Table I. Due to diversity of bids provided by different HBPs, the outliers are identified and they are removed from further processing. The outliers are too large bids (¿¿1). Fig. 3 illustrates the box plot of the highest bid value of all ad requests of two days. As shown in this figure, majority of bid values are less than 0.5.

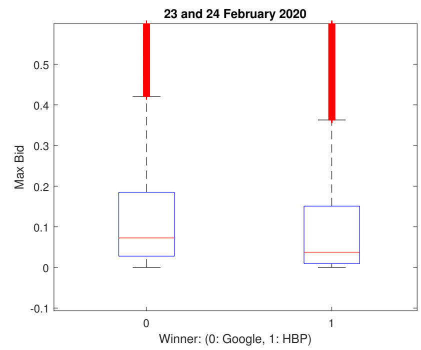

Fig. 3. Box plot of the highest bids. The data contains the ad requests of 23 and 24 of February 2020.

We evaluate the first two steps separately to show the quality of the predictions. The prediction model for predicting bids is performed on the feature vector shown in (1) to obtain $\hat{f}$ for each ad request. Categorical features like φ are converted to binary using One Hot Encoding. Different regression algorithms such as logistic regression, stochastic gradient descent, gradient boosting and random forest regressor are tested and Gradient Boosting is selected as it provides better performance [24]. The *scikit-learn* is used for implementation [25]. Table II contains the evaluation metrics for the regression task of the first step. The reserve prices range from zero to 0.6 and the mean square error of around 0.01 is acceptable since predicting bids is difficult due to unpredictable factors of auctions. Furthermore, the median absolute error and mean absolute error are around 0.05. These values are acceptable with respect to the range of reserve price. The prediction of AdX responses based on predicted bids also shows the quality of bid prediction model.

TABLE II
Performance of the bids predictor.

Test Data	Mean Absolute Error	Mean Squared Error	Median Absolute Error
25 February	0.07494	0.01281	0.05235
8 March	0.06794	0.00995	0.05192

Upon deriving the bids, they are used in feature vector (1) to predict AdX's response. If the response indicates outbidding the reserve price, the third step comes to play to optimize the reserve price. Different binary classification algorithms such as Decision Tree, Support Vector Machine, Stochastic Gradient Descent and Random Forest are tested and Random Forest with 100 trees and maximum depth 15 is selected due to its better performance. Fig. 4 shows the ROC curves of response predictor for the ad requests of two days as test data. The areas under curves are mostly higher than 0.74 which show a good binary classification especially in a dynamic environment like real time bidding.

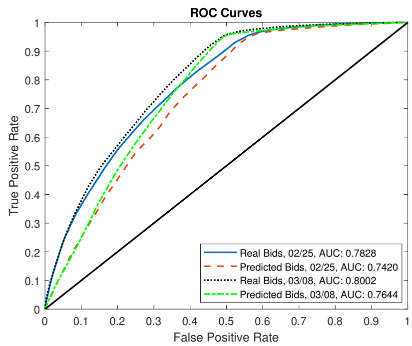

Fig. 4. ROC Curves of predicting the response of AdX using real and predicted bids. The input is χ^{win} and the output is zero or one predicting whether AdX outbids or not respectively.

The AdX response predictor is used for deriving the first reserve price that AdX fails to outbid by increasing $\hat{f}$. Let $u = 0.01$ be the price unit. This fixed value for price unit is selected because it is the rounded median of difference between highest bids. The highest reserve price for ad requests are generated by (6) and the generated data is used in (4) to derive a separate survival function for each ad slot. In order to evaluate the proposed method, two different revenues are defined and compared: (1) the real revenue: the revenue observed in the historical data and obtained by summing up the highest bids; and (2) the increased revenue: the revenue of increasing $\hat{f}$ and checking success probability by survival function which is implemented by *lifelines* tools [26]. These two revenues are illustrated in Figs. 5 and 6 for testing the method on the ad requests of February 25 and March 8, 2020.

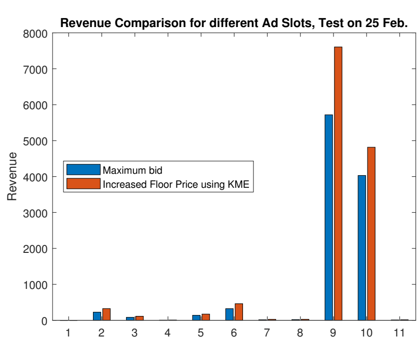

Fig. 5. Real revenue vs. increased revenues for different ad slots observed in the ad requests of February 25.

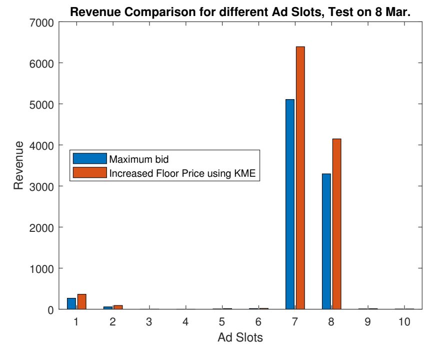

Fig. 6. Real revenue vs. increased revenues for different ad slots observed in the ad requests of March 8.

As illustrated in Figs. 5 and 6, increasing the reserve price as long as AdX outbids it, will increase the revenue considerably. The survival functions process the clean data excluding the outliers. Therefore, the domain of reserve price is between zero and 0.6 as shown in Fig. 3. The survival curves for two most frequent ad slots are shows in Figs. 7 and 8 for two different days.

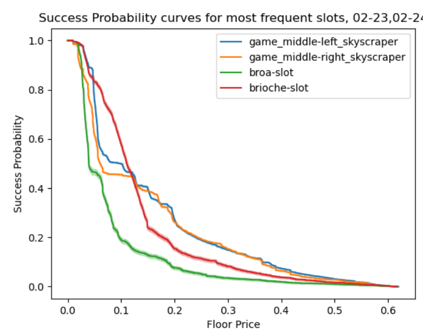

Fig. 7. Survival Curve for four most frequent ad slots of ad requests of Feb. 23 and 24.

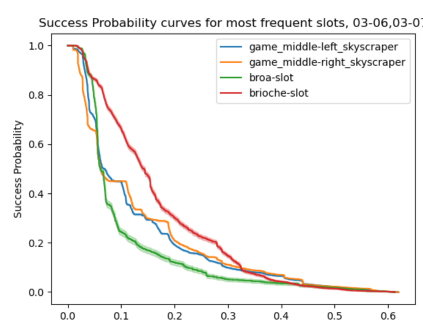

Fig. 8. Survival Curve for four most frequent ad slots of ad requests of March 6 and 7.

Figs. 7 and 8 show that the event of “sold by AdX” is almost imminent for small reserve price. For four most frequent ad slots where exist in almost all webpages of a certain website, highest bids obtained from HBPs are usually very small. These highest bids are used as reserve price, however the highest bid provided by AdX is higher than the highest bid of HBPs and the principle of second price auction bring about lower revenue than the highest possible one. Therefore, our proposed method can easily provide higher revenue.

V. Conclusion and Future Work

This paper proposes a sequence of components leading to adjust the reserve price for AdX in the systems based on header bidding and AdX. Our method assumes there is no information regarding the bids for current impression. Instead of sending the requests to HBPs and setting the highest bid as the reserve price of AdX, we use two prediction models for predicting the highest bid of HBPs and the response of AdX. Depending on AdX response, our method determines whether to send the impression to HBPs or to increase the reserve price of AdX. The method not only improves the loading time by suggesting to send the request to either HBPs or AdX rather than both, it increases the revenue by maximizing the reserve price.

The method is a combination of supervised learning components and survival analysis inspired from [12]. Our method extends the aforementioned work to set the reserve price. In spite of previous works in the context of header bidding which focuses on the revenue of SSP and Ad Exchanges, our method works as a decision support tool for ad publishers and aims to increase their revenue yielded from online advertising.

Although the proposed method theoretically increase the revenue, its intrinsic power is emerged when it is used in real header bidding system. Hence, one main future work is to test the method in a real system and consider the possible factors in that environment. Furthermore, other approaches in determining reserve price can be compared with the survival functions as a future research. The main limitation of the work is the diverse behaviors of HBPs which makes the prediction of their bids challenging.

Acknowledgment

This work is supported by EU EUROSTARS (Project E! 11582). The authors would like to thank the Headerlift and R&D Teams from Azerion and Triodor in collecting the data.

References

[1] O. Busch, "Programmatic advertising," *New York: Springer*, vol. 10, pp. 978–3, 2016.

[2] Y. Yuan, F. Wang, J. Li, and R. Qin, "A survey on real time bidding advertising," in *Proceedings of 2014 IEEE International Conference on Service Operations and Logistics, and Informatics*, pp. 418–423, IEEE, 2014.

[3] J. Wang, W. Zhang, and S. Yuan, *Display Advertising with Real-Time Bidding (Rtb) and Behavioural Targeting*. 2017.

[4] S. Yuan, J. Wang, and X. Zhao, "Real-time bidding for online advertising: measurement and analysis," in *Proceedings of the Seventh International Workshop on Data Mining for Online Advertising*, pp. 1–8, 2013.

[5] B. Chen, S. Yuan, and J. Wang, "A dynamic pricing model for unifying programmatic guarantee and real-time bidding in display advertising," in *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*, pp. 1–9, 2014.

[6] R. Qin, Y. Yuan, and F.-Y. Wang, "Optimizing the revenue for ad exchanges in header bidding advertising markets," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 432–437, IEEE, 2017.

[7] R. R. Afshar, Y. Zhang, M. Firat, and U. Kaymak, "A reinforcement learning method to select ad networks in waterfall strategy," in *Proceedings of the 11th International Conference on Agents and Artificial Intelligence*, vol. 2, pp. 256–265, 2019.

[8] R. Refaei Afshar, Y. Zhang, M. Firat, and U. Kaymak, "Reinforcement learning method for ad networks ordering in real-time bidding," *Lecture Notes in Computer Science*, vol. 11978, 2019.

[9] A. Sayedi, "Real-time bidding in online display advertising," *Marketing Science*, vol. 37, no. 4, pp. 553–568, 2018.

[10] M. Zacharski *et al.*, "Dynamic header bidding configuration," Feb. 7 2019. US Patent App. 15/669,633.

[11] K. Graham, "Understanding header bidding partners," 2018.

[12] A. Kalra, C. Wang, C. Borcea, and Y. Chen, "Reserve price failure rate prediction with header bidding in display advertising," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2819–2827, 2019.

[13] S. Sluis, "The rise of 'header bidding' and the end of the publisher waterfall," 2015.

[14] R. R. Afshar, Y. Zhang, M. Firat, and U. Kaymak, "A decision support method to increase the revenue of ad publishers in waterfall strategy," in *2019 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr)*, pp. 1–8, IEEE, 2019.

[15] J. Rhuggenaath, A. Akcay, Y. Zhang, and U. Kaymak, "A pso-based algorithm for reserve price optimization in online ad auctions," in *2019 IEEE Congress on Evolutionary Computation (CEC)*, pp. 2611–2619, IEEE, 2019.

[16] J. Rhuggenaath, A. Akcay, Y. Zhang, and U. Kaymak, "Fuzzy logic based pricing combined with adaptive search for reserve price optimization in online ad auctions," in *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 1–8, IEEE, 2019.

[17] Z. Xie, K.-C. Lee, and L. Wang, "Optimal reserve price for online ads trading based on inventory identification," in *Proceedings of the ADKDD'17*, p. 6, ACM, 2017.

[18] H. Cai, K. Ren, W. Zhang, K. Malialis, J. Wang, Y. Yu, and D. Guo, "Real-time bidding by reinforcement learning in display advertising," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pp. 661–670, ACM, 2017.

[19] M. Pachilakis, P. Papadopoulos, E. P. Markatos, and N. Kourtellis, "No more chasing waterfalls: A measurement study of the header bidding ad-ecosystem," in *Proceedings of the Internet Measurement Conference*, pp. 280–293, 2019.

[20] G. Jauvion, N. Grislain, P. Dkengne Sielenou, A. Garivier, and S. Gerchinovitz, "Optimization of a ssp's header bidding strategy using thompson sampling," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 425–432, 2018.

[21] D. G. Kleinbaum and M. Klein, *Survival analysis*, vol. 3. Springer, 2010.

[22] E. L. Kaplan and P. Meier, "Nonparametric estimation from incomplete observations," *Journal of the American statistical association*, vol. 53, no. 282, pp. 457–481, 1958.

[23] G. Louppe, "Understanding random forests: From theory to practice," *arXiv preprint arXiv:1407.7502*, 2014.

[24] J. H. Friedman, "Stochastic gradient boosting," *Computational statistics & data analysis*, vol. 38, no. 4, pp. 367–378, 2002.

[25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[26] C. Davidson-Pilon, J. Kalderstam, N. Jacobson, P. Zivich, B. Kuhn, M. Williamson, sean reed, AbdealiJK, A. Fiore-Gartland, D. Datta, L. Moneda, Gabriel, D. WIlson, A. Parij, A. Moncada-Torres, K. Stark, S. Anton, M. S. Peña, L. Besson, K. Singaravelan, Jona, H. Gadgil, D. Golland, S. Hussey, R. Kumar, M. Begun, J. Noorbakhsh, A. Klintberg, A. F. Rendeiro, and A. Flaxman, "Camdavidsonpilon/lifelines: v0.24.8," May 2020.