Selective Auctioning using Publish/Subscribe for Real-Time Bidding

Sonia Slimani and Kaiwen Zhang
École de Technologie Supérieure, Montréal, Canada

Keywords: Real-Time Bidding, Online Advertising, Publish/Subscribe System, Top-k Filtering, Machine Learning.

Abstract: Real-Time Bidding (RTB) advertising has recently experienced a massive growth in the industry of online marketing. RTB technologies allow an Ad Exchange (AdX) to conduct online auctions in order to sell targeted ad impressions by soliciting bids from potential buyers, called Demand Side Platforms (DSPs). In the OpenRTB specifications, which is a well-known open standard protocol for RTB, the AdX sends bid requests to all DSPs for every auction. This communication protocol is highly inefficient since for each given auction, only a small fraction of DSPs will actually submit a competitive bid to the AdX. The exchange of bid requests to uninterested parties waste valuable computation and communication resources. In this paper, we propose to leverage publish/subscribe to optimize the auction protocol used in RTB. We demonstrate how RTB semantics can be expressed using content-based subscriptions, which allows for selective dissemination of bid requests in order to eliminate no-bid responses. We also formulate the problem of minimizing the number of bid responses per auction, and propose combining top-k scoring with regression analysis with continuous variables as a heuristic solution to further reduce the number of irrelevant responses. We then adapt our solution by considering discrete machine learning models for a faster execution. Finally, we evaluate our proposed solutions against the OpenRTB baseline in terms of end-to-end latency and total paid price over time efficiency.

1 INTRODUCTION

1.1 Context

Real-Time Bidding (RTB) is a type of online advertising that allows websites to sell in real-time an ad impression to the highest bidder. RTB is a form of targeted advertising as each advertiser calculates its bid based on characteristics such as the banner size, the context of the web page, the user profile, etc.

OpenRTB is a specification of RTB system (IAB, 2016) which proposes open industry standards for communication between buyers and sellers of ad impressions. It is based on two main components: Ad Exchange (AdX) and Demand Side Platform (DSP). AdX is an intermediate agent between sellers and buyers of ad impressions. The AdX runs an auction for each ad request and sends corresponding bid requests to all eligible DSPs. Each DSP serves multiple advertisers, which can ask the DSP to run an ad campaign for a particular product based on target audience, predefined budget and campaign duration (Mullarkey and Hevner, 2015). Upon receipt of a bid request by the AdX, each DSP calculates the bid price based on the ad campaigns of its advertisers. The AdX collects bid responses to the auction and selects the winning DSP accordingly.

1.2 Problematic: Selective Auctioning

In the current OpenRTB specifications, the AdX broadcasts every bid request to all DSPs and awaits their bid responses. However, not every bid request will be of interest to each DSP, which may reply with a no-bid response (IAB, 2016). Bid responses with non-competitive bid amounts have no realistic chance of winning an auction, which adopts the Vickrey model (Section 2.3) . Finally, bid responses which arrive slowly or late will increase the end-to-end latency of the auction or be ignored completely, thereby degrading the efficiency of the exchange. In light of these observations, we conclude that OpenRTB wastes significant resources due to the sending of irrelevant bid requests and responses.

The impact of this overhead is characterized by two factors. First, bid request and bid response data are exchanged using HTTP in JSON (IAB, 2016), which is a heavy human-readable format. Second, a

Slimani, S. and Zhang, K.
Selective Auctioning using Publish/Subscribe for Real-Time Bidding.
DOI: 10.5220/0010111300260037
In *Proceedings of the 16th International Conference on Web Information Systems and Technologies (WEBIST 2020)*, pages 26-37
ISBN: 978-989-758-478-7

timeout value (called t_{max}) is fixed for each auction by the publisher, which is set to a sufficiently high value to collect all bid responses, including no-bid responses.

1.3 Objectives

In this paper, we propose a solution to reduce the communication overhead of RTB by selecting the subset of DSPs to be contacted for each bid request. We integrate Publish/Subscribe (Pub/Sub) semantics in the OpenRTB implementation to allow DSPs to express their interests as content-based subscriptions (Eugster et al., 2003). The AdX can then act as a pub/sub broker and filter DSPs based on their subscriptions, thus eliminating no-bid responses.

Furthermore, we can reduce the overhead by eliminating non-competitive and slow bids from each auction. We propose an extension which uses top-k filtering and regression analysis (with continuous variables) to select a subset of DSPs who are likely to submit competitive bids on time for a given auction. We then adapt our model in order to substitute continuous regression model with an efficient discrete one.

Since DSPs employ bidding strategies which do not require a complete history of past auctions, our solution can safely omit DSPs from the auction without compromising their integrity for future requests.

1.4 Contributions

In this paper, we provide the following contributions:

- We propose a pub/sub solution which maps RTB semantics, where bid requests are modeled as publications and DSP interests are expressed as content-based subscriptions (section 4.1),
- We formulate the problem of minimum bid responses selection for a RTB auction (section 4.2),
- We leverage top-k filtering on top of pub/sub to further filter DSPs based on predicted bid prices and responses time, which are computed using regression analysis (section 4.3),
- We adapt our solution in order to employ discrete machine learning models, thereby improving the speed of the top-k filtering (section 5) and finally,
- We implement our approach with OpenRTB and RabbitMQ and evaluate using the iPinYou dataset against a known baseline (section 6).

1.5 Paper Plan

We start this paper with the background in section 2 where we present different concepts used in our work.

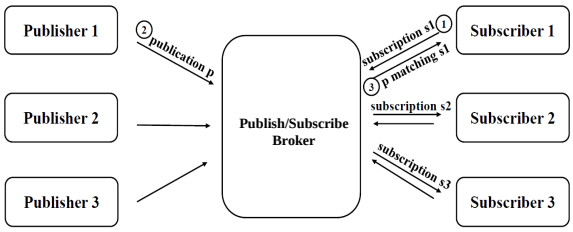


Figure 1: Publish/subscribe Overview.

In section 3, we survey related works in the literature. We continue with proposed solution in section 4, then we propose Discrete Prediction Models in section 5. We evaluate experimentally our solutions in section 6. Finally, we conclude in section 7.

2 BACKGROUND

In this section, we present different notions useful for understanding our work: Publish/Subscribe, Real-Time Bidding, Vickrey auction and OpenRTB specifications. We also describe the iPinYou dataset that we use to conduct our evaluation.

2.1 Publish/Subscribe

Publish/Subscribe system allows data producers (publishers) to send publications which are delivered to matching data consumers (subscribers) according to their interests, expressed as subscriptions.

As Figure 1 shows, (1) each subscriber can express interest in a particular publication by sending a subscription to the intermediary service of Pub/Sub. (2) Once the publisher produces this publication, (3) it delivers it to its corresponding subscribers.

According to (Eugster et al., 2003), pub/sub system is characterized by scalability, a dynamic topology and decoupling in terms of space (publishers and subscribers do not know each other and do not know who sends or receives or even how many entities participating in the interaction), time (no need for interaction at the same time) and synchronization (it is asynchronous).

Subscription types commonly supported include topic-based, type-based and content-based (Eugster et al., 2003). In our work, we use content-based subscriptions, which are expressed as predicates over key-value pairs (Canas et al., 2017). For example, the two following subscriptions SUB1 and SUB2 have two predicates each:

$$\text{SUB1} : (x = 3), (y > 4), \text{SUB2} : (x < 0), (y > 3)$$

A publisher publishes the following publication, which is only delivered to SUB1, since it satisfies all

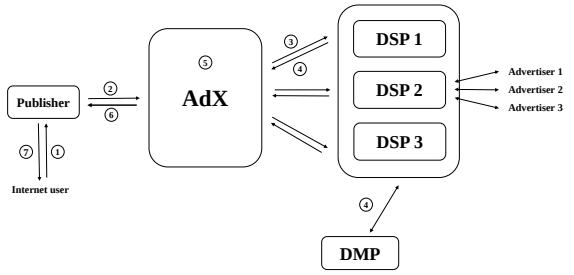


Figure 2: RTB Process Overview.

of its predicates:

$$\text{PUB} : (x = 3), (y = 5)$$

In Section 4.1, we use the aforementioned model to express DSP bidding interests as content-based subscriptions, while bid requests are modeled as content-based publications, carrying OpenRTB data as key-value pairs, which can be used to match against subscriptions by the pub/sub system.

2.2 Real-Time Bidding

Real-Time Bidding (RTB) is a form of online advertising which provides buying and selling of online ad impressions to a target audience, through real-time auctions that are conducted while the web page is loading (e.g., between 200 ms and 300 ms (Kumar, 2017)).

Figure 2 shows the typical RTB process (Yuan et al., 2014), which starts when advertiser asks a DSP to run an ad campaign for a product based on budget, target audience and campaign duration. (1) Once a user visits a web page, (2) the AdX generates a bid request for each ad request and (3) sends it to all eligible DSPs, which takes into account their clients interests and user information provided by the Data Management Platform (DMP), and then calculates bid prices based on their campaigns, and sends bid response contains a bid price or no-bid reason for each bid request. (4) Then, the AdX collects bid responses received within a t_{max} specified by the publisher: all bid responses sent after this timeout are rejected. Finally, (5) the AdX determines the outcome of the auction and chooses the DSP with the best bid price to (6) display its ad to the publisher who (7) sends it to the user. The winner pays the second-highest price since RTB employs Vickrey auction semantics.

In our work, we focus on step (3) by modifying how the AdX behaves. Instead of broadcasting the bid request to all DSPs, it will performing a filtering action, using publish/subscribe semantics, to selectively decide which DSPs to solicit a bid response from.

2.3 Vickrey Auction

Vickrey auction is a type of sealed bid auction, where bidders submit their bids for something without knowing other participants bids. Then, the highest bidder wins but pays the second highest-bid, so it encourages everyone to bid truthly. In a Vickrey auction, the auctioneer sets a reservation price (Kalra et al., 2019), which is a minimum price below which the item is not sold at all.

Theorem 1. *Let {b1 .... bn} be a set of bidders and R the reserve price:*

- *if* $\exists\, b_i > \max_{j \neq i} b_j \Rightarrow b_i$ *is the highest bidder and pays the second highest price.*
- *if* $b_i = \max_{j \neq i} b_j$ *(which means there are two highest bids)* $\Rightarrow$ *the winner is selected randomly from the highest bidders, or alphabetically or by checking who submitted their bid first or according to tie-breaking rule and pays his bid price.*
- *if* $max_{j \neq i} b_j \leq R \Rightarrow$ *the winner pays R.*

2.4 OpenRTB Specifications

OpenRTB is a project developed by IAB Technology Laboratory to specify open communication protocols and standards between buyers and sellers of ad impressions in the context of RTB advertising (IAB, 2016). OpenRTB provides an API with all essential entities (RTB Exchange and bidder) and interactions between them (sending bid request, bid response, win billing and loss notices). The RTB Exchange (AdX) sends a bid request to the bidder (DSP) with details about the site, content, user, device, location, etc. The bidder returns bid response with a bid price if it decides to participate in the auction or no–bid if not. A possible response to the request is *no-bid*, which is accompanied by a reason. After launching auction, AdX sends win notice to the winner and loss notice for other bidders.

In this paper, we analyse the different reasons and identify which ones could be avoided by integrating a publish/subscribe system to filter out DSPs using information gathered from the bid requests and bid responses.

2.5 iPinYou Dataset

iPinYou dataset is a set of auction, impression, click and conversion logs extracted from real advertising campaigns obtained from the iPinYou DSP platform. To the best of our knowledge, this is the only publicly available dataset containing RTB auctions. The original objective of the iPinYou dataset is to evaluate DSP bidding algorithms submitted for a competition in 2013 (Liao et al., 2014). In this paper, we are only interested in the auction logs, which contains information of bid requests and DSP responses (Zhang and Zhang, 2014).

As we will see in Section 4.3, our top-k scoring relies of the prediction of bid values and response times, as described in Section 4.4. In order to enable this prediction, we extract features from the OpenRTB specifications, and train our model using the corresponding attributes in the iPinYou dataset. Furthermore, since the accuracy of the model is highly dependent on the workload, we analyze the iPinYou dataset in detail and derive the statistical distribution of important attributes (using SAS JMP[1]). This analysis is used by our data generator and described further in Section 6.

3 RELATED WORKS

In this section, we survey related works in the literature, divided into three categories: bidding strategies, header bidding and publish/subscribe.

3.1 Bidding Strategies

As of today, RTB research remains limited given the relative size of its market (Yuan et al., 2014). Most of the research in this area is directed towards the optimization of DSP algorithms to calculate bid price. In (Wang et al., 2016), the authors estimate the probability of ad clicks and conversions using linear regression models (logistic regression and Bayesian Probit regression) and nonlinear models (factorisation machines, gradient tree models, and deep learning), then use this metric to optimize bidding. In (Lee et al., 2013), the authors suggest an algorithm which select high quality impressions and adjust the bid price based on historical performance, while spreading the ad campaign budget optimally across time. The algorithm is based on the estimation of click-through rate (CTR) and action or conversion rate (AR). In (Zhang and Zhang, 2014), the authors integrates the concept of budget utilization efficiency in DSP bidding strategy, in order to win as many impressions as possible. Another method proposed for bidding strategies consists in predicting the winning bid price (Wu et al., 2015). So, these approaches are complementary to our own work, as we seek to reduce the communication overhead of conducting auctions by optimizing how the AdX selects DSPs. Once selected, the DSPs may then leverage the aforementioned works to decide its own bid. To the best of our knowledge, the bidding strategies found in literature do not depend of previous auctions' results: DSPs do not use the complete auction history to calculate bid prices. Therefore, selectively filtering DSPs and omitting them from certain auctions will not affect the correctness of the bidding algorithms.

These previous works may potentially reduce the processing time required by each DSP, and thus allow the AdX to set a shorter timeout (t_{max}). Our work is complementary to these bidding strategies since it also selects a subset of DSPs to contact, thereby reducing communication overhead. In addition, we implement some of these bidding strategies in our evaluation in order to generate a realistic load for bid responses (cf. Section 6).

3.2 Header Bidding

Header bidding or pre-bidding is a technique for online advertising that allows publishers to offer an ad impression to several ad exchanges before launching a RTB auction. Each AdX proposes a price to the publisher, who chooses the highest offer and sends the ad impression to the winning AdX (Qin et al., 2017), who then internally forwards it to the winning DSP. This process helps the publishers dynamically choose the most suitable AdX for each ad impression instead of committing to reservation contracts with each of them (Sayedi, 2018).

Header bidding is complementary to our solution: In order to propose a price, an AdX must conduct an auction for its own DSPs. Thus, the AdX can employ our solution to dynamically select the set of DSPs to contact for such a RTB auction.

3.3 Publish/Subscribe

Several works optimize pub/sub systems such as:

Aggregation: The authors propose grouping publications within the same time window and sending a summary to reduce traffic (Jacobsen et al., 2014). This approach does not work for RTB because bid requests cannot be aggregated.

[1] https://www.jmp.com/en_ca/home.html

Parametric Subscriptions: The solution proposed allows the pub/sub system to autonomously adapt to the dynamic interests of the subscribers, alleviating the need to unsubscribe and resubscribe repeatedly (Jayaram et al., 2010). In our approach, the main bottleneck is the publication traffic, and not the subscription traffic.

Top-k Filtering: Top-k filtering is a commonly used technique to effectively reduce the volume of publications to be disseminated by a pub/sub system while maintaining high data relevance. In (Shraer et al., 2014), the authors maintain top-k tweets for each social story at a news website serving high volume of page views using a publish/subscribe system. The tweets are ordered based on a content/recency score function. Top-k subscriptions is defined in (Zhang et al., 2017) to deliver a publication only to the k best ranked subscribers using rank-covering for large-scale applications. Top-k filtering is also used in (Zhang et al., 2013) to reduce the amount of notifications sent in social networks. In (Chen et al., 2015), the authors propose a new solution to handle a large number of temporal spatial-keyword (TaSK) top-k requests on a geo-text object stream.

In our work, we use top-k as a method to select the most relevant subscribers for a given publication, which is close to the model of top-k subscriptions presented in (Zhang et al., 2013). To the best of our knowledge, top-k filtering was never used in the context of RTB, and our solution is adapted and optimized specifically according to RTB semantics.

4 PROPOSED SOLUTION

In this section, we first present our new solution which integrates pub/sub with OpenRTB, in order to eliminate no-bid responses. Then, we investigate how to further reduce the number of irrelevant bid responses by formulating the problem definition of minimum bid responses selection. We show how our pub/sub solution can be extended using top-k filtering to filter out bid responses that are too slow or with low values. Finally, we adapt the top-k solution by replacing the score prediction component with a discrete machine learning model.

4.1 Content-based Filtering of DSPs

Sending bid requests to DSPs who respond with a no-bid message has two major consequences. First, it generates communication overhead since 2 messages need to be exchanged between the AdX and the DSP: the request and the response. Second, the timeout value t_{max} is potentially affected if the DSP is slow to produce the no-bid response, thus increasing the end-to-end latency for that bid request.

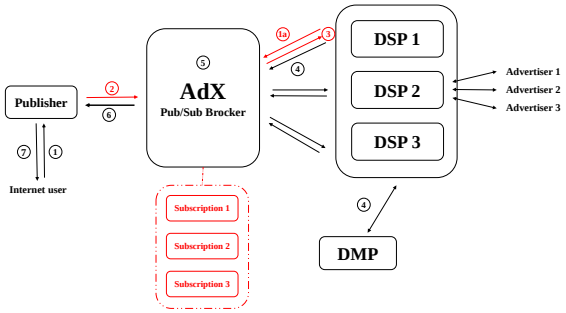


Figure 3: Proposed Content-Based Solution for Selective Auctioning.

Our analysis of the OpenRTB specifications, as well as the iPinYou dataset, reveals that most no-bid responses are related to the content of the bid request, such as unsupported device or unmatched user (error codes 6 and 8). Therefore, our solution allows the DSPs to expose any constraint on they may have regarding these two aspects in the form of a content-based subscription, thereby filtering DSPs at the AdX side and eliminating no-bid responses with these error codes. Our proposed RTB processing model is shown in Figure 3. This is accomplished by leveraging the attributes *device* and *user* from the OpenRTB specifications. They are also called *User-Agent* and *User-ProfileIDs* in the iPinYou dataset, respectively. Compared to the original model in Figure 2, the following steps have been modified:

(1a) Prior to receiving auctions, DSPs send to the AdX (equipped with a publish/subscribe broker) the following subscription:

$$Subscription : (device, =, val1), (user, =, val2)$$

where *device* is the desired user device (mobile, desktop computer, set top box, etc.), and *user* refers to characteristics of the target audience (keywords, interests, gender, etc.).

(2) Once a user visits a page web, the Publisher delivers to the AdX an ad request containing the publication:

$$Publication : (device, val1), (user, val2)$$

(3) For each inbound ad request, AdX generates a bid request as follows:

$$Bidrequest : (id, site, device, user)$$

The AdX matches the bid request publication against stored subscriptions and forwards it to interested DSPs, which calculates bid prices based on their campaigns. DSPs with unmatched subscriptions will not receive the bid request.

Note that the publish/subscribe system allows each DSP to send multiple subscriptions if necessary. Furthermore, subscriptions can be modified at any time to reflect updated interests from the DSP.

4.2 Optimal Number of Bid Responses

Now we focus on the problem of filtering DSPs beyond those who are not interested. Intuitively, the AdX should try to obtain a high winning price for each auction, while waiting just long enough to collect this winning price (which is the second-highest bid in a Vickrey auction). Reducing auction time is very important since the longer an auction takes, the longer resources (e.g., state, RAM) are blocked, and the more resources are required for achieving the same throughput which may lead to a bottleneck. Also, reducing the number of contacted DSPs optimizes the RTB system when it is overloaded with bid requests. Thus, DSPs with either low bid values or slow responses should also be filtered by our proposed solution. We define a general model that capture these two parameters using a cost model. We formalize our optimization problem which select a minimum number of auction responses while keeping a second best high price (for Vickrey auctions) and a minimum waiting time, as follows:

Given a set of bid responses $BR = \{(b_1,t_1),(b_2,t_2),...,(b_n,t_n)\}$ *where n is the number of DSPs,* t_i *is response time,* b_i *bid price of DSP*$_i$ *and* BR^c *is a set of chosen bid responses*:

$$\textbf{Minimize } |BR^c|, \textit{ where } BR^c \subseteq BR$$
$$\textbf{Subject to } U(BR^c) \geq U(BR^d), \forall BR^d \subseteq BR$$
$$U(BR_i) = w_1 \times second\_price(BR_i) + w_2 \times (t_{max} - max\_time(BR_i))$$
$$second\_price(BR_i) = max_{j \neq i} b_j, \textit{ where } b_i > max_{j \neq i} b_j, \forall (b_i,t_i),(b_j,t_j) \in BR_i$$
$$max\_time(BR_i) = max(t_i), \forall (b_i,t_i) \in BR_i$$
$$w_1 + w_2 = 1$$

In the above formula, $U(BR)$ represents the utility of a subset of bid responses, which is calculated by factoring in the two objective metrics: paid price (second-highest price) and auction time (time of the slowest response received). These two metrics are normalized using weight parameters w_1 and w_2, which can be adjusted depending on the application. The optimal set of chosen DSP is the one that yield the greatest utility, among all possible subsets of DSPs in the system, with the fewest number of DSPs selected.

Given an oracle with perfect knowledge, which can accurately return the minimum bid responses with best bid prices and time taken by each DSP prior to the auction, we can prove that the optimal solution has always size two:

Theorem 2. *The number of chosen DSPs is always 2 (*$|BR^c| = 2$*).*

Proof. (By contradiction) Suppose that the optimal set of chosen bid responses is $BR^c = (b_i,t_i),(b_s,t_s),(b_w,t_w)$. Thus, $|BR^c| = 3$ and $U((b_i,t_i),(b_s,t_s),(b_w,t_w))$ is the maximum among all possible sets of chosen bid responses.

Case 1: Without loss of generality, if $b_i < second\_price(BR^c)$, then $b_i < b_s < b_w$ and $t_i \leq max-time(BR^c) \Rightarrow (b_i,t_i)$ can be removed without affecting utility, since it does not contribute to raising the winning price (second highest price), nor can it increase the response time of remaining DSPs in the set, since each response time is independent.
Case 2: $\forall b, b \geq second\_price(BR^c)$, then $\exists b_s$ and $b_w \geq second\_price \Rightarrow (bi,ti)$ can be also removed, since it means at least 2 of the 3 bids have the same value, so removing one of the three will not affect the value of the second price. Again, removing a bid cannot increase the response time of remaining DSPs.

In both cases, $U((bi,ti),(bs,ts),(bw,tw)) = U((bs,ts),(bw,tw))$, then $|BR^c| = |BR^c - (bi,ti)| = 2 < 3$. □

Therefore, our problem can be represented as a selection of 2 bid responses from n elements, where the utility of the chosen subset is maximal among all subsets of size 2. Thus, it is a combination problem with $\binom{n}{2}$. This combination can be solved in quadratic time $O(n^2)$ (Oppen, 1980).

4.3 Top-k Filtering

In the previous section, we demonstrate that the AdX should optimally select 2 DSPs for every bid request. In practice, this is not feasible without perfect knowledge of future bid values and response times. Therefore, we propose the use of top-k filtering as a general solution to select a subset of DSPs using a fixed size k, according to a scoring function which models the utility function of the previous section. In ideal circumstances, our top-k solution would yield optimal results if $k = 2$ and the top-k scoring function corresponds perfectly to the utility function $U(BR)$.

Figure 4 shows our proposed solution with pub/sub and top-k filtering. For each arriving bid request b, the AdX compares the subscriptions of n DSPs to obtain $m \leq n$ DSPs matching publication b, and then selecting $k \leq m$ DSPs with the highest score to participate in the auction.

Figure 5 shows how the top-k filtering works in details. The performance of each DSP is predicted in order to calculate their scores. They are then sorted

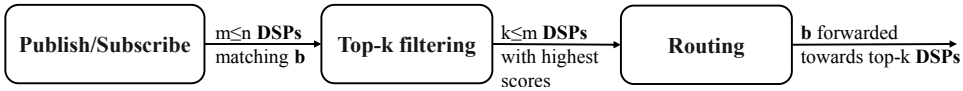


Figure 4: Processing Flow with Pub/Sub and Top-K Filtering.

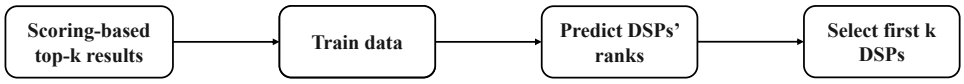


Figure 5: Scoring-based Top-K Overview.

in order to select the highest k scores. The score of a DSP_i for a given request b is calculated with the following formula:

$$score(DSP_i,b) = w_1 \times p(DSP_i,b) - w_2 \times won\_bids(DSP_i) \quad (1)$$

with $p(DSP_i,b)$ is predicted performance of DSP calculated as follows:

$$perf(DSP_i,b) = \frac{predicted\_price(DSP_i,b)}{predicted\_time(DSP_i,b)} \quad (2)$$

The above formula is similar to the theoretical formulation as it jointly considers both metrics of price and time. To introduce some fairness in the system, the score substracts *wonBids*, which indicates the number of bids already won by the DSP_i. This reduces the likelihood of a DSP to always be selected and denying the opportunity for others to bid. Finally, w_1 and w_2 are adjustable parameters used to tune the weight of each component of the score.

4.4 Score Prediction Model

To predict the performance of DSPs for any given request, we need to estimate the bidding prices and response times using its historical training data (e.g., iPinYou dataset), which consists of past bid requests (with detailed information about their features) and the corresponding responses from DSPs, including their bid prices and response times. We propose the use of regression analysis to predict price and response time as two continuous variables in $\mathbb{R}_{\geq 0}$.

First, we select features of the iPinYou dataset to use for regression, using the Filter method (KAUSHIK,). We build a correlation matrix of features found in the iPinYou dataset. Then, we remove every feature that is highly correlated with another feature (correlation coefficient $|c| > 0,5$) other than bidding price and response time. We obtain the following remaining features: Timestamp, AdExchange, AdSlotHeight, CityID, AdSlotVisibility, AdSlotFloorPrice and UserProfileIDs.

To predict bid price and response time, we compare three popular regression methods: Linear regression, Regression tree and K-Nearest Neighbors (for regression). For each DSP, we apply these algorithms the dataset and compare the results of each algorithms using the following metrics (Moayedi et al., 2019):

Mean Absolute Error (MAE): The average of the differences between the predictions and the actual values. 0 is a perfect fit.

R^2: An indication of goodness of fit of a set of forecasts compared to actual values, ranging from 0 to 1.

According to regression metrics results, we note that linear regression performs the best for the bid price with $MAE \simeq -3.50$ and $R^2 \simeq 0.88$. We also find that KNN regression works the best for the response time with $MAE \simeq -6.54$ and $R^2 \simeq -13.57$. Therefore, we will implement these two techniques in our evaluation.

5 DISCRETE PREDICTION MODELS

The proposed top-k solution requires the prediction of two continuous variables: bid price and response time. As we will see in the evaluation (Section 6), the regression analysis is slow to calculate predictions, which negatively impacts the end-to-end latency of running each auction. In order to speed up the prediction, we propose the use of discrete models.

The main insight is that the top-k filtering purpose is solely to determine which DSPs to contact in order to run a profitable and short auction: it does not need to accurately predict the winning bid or total time taken for the auction. Therefore, the current scoring mechanism is calculating more than needed, which degrades performance. We propose two prediction models using discrete variables: rank-based and binary selection-based. In order to use these models, we must adapt the original top-k filtering process, which will be described in each subsection.

5.1 Rank-based Top-k

This method is based on predicting the rank of each interested DSP by the current bid request (6) (cf. Figure 6). First, we use the same historical data as before,

except we calculate the score of each DSP based on their bid value and response time, and use the scores to establish the ranking of the DSPs for each bid request. Then, we train a linear regression algorithm to predict the rank of each DSP given the features of the bid request. Finally, the AdX selects k DSPs with the highest rank.

5.2 Binary Selection-based Top-k

While the previous solution uses a discrete variable, it still is performing more work than necessary since it tries to accurately predict the rank of each DSP. Our next approach is to classify each DSP in two categories: "yes" or "no", answering solely the question whether a DSP should be contacted or not for a given RTB auction. We accomplish this using classification analysis (cf. Figure 7). We tested five types of classifiers: FeedForward, Bayesian, NEAT, PNN and RBF. We opted to use FeedForward for its superior execution time compared to others. To train this model, we use the same dataset, but convert the score into a "yes" or "no" value based on whether this DSP belongs to the top-k for an auction or not. Note that this method does not guarantee top-k results with exactly k DSPs. Furthermore, the training is done for a specific value of k chosen in advance.

6 EVALUATION

In this section, we experimentally evaluate our solutions: Pub/Sub, and Top-k against a baseline implementation of OpenRTB. Our experiments contains a performance comparison of the three approaches with respect to auction time and paid price. Then, we conduct a sensitivity analysis of the major parameters impacting the solutions.

6.1 Experimental Setup

Implementation: Our work is based on the OpenRTB 2.0 reference implementation[2]. We implemented in J2EE our own AdX according to the OpenRTB API specifications (version 2.5) (IAB, 2016). We use RabbitMQ using the Header Exchanges as the pub/sub broker (Ionescu, 2015). Finally, we add the top-k engine to the AdX and integrate it with RabbitMQ.

Bidding Strategies: We employ 100 DSPs distributed across three different types of bidding strategies:

- CONSTANT
- RANDOM
- Below_eCPC: (Zhang and Zhang, 2014) where the offer price is calculated by multiplying the maximum of eCPCs (effective cost per click for each campaign) with the CTR (click-through rate) predicted for the ad impression.

Default Parameters: We set the selectivity to 70% of interested DSPs per bid request. Our default top-k method is the scoring-based one (cf. Section 4.3), with $k = 10$.

Workload: By default, we employ the iPinYou dataset, in particular the logs containing auction data (bid requests and responses). For the sensitivity analysis, we developed a dataset generator using iPinYou datset distributions with adjustable parameters. We generate 100 bid requests at a time.

Environment: We conduct our evaluation using an Ubuntu virtual machine version 18.04.3 LTS, with 4096 MB of RAM and 2 GHz CPU.

6.2 Performance Comparison

Using four different metrics, we compare our three approaches: OpenRTB (baseline), Pub/Sub only, and Pub/Sub with Top-k.

Number of Messages: Figure 8 shows the number of bid responses sent by DSPs and the rate of no-bids responses. Compared to the baseline, the Pub/Sub and Top-k reduce the total number of messages by 47% and 90%, respectively. In particular, 4696 no-bid responses are received by the AdX, which are completely eliminated by both of our solutions. The top-k solution can further filter out an additional 4304 messages compared to only pub/sub, which are low-scoring bid responses. Furthermore, the top-k filtering increases stability, since exactly 10 responses are received for each of the 100 bid requests.

Auction Time: Figure 9a shows the cumulative distribution function (CDF) for the end-to-end auction time of bid requests processed by the three approaches. The auction time is measured from the moment the AdX receives the bid request from the publisher to the moment it contacts the winning DSP. In the OpenRTB baseline implementation, auctions takes between $1800ms$ to $3000ms$, compared to $800 - 2,000ms$ for pub/sub, and $120 - 600ms$ for top-k. The median for each solution is $2510ms$, $1171ms$ and $140ms$ for baseline, pub/sub, and top-k respectively. Our pub/sub and top-k solutions are therefore able to reduce the auction time by 50% and 94%. This result demonstrates that a reduction in the number of DSPs contacted by the AdX decreases the chance of

[2]https://github.com/openrtb/openrtb2x

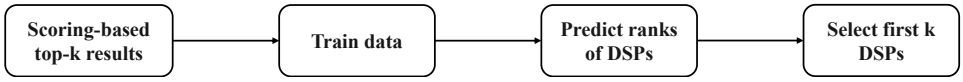


Figure 6: Rank-based Top-K Overview.

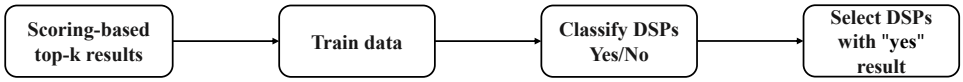


Figure 7: Binary Selection-Based Top-K Overview.

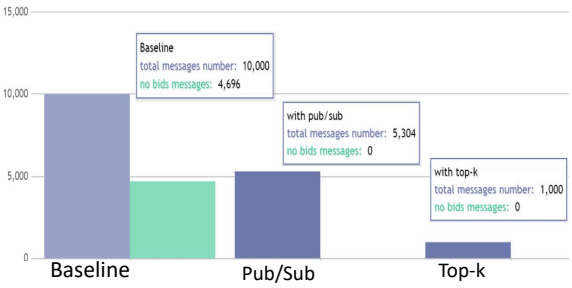


Figure 8: Baseline Comparison for the Number of Bid Responses.

encountering a slow DSP (straggler), thereby reducing the overall time.

Paid Price: Figure 9b shows the CDF for the paid price using the same bid requests for all three approaches. Here, the baseline OpenRTB is optimal since it contacts every DSP every time, guaranteeing the maximum price of each auction. The pub/sub solution also performs optimally, since it only eliminates no-bid responses which no chance of winning. For the top-k solution, a median loss of 3.61% is incurred for two reasons. First, a DSP with high bid price but slow response time might not be selected as it will score poorly for top-k. Second, the prediction model used might incorrectly underestimate the score of a DSP, causing it not to be selected when it would have impacted the paid price (i.e., submitted a bid higher than the second highest price).

Efficiency: Figure 9c evaluates the efficiency of the three solutions. This metric combines the previous two metrics by dividing the paid price with the auction time. Since the pub/sub solution yields the same paid prices as OpenRTB with shorter auction times, its efficiency is superior with a median of $\simeq 0,19$ yuan/ms against only $\simeq 0,08$ yuan/ms, a 2.375 times increase. However, top-k is even better with a median efficiency of $\simeq 1.48$ yuan/ms, which is 18.5 times better than the baseline and 7.79 times better than the pub/sub only solution. Although top-k sacrifices a small loss in paid prices, its execution time is also substantially shorter. The trade-off is therefore in favor of the top-k solution.

Summary: Compared to the baseline, the pub/sub and top-k approaches reduce the number of messages by 47% and 90%, and the auction time by 50% and 94%, respectively. The pub/sub only approaches does not compromise on the paid price, while the top-k approach suffers a loss of 3.61% in paid prices. However, this loss is offset by a huge speedup in auction time, as evidenced by the superior efficiency of the top-k approach. The top-k approach is therefore desirable if the application receives a near-infinite stream of requests that must be treated efficiently. On the other hand, if the volume of bid requests is limited, the pub/sub only approach is desirable in order to extract the maximum price per auction.

6.3 Sensitivity Analysis

We conduct a sensitivity analysis for our solutions in order to study the impact of three parameters: selectivity of bid requests, value of k, and choice of prediction model.

Selectivity: Figure 10a shows the impact of varying the selectivity of DSPs. This parameter controls how many DSPs are interested in each bid request. We test 3 scenarios: 30%, 50% and 70%. The baseline solution is unaffected by this parameter, since it contacts every DSP without considering their interests. The top-k solution is also insensitive to selectivity: if k is sufficiently low, the change in sensitivity does not impact the number of DSPs selected, which is always k per bid request. Finally, the pub/sub only approach is affected the selectivity: the median are 999ms, 1083ms, and 1171ms for 30%, 50%, and 70%. As the selectivity increases, the filtering performed by the pub/sub broker decreases, which reduces the effectiveness of the solution.

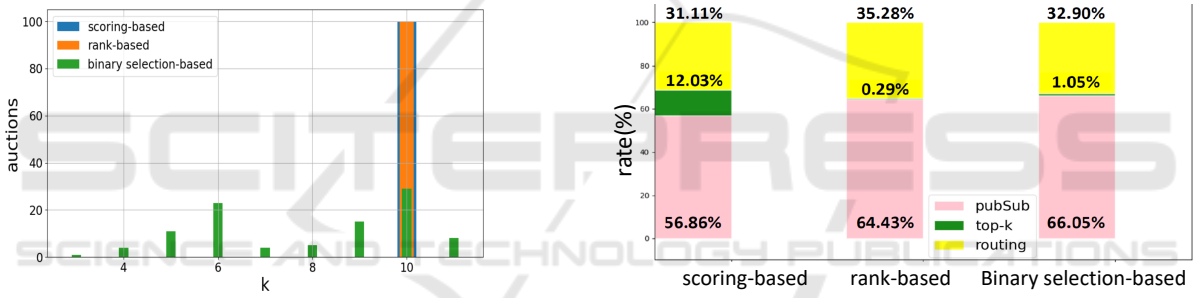
Parameter k: We test 4 different values of k: 2, 4, 8, and 10. Figure 10b shows the paid price CDF for each scenario, compared to the optimal price represented by the OpenRTB baseline. As expected, the paid price generally decreases when k decreases, since the margin of error becomes narrower for the prediction model to correctly identify the ideal DSPs for the auction. However, the difference is not substantial, with a median loss of 4.1% for $k = 2$ compared to 3.61% for $k = 10$. This indicates that the regression model used is highly accurate.

(a) Auction Time. (b) Paid Price. (c) Efficiency.

Figure 9: Baseline Comparison between OpenRTB, Pub/Sub, and Scoring-Based Top-K.

(a) Selectivity vs. Auction Time. (b) k vs. Paid Price. (c) k vs. Auction Time.

Figure 10: Sensitivity Analysis of the Scoring-Based Top-K Solution.



(a) Number of Selected DSPs for Top-K Solutions. (b) Processing time for Top-K.

Figure 11: Comparison of Top-K Solutions.

For the auction time, Figure 10c shows that a noticeable improvement can be achieved by reducing the value of k. The median value for $k = 2$ is $32ms$ compared to $140ms$ to $k = 10$.

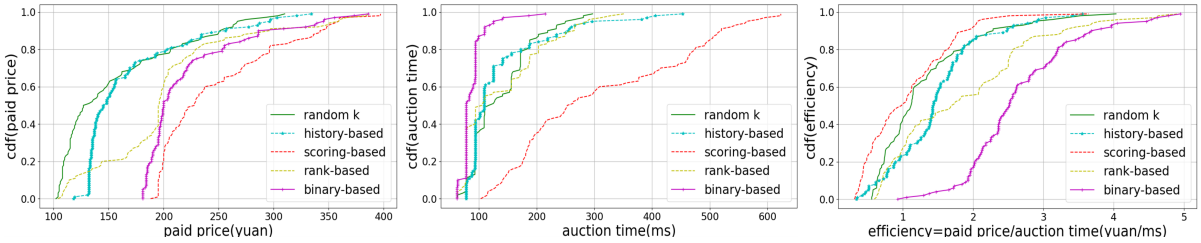
Prediction Models: We compare our three prediction models: scoring-based, rank-based, and binary selection-based, with two baselines: history-based (average of past auctions) and random k (choose k DSPs at random).

Figure 12a shows the CDF for paid prices. Both baselines (history and random) are clearly inferior to our proposed solution, with intervals of $120-340$ yuan and $100-310$ yuan. One notable exception is the rank-based one, which obtained poor prices for 20% of the auctions and thus has a wide interval of 100-390 yuan. On the other hand, the paid prices for scoring-based and binary selection-based are in the range of $180-390$ yuan, which is near optimal to the OpenRTB baseline.

Figure 12b shows the CDF for auction time. Random-k and history-based have an interval range of $40-300ms$ and $70-450ms$. The scoring-based model is the heaviest to compute, with an interval of $100-600ms$. The rank-based model has an interval of $40-330ms$ and the binary selection-based top-k outperforms all others with an interval of $40-220ms$.

We compare also the efficiency (paid price/auction time) in Figure 12c which clearly demonstrates that the binary selection-based is the most efficient model with a range of $0.9-5$ yuan/ms compared to other models that are almost in the same interval $0.5-4$ yuan/ms.

Since the binary selection-based model does not guaranteed a fixed k per bid request, we also measure the variation in the size of the sets returned in Figure 11a. The results show that the set vary between 3 to 11 DSPs selected. For rank-based and scoring-based, the results confirm that the number of DSPs



(a) Top-K Model vs. Paid Price. (b) Top-K Model vs. Auction Time. (c) Top-K Model vs. Efficiency.

Figure 12: Comparison of Top-K Models.

Table 1: Summary of Top-k Filtering Solutions.

Top-K Models	Advantages	Disadvantages
Scoring-based	Near-optimal prices; flexible choice of k	Poor execution time; loss of efficiency
Rank-based	Flexible choice of k	Uneven performance
Binary selection-based	Best execution time and efficiency	Variable size of results; the value of k is fixed during training

selected is always k (10).

Figure 11b shows a decomposition of the end-to-end latency into three parts: pub/sub filtering, top-k ranking, and the routing. The main difference between our various solutions is the time taken to run the top-k model: it is almost 12%, 0.3% and 1% of the overall time in scoring, rank-based and binary selection-based models respectively. This result confirms that the regression analysis used by the scoring-based solution is slow due to its usage of continuous variables. The rank-based and binary selection-based models have similar top-k processing times since they are both discrete. While the rank-based model is slightly faster in regards to the top-k processing time than the binary selection-based one, the overall auction time is better with the latter (as seen in Figure 12b), because it contacts fewer DSPs on average (see Figure11a) which reduces the routing time.

Summary: When compared to the baseline and the pub/sub solution, the top-k solution stands out as being the most reliable, since it is not sensitive to the selectivity of the bid requests. For the scoring-based solution, k can be set to a surprisingly low number (as low as 2), with little loss of price and substantial speedup in auction time. When comparing prediction models for top-k solutions, each of three proposed approaches have advantages and disadvantages, as highlighted in Table 1. For maximum efficiency, the binary-based selection model stands out as it is noticeably faster with minimal price loss, with the drawback of returning uneven-sized results and being inflexible. The scoring-based model has the worst efficiency, but can obtain better prices than other solutions. The rank-based solution is a compromise between the two, as it has average efficiency and retains flexibility. However, it suffers from uneven performance for a minority of requests (20%).

7 CONCLUSIONS

The current standard for real-time bidding (RTB) broadcasts bid requests to all DSP bidders, which generates massive communication and computation overhead. We propose the use of publish/subscribe to express the interests of DSP bidders as content-based subscriptions, in order to eliminate no-bid responses through auctioning with selected DSPs only.

Furthermore, we explore how to further reduce the number of DSPs contacted by avoiding bids with low prices or slow response times. We formulate the problem of optimal number of bid responses, and demonstrate how top-k filtering can be used to address this problem in a online setting. Top-k filtering relies on a prediction model in order to guess which DSPs should participate in which auction. Our scoring-based model uses regression analysis with continuous variables, while our rank-based and binary selection-based models both employ discrete analysis.

Our evaluation confirms the effectiveness of our approach in reducing the number of messages required and the auction time, while maintaining optimal or near-optimal winning prices. Our most effective solution is the OpenRTB implementation using Pub/Sub with a binary selection-based top-k filtering mechanism.

For our future work, we wish to test the applicability of our approach across a wider range of datasets beyond iPinYou, and to implement more sophisticated

bidding strategies which could challenge the accuracy of our prediction models. We will also investigate making our solution self-adaptive, tuning its parameters (e.g., value of k) by monitoring the current conditions.

REFERENCES

Canas, C., Zhang, K., Kemme, B., Kienzle, J., and Jacobsen, H. A. (2017). Self-Evolving Subscriptions for Content-Based Publish/Subscribe Systems. *International Conference on Distributed Computing Systems*.

Chen, L., Cong, G., Cao, X., and Tan, K. L. (2015). Temporal Spatial-Keyword Top-k publish/subscribe. *International Conference on Data Engineering*.

Eugster, P. T., Felber, P. A., Guerraoui, R., and Kermarrec, A.-M. (2003). The many faces of publish/subscribe. *ACM Computing Surveys*.

IAB (2016). Openrtb api specification version 2.5. Technical report.

Ionescu, V. M. (2015). The analysis of the performance of RabbitMQ and ActiveMQ. *14th RoEduNet International Conference - Networking in Education and Research*.

Jacobsen, H.-A., Pandey, N. K., Vitenberg, R., Zhang, K., and Weiss, S. (2014). Distributed event aggregation for content-based publish/subscribe systems. *8th ACM International Conference on Distributed Event-Based Systems*.

Jayaram, K. R., Jayalath, C., and Eugster, P. (2010). Parametric subscriptions for content-based publish/subscribe networks. In *Middleware 2010*. Springer Berlin Heidelberg.

Kalra, A., Borcea, C., Wang, C., and Chen, Y. (2019). Reserve price failure rate prediction with header bidding in display advertising. *the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

KAUSHIK, S. Introduction to feature selection methods with an example (or how to select the right variables?). Available at https://www.analyticsvidhya.com/.

Kumar, J. (2017). Timeout Analysis, Troubleshooting and Notification in Real Time Bidding Advertising System with Implementation. *Computer Science and Engineering*.

Lee, K. C., Jalali, A., and Dasdan, A. (2013). Real time bid optimization with smooth budget delivery in online advertising. *the 7th International Workshop on Data Mining for Online Advertising, ADKDD 2013 - Held in Conjunction with SIGKDD 2013*.

Liao, H., Peng, L., Liu, Z., and Shen, X. (2014). iPinYou Global RTB Bidding Algorithm Competition Dataset. *20th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Moayedi, H., Bui, D. T., Dounis, A., and Lyu, Z. (2019). applied sciences Predicting Heating Load in Energy-E ffi cient Buildings Through Machine Learning Techniques.

Mullarkey, M. T. and Hevner, A. R. (2015). Entering action design research. In *New Horizons in Design Science: Broadening the Research Agenda*.

Oppen, D. C. (1980). Complexity, convexity and combinations of theories. *Theoretical Computer Science*.

Qin, R., Yuan, Y., and Wang, F. Y. (2017). Optimizing the revenue for ad exchanges in header bidding advertising markets. *2017 IEEE International Conference on Systems, Man, and Cybernetics*.

Sayedi, A. (2018). Real-time bidding in online display advertising. *Marketing Science*.

Shraer, A., Gurevich, M., Fontoura, M., and Josifovski, V. (2014). Top-k publish-subscribe for social annotation of news. *the VLDB Endowment*.

Wang, J., Zhang, W., and Yuan, S. (2016). Display Advertising with Real-Time Bidding (RTB) and Behavioural Targeting. *Foundations and Trends® in Information Retrieval*.

Wu, W. C. H., Yeh, M. Y., and Chen, M. S. (2015). Predicting winning price in real time bidding with censored data. *the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Yuan, Y., Wang, F., Li, J., and Qin, R. (2014). A survey on real time bidding advertising. *IEEE International Conference on Service Operations and Logistics, and Informatics*.

Zhang, C. R. and Zhang, E. (2014). Optimized bidding algorithm of real time bidding in online ads auction. *International Conference on Management Science and Engineering*.

Zhang, K., Sadoghi, M., Muthusamy, V., and Jacobsen, H. A. (2013). Distributed ranked data dissemination in social networks. *International Conference on Distributed Computing Systems*.

Zhang, K., Sadoghi, M., Muthusamy, V., and Jacobsen, H.-A. (2017). Efficient covering for top-k filtering in content-based publish/subscribe systems. *Middleware '17*.