

Deep Landscape Forecasting for Real-time Bidding Advertising

Kan Ren, Jiarui Qin, Lei Zheng, Zhengyu Yang, Weinan Zhang, Yong Yu
Shanghai Jiao Tong University
{kren, qinjr, zhenglei, zyyang, wnzhang, yyu}@apex.sjtu.edu.cn

ABSTRACT

The emergence of real-time auction in online advertising has drawn huge attention of modeling the market competition, i.e., *bid landscape forecasting*. The problem is formulated as to forecast the probability distribution of market price for each ad auction. With the consideration of the censorship issue which is caused by the second-price auction mechanism, many researchers have devoted their efforts on bid landscape forecasting by incorporating survival analysis from medical research field. However, most existing solutions mainly focus on either counting-based statistics of the segmented sample clusters, or learning a parameterized model based on some heuristic assumptions of distribution forms. Moreover, they neither consider the sequential patterns of the feature over the price space. In order to capture more sophisticated yet flexible patterns at fine-grained level of the data, we propose a Deep Landscape Forecasting (DLF) model which combines deep learning for probability distribution forecasting and survival analysis for censorship handling. Specifically, we utilize a recurrent neural network to flexibly model the *conditional* winning probability w.r.t. each bid price. Then we conduct the bid landscape forecasting through probability chain rule with strict mathematical derivations. And, in an end-to-end manner, we optimize the model by minimizing two negative likelihood losses with comprehensive motivations. Without any specific assumption for the distribution form of bid landscape, our model shows great advantages over previous works on fitting various sophisticated market price distributions. In the experiments over two large-scale real-world datasets, our model significantly outperforms the state-of-the-art solutions under various metrics.

KEYWORDS

Bid Landscape Forecasting; Real-time Bidding; Market Price Modeling; Winning Probability Estimation; Deep Learning; Survival Analysis

ACM Reference Format:

Kan Ren, Jiarui Qin, Lei Zheng, Zhengyu Yang, Weinan Zhang, Yong Yu. 2019. Deep Landscape Forecasting for Real-time Bidding Advertising. In *The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '19)*, August 4–8, 2019, Anchorage, AK, USA. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3292500.3330870>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions.acm.org.

KDD '19, August 4–8, 2019, Anchorage, AK, USA

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-6201-6/19/08...\$15.00

<https://doi.org/10.1145/3292500.3330870>

1 INTRODUCTION

Emerged from 2009 [22], real-time bidding (RTB) has become one of the most important media buying mechanisms in online advertising. In RTB, the advertisers propose the bid price in real time, according to their own bidding strategies and the auction side information [39], then the ad exchange decides the winner in the market of the auction, i.e., the advertiser with the highest bidding price in this auction [36]. From the view of advertiser, specifically, the bid price is decided according to the estimated *utility* [27] and the *cost* of the given auction request [26, 27, 32]. On one hand, the utility generally represents the positive user response probability such as click-through rate (CTR) or conversion rate (CVR). On the other hand, the cost is the price which the advertiser would probably pay for the given auction.

Note that, the true charge for the winner of the auction is the highest bid price from her competitors, which is defined as the *market price*¹ in the second-price auction mechanism. So that, from an advertiser's perspective, predicting the market price is a crucial but challenging problem since the highest bid from hundreds or even thousands of advertisers for a specific ad impression is highly dynamic and almost impossible to predict by modeling each advertiser's strategy [1]. Moreover, only with the market price distribution, the advertiser can estimate the corresponding *winning probability* given an arbitrary bid price, which supports the subsequent bid decision making [26, 39]. For example, Lin et al. [19] adopted a bidding strategy by proposing the bid price according to the estimated market price and Ren et al. [26] presented a method of learning bidding strategy through optimizing advertisers' profits which requires the forecasted bid landscape for each bid request sample. Thus, it is more practical to model the market price as a stochastic variable [31, 32] and predict its distribution given each ad request feature, named as *bid landscape forecasting* [6, 29] and illustrated in Figure 1.

The previous works on bid landscape forecasting can be divided as two streams. The first stream is mainly based on statistically counting from the segmented samples, for example counting per campaign [41] or by some particular attribute combinations [29]. Different samples in the same segment share the same market price distribution which is too coarse-grained and often result in low prediction performance. The second stream is based on predefining a parameterized distribution form, such as log-normal distribution [6], Gaussian distribution [31, 32] or Gamma distribution [43], and then learning the distribution parameters with the observed data. However, as is discussed in [35] these assumptions are often too restricted and rejected by statistical tests thus lack of generalization.

¹The terms 'market price' and 'winning (bid) price' are used interchangeably in the related literatures [1],[6],[32]. In this paper, we use 'market price'.

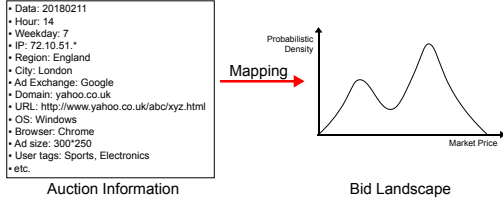


Figure 1: The main task of the bid landscape forecasting.

Yet there is another challenge of the bid landscape forecasting which is the *censorship* issue. Since RTB adopts the one-slot second-price auction mechanism [36], only the winner, who submits the highest bid price, will know the market price, i.e., the charged price, while others can only know that the market price is higher than their bids, which is called right-censored data [29, 32]. To handle this censorship, many works [29, 31, 32, 41] borrow the idea of survival analysis in medical data science and take the losing logs into consideration to better model the true market price distribution. However, these methods rely on only the losing logs and do not take a comprehensive view of considering both winning logs and losing logs for censorship handling.

In this paper, we propose a deep neural network methodology name as Deep Landscape Forecasting (DLF) model, without any presumed heuristic forms of market price distribution, to better capture the sophisticated patterns for each auction in bid landscape forecasting. Specifically, we utilize a recurrent neural network to model the conditional probability of the winning event given a bid. And then the model forecasts the distribution of market price by probability chain rule and naturally derive the winning probability distribution of arbitrary bid prices, for the given auction. We not only train the model through maximizing the log-likelihood of the true market price in the winning logs. Moreover, we also adopt a comprehensive loss function over both winning logs and losing logs, to handle the censorship.

The novelty of our methodology are threefold.

- **Fine-grained prediction:** Rather than forecasting the bid landscape over sample segments [29, 41], our method can predict the “personalized” market price distribution and the corresponding winning probability distribution for each specific bid request.
- **No assumption of distribution forms:** Based on the novel modeling methodology, our model manages to generate flexible forecasting results for each ad request without making any prior assumptions about the market price distribution, which will be illustrated in the experiment.
- **Novel Censorship loss function:** We adopt a comprehensive loss function for censorship handling and make a step further upon the traditional survival analysis methodologies [32] to better model the market price distribution.

To our knowledge, this is the first work that proposes an end-to-end deep learning model without any distributional assumptions for bid landscape forecasting. To deal with the right-censored problem, we use both the observed winning data and censored losing data to derive an unbiased learning. In addition, the experimental results over two real-world datasets show the significant improvement of our model over strong baselines under various evaluation metrics.

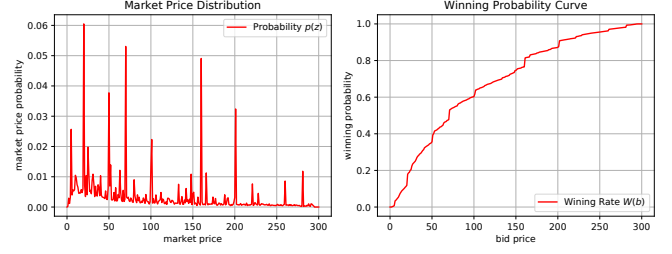


Figure 2: Market price distribution and winning function.

2 RELATED WORKS

Bid Landscape Forecasting. As is discussed in the above section, bid landscape forecasting has become an important component in RTB advertising and drawn much attention in recent works [19, 26, 27].

In the view of distribution modeling methods, there are two phases. In the early phase, researchers proposed several heuristic forms of functions to model the market price distribution. In [26, 27, 39], the authors provided some analytic forms of winning probability w.r.t. the bid price applied on the campaign level, which is based on the observation of the winning logs. Later in the recent researches, some well-studied distributions are applied in market price modeling. Cui et al. [6] presented a log-normal distribution to model the market price ground truth. Wu et al. [32] proposed a regression model based on Gaussian distribution to fit the market price. Recently, Gamma distribution for market price modeling has also been studied in the work [43]. The main drawback of these distributional methods is that these restricted empirical preassumptions may lose the effectiveness of handling various dynamic data and they even ignore the sophisticated real data divergence as we show in Figure 2.

In the view of forecasting, the goal is to predict the true market price distribution, i.e., bid landscape, for the given auction sample. A template-based method was present in [6] to fetch the corresponding market price distribution w.r.t. the given auction request. Wang et al. [29] proposed a clustering-based tree model to automatically segment the data samples and built a non-parametric estimator for each leaf node to predict the market price distribution. These methods can only perform coarse-grained bid landscape forecasting based on each data segment which misses the individual patterns of each ad request. The authors in [32] proposed a linear regression method to model the market price w.r.t. auction features. Nevertheless, those methods do not care much about the non-linear patterns in the real data, i.e., such as the co-occurrence and correlations between features [24] and the similarity/distinction among data segments, which may result in poor forecasting performance on different ad campaigns. Recently, Wu et al. [31] have proposed one work about winning price prediction with deep models. However, they use several assumptions about the form of market price distribution which is not flexible in practice. Moreover, as is stated in their paper, the goal of their method is to directly predict the winning price, rather than forecasting the bid landscape, thus their method does not outperform the tree-based model [29] over the log-likelihood metric which is one of the main evaluation metrics.

In this paper, we focus on fine-grained bid landscape forecasting at impression-level, and utilize deep recurrent neural network to flexibly model the non-linear and sequential patterns in market price distribution without any prior assumption of the distribution form.

Learning over Censored Data. The data censorship is another challenge for bid landscape forecasting. In the online advertising field, many models based on survival analysis have been studied so far. Wu et al. [31, 32] proposed a censored regression model using the lost auction data to alleviate the data bias problem. Nevertheless, the Gaussian distribution or other distributional assumptions [43] turn out to be too restricted while lacking of flexibility for modeling sophisticated yet practical distributions. Another problem is that these regression models [31, 32, 43] can only provide a point estimation, i.e., the expectation of the market price without standard deviation, which fails to provide winning probability estimation given an arbitrary bid price to support the subsequent bidding decision [26]. Amin et al. [1] implemented Kaplan-Meier estimator [11] for handling the data censorship in sponsored search. Kaplan-Meier estimator is a classic method in survival analysis which deals the right censored data in medical research [8, 25]. The authors of [29, 41] also utilized this non-parametric estimator to predict the winning probability. However, Kaplan-Meier estimator is merely statistically counting on the segmented data samples, thus fails to provide a fine-grained estimation, i.e., prediction on a single ad auction level.

Another school of survival analysis methods is Cox proportional hazard model [5]. This method commonly assumes that the instant hazard rate of event (i.e., auction winning in our case) occurrence is based on a base distribution multiplied by an exponential tuning factor. Recent works including [12, 20, 23] all used the Cox model with predefined base function to model the hazard rate of each sample, such as Weibull distribution, log-normal distribution or log-logistic distribution [15]. However, the problem is that the strong assumption of the data distribution may result in poor generalization in real-world data.

Considering all the limitations above, we propose our deep landscape forecasting method which models the conditional winning probability given the sequential price patterns through recurrent neural network, and maximizes the partial likelihood over the historic winning and losing data. Note that our model takes end-to-end learning in a unified learning objective without making any distributional assumptions, which is capable of fitting various bid landscape data and providing fine-grained prediction for each ad impression.

Recurrent Neural Network. Due to its adequate model capability and the support of big data, deep learning, a.k.a. deep neural network, has drawn great attention. Among them, recurrent neural network (RNN) whose idea was firstly developed two decades ago shows satisfying performance for sequential pattern mining. And its variants like long short-term memory (LSTM) [10] employ memory structures to better captures dynamic sequential patterns. In this paper we borrow the idea of RNN for modeling conditional winning probability. And in the experiments, our model shows

superior better performance against state-of-the-art baselines including survival analysis methods [12, 14] and recent bid landscape forecasting models [29, 31].

3 METHODOLOGY

In this section, we firstly present the preliminaries used in our discussion in Section 3.1 and formulate the problem in Section 3.2. Then in Section 3.3 we discuss our bid landscape forecasting model in detail. We conduct a deep analysis of the model in Section 3.4.

3.1 Preliminaries

In RTB scenario, the advertiser is asked to propose a bid price b after receiving a bid request \mathbf{x} for auction. The bid request contains three parts of the auction information including user (e.g., location, browser label, etc.), publisher (e.g., web URL and ad slot size, etc.) and the ad content (e.g., product type, time and creative content). The goal of the advertiser is to propose an appropriate bid price and win the auction in a cost-effective manner [26, 36].

One of the challenges is that it is infeasible to model the bidding strategy of each competing bidder since the participating advertisers do not interact with each other [29]. It is natural to model the market as a whole and regard the market price as a variable z [19, 29, 39]. Recall that the market price is the second highest bid price among all the bidders in the second-price auction, i.e., the highest bid price from the competitors in the view of the auction winner. The probability density function (P.D.F.) of the market price z is $p(z)$, $z > 0$.

Now that we have the P.D.F. $p(z)$ of the market price z , we can derive the winning probability of proposing the price at b as

$$W(b) \doteq \Pr(z < b) = \int_0^b p(z)dz, \quad (1)$$

which is the probability that our bid price is larger than the market price, i.e., winning the auction. Then the straightforward definition of the “losing” function is

$$S(b) \doteq \Pr(z \geq b) = 1 - W(b) = \int_b^\infty p(z)dz, \quad (2)$$

which represents the losing probability of proposing the bid price b . Note that in survival analysis [12, 17], the market price is regarded as the patient’s underlying survival period and the bid price is the investigation period, thus **winning** and **losing** the ad auction respond to the “death” and “survival” status of one patient [41].

3.2 Problem Definition

The data of the bidding logs are represented as a set of triples $\{(\mathbf{x}, b, z)\}$, where \mathbf{x} is the feature of the bid request, b is the proposed bid price in that auction. Here z is the observed market price if the advertiser previously won this auction and she has already known the true market price, but z is unknown (and we marked z as null) for those losing auctions.

The main problem of bid landscape forecasting is to estimate the probability distribution $p(z|\mathbf{x})$ of the market price z with regard to the bid request feature \mathbf{x} . Formally speaking, the derived model is a “mapping” function which learns the patterns within the data and

predicts the market price distribution of each auction as

$$p(z|\mathbf{x}) = T(\mathbf{x}). \quad (3)$$

The general goal has been illustrated in Figure 1.

3.3 Deep Landscape Forecasting

In this part, we formulate our deep learning method with censorship handling for bid landscape forecasting, which we call as Deep Landscape Forecasting (DLF) model.

3.3.1 Discrete Price Model. First we transform the modeling from continuous space to discrete space. Note that, since all the price in real-time bidding advertising is discrete, it is natural to propose the discrete price model and derive the probability functions in the discrete price schema.

In the discrete context, a set of L prices $0 < b_1 < b_2 < \dots < b_L$ is obtained which arises from the finite precision of price determinations. Analogously we may also consider the grouping of continuous prices as $l = 1, 2, \dots, L$ uniformly divided disjoint intervals $V_l = (b_l, b_{l+1}]$ where $b_0 = 0$ and b_L is the last observation interval boundary for the given sample, i.e., the proposed bid price in the auction. V_L is the largest price interval in the whole price space. This setting is appropriately suited in our task and has been widely used in medical research [17] and bid landscape forecasting field [29, 41] where the price is always integer [40] thus we set $b_{l+1} - b_l = 1$.

As such, our winning function and losing function over discrete price space is

$$\begin{aligned} W(b_l) &\doteq \Pr(z < b_l) = \sum_{j < l} \Pr(z \in V_j), \\ S(b_l) &\doteq \Pr(b_l \leq z) = \sum_{j \geq l} \Pr(z \in V_j), \end{aligned} \quad (4)$$

where the input to the two functions is the bid price b_l from the advertiser. And the discrete market price probability function at the l -th price interval is

$$\begin{aligned} p_l &= \Pr(z \in V_l) = W(b_{l+1}) - W(b_l) \\ &= [1 - S(b_{l+1})] - [1 - S(b_l)] \\ &= S(b_l) - S(b_{l+1}). \end{aligned} \quad (5)$$

We define the *conditional winning probability* given the price b_l as

$$h_l = \Pr(z \in V_l | z \geq b_{l-1}) = \frac{\Pr(z \in V_l)}{\Pr(z \geq b_{l-1})} = \frac{p_l}{S(b_{l-1})}, \quad (6)$$

which means the probability that the market price z lies in the interval $V_l = (b_l, b_{l+1}]$ given the condition that z is larger than the bid prices which are smaller than b_l . The meaning of h_l is the conditional probability of *just* winning the auction by proposing the bid price at the l -th price interval.

3.3.2 Recurrent Neural Network Model. Till now, we have presented the discrete price model and discuss the winning and losing probability over the discrete price space. We here propose our DLF model based on recurrent neural network f_θ with the parameter θ , which captures the sequential patterns for conditional probability h_l^i at every price interval V_l for the i -th sample.

The detailed structure of DLF network is illustrated in Figure 3. At each price interval V_l , the l -th RNN cell predicts the conditional

winning probability h_l^i given the bid request feature \mathbf{x}^i conditioned upon the previous events as

$$\begin{aligned} h_l^i &= \Pr(z \in V_l | z \geq b_{l-1}, \mathbf{x}^i; \theta) \\ &= f_\theta(\mathbf{x}^i, b_l | \mathbf{r}_{l-1}), \end{aligned} \quad (7)$$

where f_θ is the RNN function taking (\mathbf{x}^i, b_l) as input and h_l^i as output. \mathbf{r}_{l-1} is the hidden vector calculated from the last RNN cell. In our paper we implement the RNN function as a standard LSTM unit [10], which has been widely used in sequence data modeling. We describe the implementation details of f_θ in the appendix.

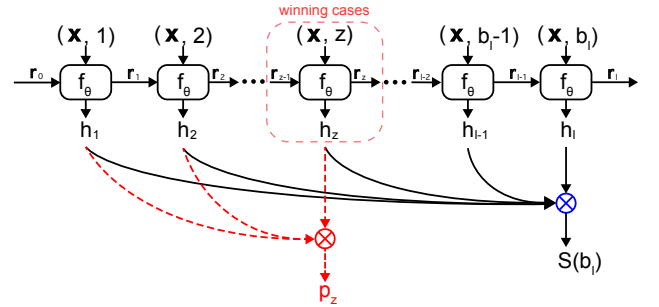


Figure 3: Detailed illustration of DLF model. Note that only the winning cases have the true market price and can calculate p_z for the loss L_1 .

From Eqs. (4), (6) and (7), we can easily derive the losing probability function $S(b)$ and the winning probability function $W(b)$ with the bidding price b for the i -th individual sample as

$$\begin{aligned} S(b|\mathbf{x}^i; \theta) &= \Pr(b \leq z|\mathbf{x}^i; \theta) = \Pr(z \notin V_1, z \notin V_2, \dots, z \notin V_l|\mathbf{x}^i; \theta) \\ &= \Pr(z \notin V_1|\mathbf{x}^i; \theta) \cdot \Pr(z \notin V_2|z \geq b_1, \mathbf{x}^i; \theta) \cdot \dots \\ &\quad \cdot \Pr(z \notin V_{l_b}|z \geq b_1, \dots, z \geq b_{l_b-1}, \mathbf{x}^i; \theta) \\ &= \prod_{l:l \leq l_b} [1 - \Pr(z \in V_l | z \geq b_l, \mathbf{x}^i; \theta)] = \prod_{l:l \leq l_b} (1 - h_l^i), \\ W(b|\mathbf{x}^i; \theta) &= \Pr(b > z|\mathbf{x}^i; \theta) = 1 - S(b|\mathbf{x}^i; \theta) = 1 - \prod_{l:l \leq l_b} (1 - h_l^i), \end{aligned} \quad (8)$$

where l_b is the price interval index for b . Here we use probability chain rule to calculate the joint losing probability $S(b)$ at the given bid price b through multiplying the conditional losing probability $(1 - h_l)$, i.e., inverse of the conditional winning probability.

Moreover, taking Eqs. (5) and (6) into consideration, the probability of market price z directly lying in the interval of V_l for the i -th sample is

$$p_l^i = \Pr(z \in V_l|\mathbf{x}^i; \theta) = h_l^i \prod_{l:l < l^i} (1 - h_l^i). \quad (9)$$

3.3.3 Loss Functions. Since there is no ground truth of either market price distribution or winning probability, here we maximize the log-likelihood over the empirical data distribution to learn our deep model. We consider from two aspects for the loss function.

The first loss is based on the P.D.F. and it aims to minimize the negative log-likelihood of the market price $z = z^i$ over the winning

logs as

$$\begin{aligned}
 L_1 &= -\log \prod_{(\mathbf{x}^i, z^i) \in \mathbb{D}_{\text{win}}} \Pr(z^i \in V_{l^i} | \mathbf{x}^i; \theta) = -\log \prod_{(\mathbf{x}^i, z^i) \in \mathbb{D}_{\text{win}}} p_{l^i}^i \\
 &= -\log \prod_{(\mathbf{x}^i, z^i) \in \mathbb{D}_{\text{win}}} h_{l^i}^i \prod_{l: l < l^i} (1 - h_l^i) \\
 &= - \sum_{(\mathbf{x}^i, z^i) \in \mathbb{D}_{\text{win}}} \left[\log h_{l^i}^i + \sum_{l: l < l^i} \log(1 - h_l^i) \right], \tag{10}
 \end{aligned}$$

where l^i is the interval index of the true market price $z^i \in V_{l^i}$ given the feature vector \mathbf{x}^i .

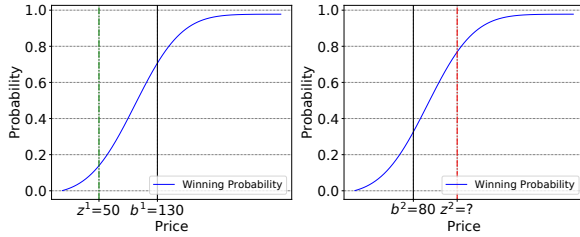


Figure 4: Two examples of winning curves.

The second loss is based on the C.D.F., i.e., winning probability distribution. Recall that there are winning cases and losing cases in the dataset. As is shown in Figure 4, the left subfigure is the winning case where z^1 has been known and $z^1 < b^1$; The right figure is the losing case where z^2 is unknown (censored) but we only have the knowledge that $z^2 \geq b^2$. Thus, there are two motivations about the second loss.

For the winning cases as in the left part of Figure 4, we need to “push down” the winning probability during the price range of $[0, z]$, while “pull up” the winning probability during the price range of $[z, \infty)$, especially for the winning probability in $[z, b]$. Thus, on one hand, we adopt the loss over the winning cases that

$$\begin{aligned}
 L_{\text{win}} &= -\log \prod_{(\mathbf{x}^i, b^i) \in \mathbb{D}_{\text{win}}} \Pr(b^i > z | \mathbf{x}^i; \theta) \\
 &= -\log \prod_{(\mathbf{x}^i, b^i) \in \mathbb{D}_{\text{win}}} W(b^i | \mathbf{x}^i; \theta) \\
 &= - \sum_{(\mathbf{x}^i, b^i) \in \mathbb{D}_{\text{win}}} \log \left[1 - \prod_{l: l \leq l^i} (1 - h_l^i) \right]. \tag{11}
 \end{aligned}$$

As for the losing cases in the right part of Figure 4, we just need to “push down” the winning probability since we have no idea about the true market price but we only know that $z \geq b$. On the other hand, we just adopt the loss over the losing auctions as

$$\begin{aligned}
 L_{\text{lose}} &= -\log \prod_{(\mathbf{x}^i, b^i) \in \mathbb{D}_{\text{lose}}} \Pr(z \geq b^i | \mathbf{x}^i; \theta) \\
 &= -\log \prod_{(\mathbf{x}^i, b^i) \in \mathbb{D}_{\text{lose}}} S(b^i | \mathbf{x}^i; \theta) \\
 &= - \sum_{(\mathbf{x}^i, b^i) \in \mathbb{D}_{\text{lose}}} \sum_{l: l \leq l^i} \log(1 - h_l^i). \tag{12}
 \end{aligned}$$

3.4 Model Realization

In this section, we unscramble some intrinsic properties of our deep model and analyze the model efficiency in this section.

Properties of Loss Function. First of all, we take the view of winning prediction of our methodology. As is known that there is a winning status, i.e., an indicator of winning the auction, for each sample as

$$w^i = \begin{cases} 1, & \text{if } b^i > z^i, \\ 0, & \text{otherwise } b^i \leq z^i. \end{cases} \tag{13}$$

For the winning logs, each sample (\mathbf{x}^i, z^i, b^i) is uncensored (i.e., z^i is known) where $w^i = 1$. While for the losing logs, the true market price z^i is unknown but the advertiser only has the idea that $z^i \geq b^i$, so that $w^i = 0$.

Therefore, taking Eqs. (11) and (12) altogether and we may find that the combination of L_{win} and L_{lose} describes the classification of winning the auction as

$$\begin{aligned}
 L_2 &= L_{\text{win}} + L_{\text{lose}} \\
 &= -\log \prod_{(\mathbf{x}^i, b^i) \in \mathbb{D}_{\text{win}}} \Pr(b^i > z | \mathbf{x}^i; \theta) - \log \prod_{(\mathbf{x}^i, b^i) \in \mathbb{D}_{\text{lose}}} \Pr(z \geq b^i | \mathbf{x}^i; \theta) \\
 &= -\log \prod_{(\mathbf{x}^i, b^i) \in \mathbb{D}} [W(b^i | \mathbf{x}^i; \theta)]^{w^i} \cdot [1 - W(b^i | \mathbf{x}^i; \theta)]^{1-w^i} \tag{14} \\
 &= - \sum_{(\mathbf{x}^i, b^i) \in \mathbb{D}} \{ w^i \cdot \log W(b^i | \mathbf{x}^i; \theta) + (1 - w^i) \log [1 - W(b^i | \mathbf{x}^i; \theta)] \},
 \end{aligned}$$

which is the cross entropy loss for predicting winning probability when bidding at b^i given \mathbf{x}^i over all the data $\mathbb{D} = \mathbb{D}_{\text{win}} \cup \mathbb{D}_{\text{lose}}$.

Combining all the objective functions and our goal is to minimize the negative log-likelihood over all the data samples including both winning logs and losing logs as

$$\arg \min_{\theta} \alpha L_1 + (1 - \alpha) L_2, \tag{15}$$

where the hyperparameter α controls the order of magnitudes of the gradients from the two losses at the same level to stabilize the model training.

In the traditional survival analysis methods [5, 13] and the related works for bid landscape forecasting [32, 43], they usually adopts only L_1 based on P.D.F. and L_{lose} for censorship handling. We propose a comprehensive loss function which learns from both winning logs and losing logs. From the discussion above, L_{win} and L_{lose} collaboratively learns the data distribution from the C.D.F. view.

Model Efficiency. Here we analyze the computational complexity of our DLF model. As is shown in Eq. (7), each recurrent unit f_{θ} takes $(\mathbf{x}, b_l, \mathbf{r}_{l-1})$ as input and outputs probability scalar h_l and hidden vector \mathbf{r}_l to the next unit. Recall that the maximal price interval is L , so the calculation of the recurrent units will runs for maximal L times. We assume the average case time performance of recurrent units f_{θ} is $O(C)$, which is related to the implementation of the unit [38], e.g., recurrent depth, recurrent skip coefficients, yet can be parallelized through GPU processor. The subsequent calculation is to obtain the multiplication results of h_l or $(1 - h_l)$ to get the results of $p(z)$ and $S(b)$, as that in Figure 3, whose complexity is $O(L)$. Thus the overall time complexity of DLF model is $O(CL) +$

$O(L) = O(CL)$, which is the same as the original recurrent neural network model.

In many literatures, recurrent neural networks have been deployed in recommender system [30], online advertising platform [42] and machine translation system [33], each of which shows promising time efficiency in large scale online systems and, to some extent, guarantees online inference efficiency for our DLF model. We may also optimize the implementation of the RNN unit through other techniques, such as Quasi-RNN [3] and sliced-RNN [34]. Moreover, the landscape forecasting module could be parallelly executed with the utility estimation in RTB scenario, e.g., click-through rate prediction model, and jointly feed the results for final bid decision making. In our experiments, under the recommended settings, we evaluate our model and it achieved 22 milliseconds for averaged inference time given one sample, satisfying the 100 milliseconds requirement of bid decision in the RTB scenario [28].

4 EXPERIMENTS

In this section, we present the experimental setup and the corresponding results under various evaluation metrics with significance test. Furthermore, we look deeper into our model and analyze some insights of the experiment results. Moreover, we have published the implementation code for reproducible experiments².

4.1 Datasets and Experiment Flow

We use two real-world RTB datasets in our experiment. **iPinYou** RTB dataset, which has been published in [18], contains 64.7M bidding records, 19.5M impressions, 14.79K clicks and 16.0K CNY expense on 9 campaigns from different advertisers during 10 days in 2013. Each bidding log has 16 attributes, including weekday, hour, user agent, region, ad slot ID, etc. The auctions during the last 3 days are set as test data while the rest as training data. The other bidding dataset is **YOYI** dataset which was published in [27]. It includes 402M impressions, 500K clicks and 428K CNY expense during 8 days in Jan. 2016. More details of the two datasets have been provided in [18] and [27] respectively.

Data Preparation. For simulating the real bidding market in an online fashion and show the advantages of our deep survival model, we take the original data of impression log as full-volume auction data, and perform a truthful bidding strategy [16] to simulate the bidding process, which produces the winning bid dataset \mathbb{D}_{win} and the losing bid dataset \mathbb{D}_{lose} respectively. For each data sample $(x_{\text{win}}, b_{\text{win}}, z_{\text{win}}) \in \mathbb{D}_{\text{win}}$, the real market price $z_{\text{win}} < b_{\text{win}}$ is known for the advertisers, while for each $(x_{\text{lose}}, b_{\text{lose}}, z_{\text{lose}}) \in \mathbb{D}_{\text{lose}}$ the corresponding market price $z_{\text{lose}} \geq b_{\text{lose}}$ is hidden. It guarantees the similar situation as that faced by all the advertisers in the real world marketplace. This simulation and data processing method have been widely used in bid landscape forecasting literatures [29, 32, 41, 43].

After data preparation, we make some statistics over the resulted datasets. As is illustrated in Table 1, we can find that the averaged market price in \mathbb{D}_{win} is much lower than that of \mathbb{D}_{lose} , which is reasonable because of the second-price auction mechanism and

Table 1: The statistics of the datasets. WR: winning rate; AMP: averaged market price.

Campaign	Total #	Winning #	WR	AMP	AMP (\mathbb{D}_{win})	AMP (\mathbb{D}_{lose})
1458	3,697,694	1,116,644	0.3020	69.6696	27.4265	87.9452
2259	1,252,753	396,283	0.3163	96.7888	27.1986	128.9877
2261	1,031,479	321,931	0.3121	87.6479	18.9000	118.8396
2821	1,984,525	228,833	0.1153	93.8962	13.2118	104.4125
2997	468,500	70,747	0.1510	60.4188	7.2762	69.8711
3358	2,043,032	315,010	0.1542	95.4967	21.2540	109.0308
3386	3,393,223	819,447	0.2415	78.0327	23.8983	95.2682
3427	3,130,560	654,989	0.2092	81.9650	25.2118	96.9808
3476	2,494,208	723,847	0.2902	80.0719	31.2218	100.0453
Overall	19495974	4647731	0.2384	82.0744	25.0484	99.9244
YOYI	401,617,064	202,214,191	0.5035	55.7444	24.4488	87.4842

also reflects the bias of the model without using losing (censored) logs.

In these datasets, since all the prices are integer value, we bucketize the discrete price interval as $\text{interval_size} = 1$ and the maximal price interval number L is equal to the largest integer price in the dataset where $L = 300$.

Evaluation Phase. In this phase, the corresponding market price distribution $p(z|\mathbf{x})$ with the true market price of each sample \mathbf{x} in the test data is estimated by all of the compared models respectively. The corresponding winning function $W(b|\mathbf{x})$ and losing function $S(b|\mathbf{x})$ can be easily obtained through the forecasted market price distribution $p(z|\mathbf{x})$ as that in Eqs. (1), (2) and (4). We assess the performance of different settings in several measurements, as listed in the next subsection.

4.2 Evaluation Metrics

In our experiments, we evaluate all the models under two metrics and conduct the significance test between our model and the other baselines. Note that, there are two goals for bid landscape forecasting, i.e., forecasting of market price distribution (P.D.F.) and the corresponding winning probability (C.D.F.) estimation given arbitrary bid prices.

First we use average negative log probability (**ANLP**) as [29] to evaluate the performance of forecasting the market price distribution. Specifically, ANLP is to assess the likelihood of the co-occurrence of the test bid requests with the corresponding market prices, which is calculated as

$$\bar{P} = -\frac{1}{|\mathbb{D}_{\text{test}}|} \sum_{(\mathbf{x}^i, z^i) \in \mathbb{D}_{\text{test}}} \log p_z(z^i|\mathbf{x}^i), \quad (16)$$

where $p_z(z|\mathbf{x})$ is the learned bid landscape forecasting function of each model.

The last evaluation metric is concordance index (**C-index**), which is the most common evaluation used in survival analysis [9, 17, 20] and reflects a measure of how well a model predicts the ordering of samples according to their market prices. That is, given the bid price b , two auction samples $d^1 = (\mathbf{x}^1, z^1)$ with large market price $z^1 \geq b$ and $d^2 = (\mathbf{x}^2, z^2)$ with small market price $z^2 < b$ should be ordered as $d^1 < d^2$ where d^1 is placed before d^2 . This evaluation is the same as the area under ROC curve (AUC) metric in the classification tasks [24, 27] when there is only one event of interest (i.e., winning in our task) [17]. From the classification view of auction winning probability estimation by proposing b , C-index assesses

²Reproducible code: <https://github.com/rk2900/DLF>.

the ordering performance among all the winning and losing pairs at b among the test data thus illustrates the performance of winning probability estimation.

Finally, we conduct the significance test to verify the statistical significance of the performance improvement of our model w.r.t. the baseline models. Specifically, we deploy a Mann-Whitney U test [21] under C-index metric, and a t-test [2] under ANLP metric.

4.3 Compared Settings

We compare our proposed DLF model with nine baseline models including traditional Cox proportional hazard function model, survival tree model, multi-task learning method and other deep learning models.

- **KM** is Kaplan-Meier estimator [11] which is a statistic-based non-parametric method and has been used in several bid landscape forecasting works [29, 41].
- **Lasso-Cox** is a semi-parametric method [37] based on Cox proportional hazard model [5] with l_1 -regularization.
- **DeepSurv** is a Cox proportional hazard model with deep neural network [13] for feature extraction upon the sample covariates. The loss function is the negative partial likelihood of the winning and losing outcomes.
- **Gamma** is the gamma distribution based regression model [43]. The winning price of each bid request is modeled by a unique gamma distribution with respect to its features.
- **MM** is the mixture regression model. This model uses both linear regression and censored regression, and combines two models and predicts as a mixture manner [32].
- **MTLSA** is the recently proposed multi-task learning model [17]. It transforms the original survival analysis problem into a series of binary classification problems, and uses a multi-task learning method. The original model predicts the death rate of a patient, we change it to predict the winning rate of bidding in an auction.
- **STM** is the survival tree model. This model combines Kaplan-Meier estimator and decision trees with bi-clustering to predict bid landscape. This model is proposed in [29] and, to our knowledge, achieved state-of-the-art performance in bid landscape forecasting.
- **DeepHit** is a deep neural network model [14] which predicts the probability of each bidding price from the minimum price to the maximum price.
- **DWPP** [31] is a deep winning price prediction method using neural network to directly predict the market price, with assumption of the distribution form as Gaussian distribution.
- **RNN** is based on our DLF model. However, it only optimizes over the winning logs *without* considering the censored information, whose loss function is only $(L_1 + L_{win})$. This model is used to illustrate the power of partial likelihood loss L_{lose} over the censored losing data.
- **DLF** is our deep landscape forecasting model which has been described in Section 3.

The details of the experimental configurations, such as hardware and training procedure, have been included in the appendix.

4.4 Experimental Results

In this part, we present the detailed performance of all the compared models over the evaluation metrics.

Table 2: Averaged negative log probability performance: the smaller, the better. (* indicates p-value $< 10^{-6}$ in significance test).

iPinYou	ANLP									
	KM	Lasso-Cox	DeepSurv	Gamma	MM	MTLSA	STM	DeepHit	DWPP	RNN
1458	10.532	38.608	38.652	5.956	5.788	9.791	4.761	5.510	29.204	9.506
2259	14.671	28.234	29.658	6.069	7.328	10.248	5.471	5.586	39.263	9.625
2261	14.665	39.129	39.390	5.986	7.020	10.261	4.818	5.442	32.805	9.417
2821	19.582	43.099	43.072	7.838	7.262	9.895	5.572	5.614	40.537	23.099
2997	16.203	32.849	33.052	5.999	6.702	9.167	5.083	5.470	34.940	16.639
3358	19.253	44.769	44.885	6.736	7.177	9.484	5.539	5.616	40.958	13.806
3386	15.973	39.781	41.943	6.488	6.141	8.834	5.228	5.549	32.550	10.743
3427	16.902	41.558	41.698	6.002	6.185	9.090	5.321	5.552	33.387	9.565
3476	10.507	39.551	39.518	5.710	6.022	10.240	4.537	5.554	31.609	7.891
Overall	15.366	38.620	39.096	6.310	6.552	9.668	5.148	5.544	35.028	12.255
YOYI	7.907	30.946	27.897	6.475	5.652	10.286	4.503	5.567	29.108	5.885

4.4.1 Landscape Forecasting Performance. We first analyze the probability density estimation performance for learning the bid landscape forecasting. Though there is no ground truth for the market price distribution $p(z)$, we may also use the negative log-likelihood result to evaluate the performance over the test data. Table 2 lists the ANLP performance of the compared models. From the table, we may find that our DLF model achieves significant improvements against the other baselines including the state-of-the-art model STM on both iPinYou and YOYI datasets.

We also find from the table that (i) The survival tree model STM achieves relatively better performance than other baselines which may be the result of the well clustering methodology and the non-parametric survival analysis. (ii) All of the models with survival analysis, i.e., DeepSurv, Gamma, MM, STM, MTLSA, DeepHit, perform much better than RNN model which does not consider censored data into model training. (iii) DeepHit model gets worse results than DLF model probably for the reason that it does not model the price-level sequential dependency as that in our method, which in contrast reflects that the conditional sequential modeling of DLF model in Eq. (7) has significantly improved the forecasting performance in the market price distribution modeling. (iv) Though DWPP utilizes deep model for feature extraction, it performs poor under ANLP metric which has also been reported in their paper [31]. The reason may be the assumed Gaussian form of the market price distribution lacks generalization in the practical applications.

Table 3: C-index performance: the larger, the better, (* indicates p-value $< 10^{-6}$ in significance test).

iPinYou	C-index									
	KM	Lasso-Cox	DeepSurv	Gamma	MM	MTLSA	STM	DeepHit	DWPP	RNN
1458	0.698	0.820	0.835	0.612	0.698	0.505	0.764	0.861	0.866	0.894
2259	0.685	0.775	0.791	0.584	0.685	0.505	0.768	0.785	0.729	0.791
2261	0.666	0.847	0.890	0.564	0.666	0.508	0.812	0.838	0.807	0.874
2821	0.677	0.741	0.714	0.563	0.678	0.507	0.790	0.810	0.746	0.737
2997	0.734	0.910	0.852	0.641	0.734	0.517	0.835	0.907	0.885	0.762
3358	0.704	0.866	0.896	0.601	0.706	0.542	0.811	0.888	0.744	0.819
3386	0.716	0.845	0.854	0.569	0.719	0.512	0.849	0.881	0.833	0.800
3427	0.724	0.830	0.845	0.586	0.742	0.508	0.798	0.873	0.796	0.804
3476	0.692	0.865	0.877	0.676	0.692	0.505	0.830	0.879	0.861	0.917
Overall	0.700	0.834	0.840	0.600	0.703	0.513	0.807	0.858	0.807	0.823
YOYI	0.791	0.847	0.862	0.528	0.791	0.510	0.886	0.878	0.856	0.898

4.4.2 Winning Prediction Performance. In this part we illustrate the measurement of the winning prediction under the given bid price b of the sample \mathbf{x} . As is discussed before, this can be regarded as a binary classification problem, so we present the performance of C-index in Table 3. From the table we can observe that DLF achieves the best C-index value among all the compared models on both iPinYou and YOYI datasets which show the classification effectiveness of our model. Especially, our model gains over 12.9% average improvements against the state-of-the-art STM model.

We can also conduct below findings from the table. (i) All the deep models including DeepSurv, DeepHit, DWPP, RNN and DLF have relatively better C-index performance than the other non-deep baselines. Even the RNN model without any censorship handling achieves satisfying prediction performance, which again reflects the advantage of our novel modeling perspective with sequential pattern mining. (ii) MM and Gamma do not perform well which may be accounted for that these models adopt restricted assumptions of the base distribution for the probability dense function. This phenomenon verifies our analysis in Sections. 1 and 2 and reveals the importance of modeling without distributional assumptions. (iii) DWPP performs not well which is reasonable because it is optimized for market price regression rather than winning probability estimation. However, please note that, the forecasting of market price distribution and the corresponding winning probability estimation are more general for RTB advertising.

4.4.3 Model Convergence. To illustrate the model training and convergence of DLF model, we plot the learning curve and the evaluation results on iPinYou Campaign 3476 and YOYI datasets in Figure 5. Recall that our model optimizes over two loss functions, i.e., the ANLP loss L_1 and the cross entropy loss L_2 . We apply a method [4] whose idea is to feed the batch of training data under each one of the two losses alternatively. From the learning curves we can find that (i) DLF converges quickly and the values of both losses drop to stable convergence at about the first complete iteration over the whole training dataset. (ii) The two losses are alternatively optimizing and facilitate each other during the training, which proves the learning stability of our model.

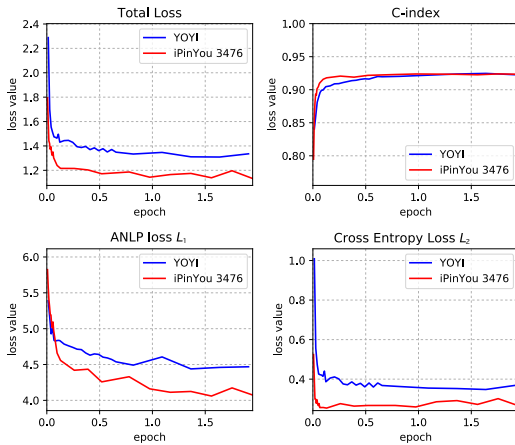


Figure 5: Learning curves over Campaign 3476 of iPinYou and YOYI. Here “epoch” means one iteration over the whole training data and $\alpha = 0.25$ in Eq. (15).

4.5 Further Investigation

In this section, we illustrate some comprehensive analysis of the prediction results of different models. As is illustrated in Figure 6, we plot the winning probability (C.D.F.) and the corresponding market price distribution (P.D.F.) of an example sample, calculated from all the compared models. The true market price of this sample is $z^i = 67$.

Accurate Bid Landscape Forecasting. As is illustrated in the figure, when the value of market price distribution $p(z^i)$ is high, the corresponding winning probability increases rapidly. Thus accurately predicting $p(z^i)$ makes the winning probability estimation reasonable. From the figure, we may find that our DLF model accurately places *highest* probability density on the true market price $z^i = 67$, while the other models cannot conduct reasonable forecasting results. This finding reflects the advantage of our DLF model under ANLP metric since it directly measures the accuracy of bid landscape forecasting.

Flexibility of Forecasted Distribution. As has been discussed before, we do not make any assumptions of the distribution form of either market price distribution or winning probability function, thus our DLF model can model sophisticated distribution forms as illustrated in the figure. However, those models assuming specific distribution forms, i.e., Lasso-Cox, Gamma, DeepSurv and DWPP, cannot reasonably model this practical case since their strong assumptions lack generalization in real-world applications. Specifically, Lasso-Cox, DeepSurv and DWPP show similar results of P.D.F. and C.D.F. which may be accounted for the same Gaussian distribution form adopted by these models. Thus it further shows the disadvantage of assuming distribution forms for bid landscape forecasting.

Effective Censorship Handling. Comparing the forecasted result of DLF with that of RNN which lacks the censorship handling, we can find that though RNN model predicts the market price distribution and the winning probability with the similar shape to our DLF model, it over-estimates the winning probability and place the probability density of $p(z^i)$ not accurately. It is easy to explain since without censorship handling, the model may result in biased landscape forecasting results as also shown in [29].

5 CONCLUSION AND FUTURE WORK

In this paper, we analyzed the challenges of bid landscape forecasting and the cons of state-of-the-art methods. To tackle with these problems, we proposed a fined-grained bid landscape forecasting model based on deep recurrent neural network and survival analysis modeling. Note that we do not assume any distribution forms for bid landscape forecasting. Our DLF model not only captures complex patterns in the bid landscape of each bid request, but also considers prediction of the winning status over both winning logs and losing (censored) logs. The comprehensive experiment results have shown the significant advantages of our model comparing with other strong baselines. A deep investigation has been performed to verify the robustness and correctness of our model.

For the future work, we plan to incorporate the proposed bid landscape forecasting model into bid optimization for profit maximization [7, 19, 26] in real-time bidding advertising.

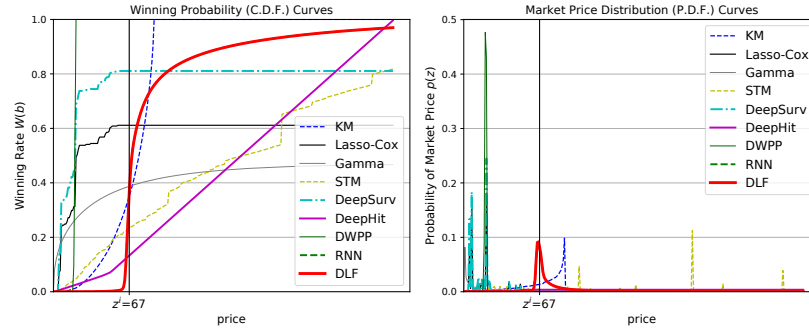


Figure 6: A comprehensive visualization of winning probability $W(b|x^i)$ estimation and market price probability $p(z|x^i)$ prediction over different models. The vertical black line is the true market price $z^i = 67$ of this sample.

Acknowledgments. The corresponding author Weinan Zhang thanks the support of National Natural Science Foundation of China (61702327, 61772333, 61632017) and Shanghai Sailing Program (17YF1428200).

REFERENCES

- [1] Kareem Amin, Michael Kearns, Peter Key, and Anton Schwaighofer. 2012. Budget optimization for sponsored search: Censored learning in MDPs. *UAI* (2012).
- [2] Bhaskar Bhattacharya and Desale Habtzghi. 2002. Median of the p value under the alternative hypothesis. *The American Statistician* (2002).
- [3] James Bradbury, Stephen Merity, Caiming Xiong, and Richard Socher. 2017. Quasi-recurrent neural networks. *ICLR* (2017).
- [4] Xuezhi Cao, Haokun Chen, Xuejian Wang, Weinan Zhang, and Yong Yu. 2018. Neural Link Prediction over Aligned Networks. In *AAAI*.
- [5] David R Cox. 1992. Regression models and life-tables. In *Breakthroughs in statistics*. Springer.
- [6] Ying Cui, Ruofei Zhang, Wei Li, and Jianchang Mao. 2011. Bid landscape forecasting in online ad exchange marketplace. In *KDD*.
- [7] Diemert Eustache, Meynet Julien, Pierre Galland, and Damien Lefortier. 2017. Attribution Modeling Increases Efficiency of Bidding in Display Advertising. In *ADKDD*.
- [8] Louis Gordon and Richard A Olshen. 1985. Tree-structured survival analysis. *Cancer treatment reports* (1985).
- [9] Frank E Harrell, Kerry L Lee, Robert M Califf, David B Pryor, and Robert A Rosati. 1984. Regression modelling strategies for improved prognostic prediction. *Statistics in medicine* (1984).
- [10] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* (1997).
- [11] Edward L Kaplan and Paul Meier. 1958. Nonparametric estimation from incomplete observations. *Journal of the American statistical association* (1958).
- [12] Jared Katzman, Uri Shaham, Jonathan Bates, Alexander Cloninger, Tingting Jiang, and Yuval Kluger. 2016. Deep survival: A deep cox proportional hazards network. *arXiv preprint arXiv:1606.00931* (2016).
- [13] Jared L Katzman, Uri Shaham, Alexander Cloninger, Jonathan Bates, Tingting Jiang, and Yuval Kluger. 2018. DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network. *BMC medical research methodology* 18, 1 (2018), 24.
- [14] Changhee Lee, William R Zame, Jinsung Yoon, and Mihaela van der Schaar. 2018. Deephit: A deep learning approach to survival analysis with competing risks. *AAAI*.
- [15] Elisa T Lee and John Wang. 2003. *Statistical methods for survival data analysis*. Vol. 476. John Wiley & Sons.
- [16] Kuang-chih Lee, Burkay Orten, Ali Dasdan, and Wentong Li. 2012. Estimating conversion rate in display advertising from past performance data. In *KDD*.
- [17] Yan Li, Jie Wang, Jieping Ye, and Chandan K Reddy. 2016. A Multi-Task Learning Formulation for Survival Analysis. In *KDD*.
- [18] Hairan Liao, Lingxiao Peng, Zhenchuan Liu, and Xuehua Shen. 2014. iPinYou global rtb bidding algorithm competition dataset. In *ADKDD*.
- [19] Chi-Chun Lin, Kun-Ta Chuang, Wush Chi-Hsuan Wu, and Ming-Syan Chen. 2016. Combining Powers of Two Predictors in Optimizing Real-Time Bidding Strategy under Constrained Budget. In *CIKM*.
- [20] Margaux Luck, Tristan Sylvain, Héloïse Cardinal, Andrea Lodi, and Yoshua Bengio. 2017. Deep Learning for Patient-Specific Kidney Graft Survival Analysis. *arXiv preprint arXiv:1705.10245* (2017).
- [21] Simon J Mason and Nicholas E Graham. 2002. Areas beneath the relative operating characteristics (ROC) and relative operating levels (ROL) curves: Statistical significance and interpretation. *Quarterly Journal of the Royal Meteorological Society* (2002).
- [22] S Muthukrishnan. 2009. Ad exchanges: Research issues. In *Internet and network economics*. Springer, 1–12.
- [23] Mee Young Park and Trevor Hastie. 2007. L1-regularization path algorithm for generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* (2007).
- [24] Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang. 2016. Product-based neural networks for user response prediction. In *ICDM*.
- [25] Rajesh Ranganath, Adler Perotte, Noémie Elhadad, and David Blei. 2016. Deep survival analysis. *arXiv preprint arXiv:1608.02158* (2016).
- [26] Kan Ren, Weinan Zhang, Ke Chang, Yifei Rong, Yong Yu, and Jun Wang. 2018. Bidding Machine: Learning to Bid for Directly Optimizing Profits in Display Advertising. *TKDE* (2018).
- [27] Kan Ren, Weinan Zhang, Yifei Rong, Haifeng Zhang, Yong Yu, and Jun Wang. 2016. User response learning for directly optimizing campaign performance in display advertising. In *CIKM*.
- [28] Jun Wang, Weinan Zhang, and Shuai Yuan. 2017. Display Advertising with Real-Time Bidding (RTB) and Behavioural Targeting. *Now Publisher* (2017).
- [29] Yuchen Wang, Kan Ren, Weinan Zhang, Jun Wang, and Yong Yu. 2016. Functional bid landscape forecasting for display advertising. In *ECML-PKDD*.
- [30] Sai Wu, Weichao Ren, Chengchao Yu, Gang Chen, Dongxiang Zhang, and Jingbo Zhu. 2016. Personal recommendation using deep recurrent neural networks in NetEase. In *ICDE*.
- [31] Wush Wu, Mi-Yen Yeh, and Ming-Syan Chen. 2018. Deep Censored Learning of the Winning Price in the Real Time Bidding. In *KDD*.
- [32] Wush Chi-Hsuan Wu, Mi-Yen Yeh, and Ming-Syan Chen. 2015. Predicting Winning Price in Real Time Bidding with Censored Data. In *KDD*.
- [33] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google’s neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144* (2016).
- [34] Zeping Yu and Gongshen Liu. 2018. Sliced Recurrent Neural Network. *COLING*.
- [35] Shuai Yuan, Jun Wang, Bowei Chen, Peter Mason, and Sam Seljan. 2014. An empirical study of reserve price optimisation in real-time bidding. In *KDD*.
- [36] Shuai Yuan, Jun Wang, and Xiaoxue Zhao. 2013. Real-time bidding for online advertising: measurement and analysis. In *ADKDD*.
- [37] Hao Helen Zhang and Wenbin Lu. 2007. Adaptive Lasso for Cox’s proportional hazards model. *Biometrika* 94, 3 (2007), 691–703.
- [38] Saizheng Zhang, Yuhuai Wu, Tong Che, Zhouhan Lin, Roland Memisevic, Ruslan R Salakhutdinov, and Yoshua Bengio. 2016. Architectural complexity measures of recurrent neural networks. In *NIPS*.
- [39] Weinan Zhang, Shuai Yuan, and Jun Wang. 2014. Optimal real-time bidding for display advertising. In *KDD*.
- [40] Weinan Zhang, Shuai Yuan, and Jun Wang. 2014. Real-Time Bidding Benchmarking with iPinYou Dataset. *arXiv:1407.7073* (2014).
- [41] Weinan Zhang, Tianxiong Zhou, Jun Wang, and Jian Xu. 2016. Bid-aware Gradient Descent for Unbiased Learning with Censored Data in Display Advertising. In *KDD*.
- [42] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Deep Interest Evolution Network for Click-Through Rate Prediction. *AAAI*.
- [43] W. Y. Zhu, W. Y. Shih, Y. H. Lee, W. C. Peng, and J. L. Huang. 2017. A gamma-based regression for winning price estimation in real-time bidding advertising. In *2017 IEEE International Conference on Big Data (Big Data)*. 1610–1619.

A MODEL IMPLEMENTATIONS OF DLF

In this section, we describe the details of the implemented architecture of the proposed model (DLF).

Recall that, each sample is a triple (\mathbf{x}, z, b_l) where $\mathbf{x} \in \mathbb{R}^K$ is the K -dimensional vector representing the feature of the sample, $z \in \mathbb{N}^+$ is an integer of the true event time and $b_l \in \mathbb{N}^+$ is a integer of the proposed bid price.

As is illustrated in Figure 3, the input to our DLF model is the triple (\mathbf{x}, z, b_l) . For each recurrent unit, we feed the input as (\mathbf{x}, b_j) where $j \in [1, l]$ is an integer representing the price interval of the current unit.

Note that \mathbf{x} is a multi-hot encoded feature vector including a series of one-hot encoded features, and we first put it through an embedding layer as

$$\mathbf{e} = \text{embed}(\mathbf{x}).$$

After getting the embedding vector \mathbf{e} , we concatenate the input vector \mathbf{e} and the bid b_j as

$$\mathbf{v}_j = \text{concat}(\mathbf{e}, b_j).$$

Specifically, we implement Long Short-Term Memory (LSTM) [10] as the recurrent unit $f_\theta(\mathbf{x}, b_j \mid \mathbf{r}_{j-1})$ as

$$\begin{aligned} \mathbf{f}_j &= \sigma(\mathbf{W}_f \cdot \mathbf{v}_j + \mathbf{b}_f) \\ \mathbf{i}_j &= \sigma(\mathbf{W}_i \cdot \mathbf{v}_j + \mathbf{b}_i) \\ \mathbf{o}_j &= \sigma(\mathbf{W}_o \cdot \mathbf{v}_j + \mathbf{b}_o) \\ \mathbf{r}_j &= \mathbf{f}_j \odot \mathbf{r}_{j-1} + \mathbf{i}_j \odot \tanh(\mathbf{W}_s \cdot \mathbf{v}_j + \mathbf{b}_s) \\ \mathbf{l}_j &= \mathbf{o}_j \odot \tanh(\mathbf{r}_j). \end{aligned} \tag{17}$$

Here \mathbf{r}_j is the hidden state vector of the j th recurrent unit. After we get the output of each unit \mathbf{l}_j , a fully connected layer with sigmoid activation function predicts the hazard rate as is described in Eq. (7) of our main paper as

$$h_j = \sigma(\mathbf{W}_h \cdot \mathbf{l}_j + \mathbf{b}_h).$$

Then we may calculate the market price probability $p(z)$ w.r.t. market price z , the winning rate $W(b)$ and the losing rate $S(b)$ at the proposed bid price b as Eqs. (8) and (9) of our main paper. More details can be referred to our published code and the link of code repository with datasets is <https://github.com/rk2900/DLF>.

B EXPERIMENT CONFIGURATIONS

B.1 Hyperparameter

All the models are trained until convergence and we consider learning rate from $\{1 \times 10^{-4}, 5 \times 10^{-3}, 1 \times 10^{-3}\}$. The value of α is tuned to 0.25. Batch size is fixed on 128 and embedding dimension is 32. All the deep learning models take input features and feed through an embedding layer for the subsequent feedforward calculation. The hyperparameters of each model are tuned and the best performances have been reported.

B.2 Hardware

The models are trained under the same hardware settings with an Intel(R) Core(TM) i7-6900K CPU processor, an NVIDIA GeForce GTX 1080Ti GPU processor and 128 GB memory. The training time of each compared model is less than ten hours (as reported from the slowest training model MTLA) on each dataset.