Applying Deep Learning Based Probabilistic Forecasting to Food Preparation Time for On-Demand Delivery Service

Chengliang Gao
Meituan
gaochengliang@meituan.com

Fan Zhang
Meituan
zhangfan59@meituan.com

Yue Zhou
Meituan
zhouyue08@meituan.com

Ronggen Feng
Meituan
fengronggen@meituan.com

Qiang Ru
Meituan
ruqiang@meituan.com

Kaigui Bian
Peking University
bkg@pku.edu.cn

Renqing He
Meituan
herenqing@meituan.com

Zhizhao Sun
Meituan
sunzhizhao@meituan.com

ABSTRACT

On-demand food delivery service has widely served people's daily demands worldwide, e.g., customers place over 40 million online orders in Meituan food delivery platform per day in Q3 of 2021. Predicting the food preparation time (FPT) of each order accurately is very significant for the courier and customer experience over the platform. However, there are two challenges, namely incomplete label and huge uncertainty in FPT data, to make the prediction of FPT in practice. In this paper, we apply probabilistic forecasting to FPT for the first time and propose a non-parametric method based on deep learning. Apart from the data with precise label of FPT, we make full use of the lower/upper bound of orders without precise label, during feature extraction and model construction. A number of categories of meaningful features are extracted based on the detailed data analysis to produce sharp probability distribution. For probabilistic forecasting, we propose S-QL and prove its relationship with S-CRPS for interval-censored data for the first time, which serves the quantile discretization of S-CRPS and optimization for the constructed neural network model. Extensive offline experiments over the large-scale real-world dataset, and online A/B test both demonstrate the effectiveness of our proposed method.

CCS CONCEPTS

• **Mathematics of computing** → **Probabilistic inference problems**; • **Information systems** → **Data mining**; • **Computing methodologies** → **Neural networks**.

KEYWORDS

On-demand food delivery; Food preparation time; Probabilistic forecasting; Deep learning

KDD '22, August 14–18, 2022, Washington, DC, USA

ACM ISBN 978-1-4503-9385-0/22/08…$15.00
https://doi.org/10.1145/3534678.3539035

ACM Reference Format:
Chengliang Gao, Fan Zhang, Yue Zhou, Ronggen Feng, Qiang Ru, Kaigui Bian, Renqing He, and Zhizhao Sun. 2022. Applying Deep Learning Based Probabilistic Forecasting to Food Preparation Time for On-Demand Delivery Service. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3534678.3539035

1 INTRODUCTION

Fueled by the rapid development of cellular network, mobile Internet and e-commerce, online food ordering and delivery service has changed people's daily life. On-demand food delivery platforms, such as Uber Eats, DoorDash, Ele.me and Meituan[1] have served millions of customers every day. For instance, in China, customers place more than 40 million online orders in Meituan food delivery platform per day in Q3 of 2021[2].

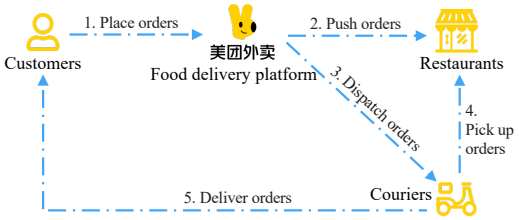


Figure 1: A typical procedure of food ordering and delivery.

A typical food ordering and delivery procedure in Meituan food delivery platform is shown in Fig. 1. After the customer places an order, the platform pushes the order to the restaurant, and dispatches it to the courier at an appropriate time. The restaurant will prepare the food, and the courier will pick up the food and deliver the order to the customer. As a core system module, the prediction of food preparation time (FPT), a.k.a. the time from the restaurant receiving an order to completing food preparation, of each order is used to manage order dispatching and suggest the courier when to pick up the food. On average, FPT accounts for about 30% of the order

[1] https://waimai.meituan.com
[2] http://media-meituan.todayir.com/20211126173203894710025711_tc.pdf

fulfillment cycle time in Meituan food delivery platform, which is quite considerable. Same as Uber Eats[3], the platform expects that couriers arrive at restaurants the moment food is ready. From the couriers' point of view, arriving at the restaurant too early means a long waiting time, and will degrade their delivery experience as couriers have other orders that will be delayed due to waiting too long for one order. Meanwhile, customers would not like their food to be picked up and delivered too late considering taste of the ordered dishes. Therefore, predicting FPT of each order accurately is very significant to courier and customer experience.

FPT prediction is a typical regression problem, but with two following challenges which are pretty common in the O2O (Online To Offline) business. (1) **Incomplete label of FPT**. Restaurants usually have no manpower to report the time when food is prepared. The platform can collect the precise FPT of an order only when the restaurant reports it, or when the courier waits for over a certain time in the restaurant and the pickup time can be adopted. As a result, only a portion of orders (no more than 70%) have precise FPT values on the platform, called precise-label (PL) orders in this paper. Remaining non-PL orders are interval-censored data, and we can only get the lower and upper bounds of FPT, called range-label (RL) orders. For each RL order, the courier's pickup time is treated as the upper bound of FPT, as the food must be prepared already before being picked up. Meanwhile, we can get the lower bound of an RL order as well, when the food must have not been prepared. If the courier has ever arrived at the restaurant, but fails to pick up the order and leaves due to long wait, the lower bound is set as the time when the courier leaves (otherwise as zero). Zhu et al. give similar discussion on predicting order fulfillment cycle time in Ele.me as well [24]. Apart from the data with precise label, the lower/upper bound of RL orders also contains rich information valuable for FPT prediction, and researches on regression for interval-censored data [1, 16, 21] inspire us a lot to process RL orders reasonably. (2) **Huge uncertainty in FPT data**. Three key factors that affect FPT are unavailable, which leads to uncertainty in FPT prediction: (a) Supply of a restaurant (We have no access to the detailed information of a restaurant's kitchen, such as the number of chefs, cooking utensils and ingredients); (b) Procedures of food preparation (Each restaurant may have its own food preparation procedure, such as preparation sequence, parallelism of orders and dishes in the same order); (c) Situation of "dine in" (Dishes people order offline in the restaurant will greatly affect the FPT of online orders). Hence, aforementioned uncertainty severely limits the accuracy of prediction in deterministic spot forecasting (SF).

In this case, we are determined to apply probabilistic forecasting (PF) to FPT, aiming to produce more information about uncertainty for better decision making. As known, PF is widely adopted in scenarios with strong uncertainty, such as earthquake, wind power and electricity price [13, 19, 23] and problems with interval-censored data, e.g., survival prediction [2], and is recognized as an essential ingredient of optimal decision making [5]. From a predictive standpoint, regression should aim to produce calibrated and sharp conditional distribution of the response variable given a set of explanatory variables [11]. Calibration refers to the statistical consistency of predicted distributions and observations. For example, it should actually rain approximately 3 of 10 times among all days with a 30% forecast of raining. Sharpness characterizes concentration of the predicted distribution, as sharp distributions provide more information for decision making. SF focuses on the conditional expectation/median of the response variable, but it treats higher moments as fixed values, which is usually not valid in real-world situation [18]. In this paper, we apply PF to FPT for the first time and propose a non-parametric method based on deep learning, which predicts the quantiles for a series of cumulative densities. The model is optimized by minimizing S-CRPS (Survival Continuous Ranked Probability Score) proposed to handle interval-censored data in [2], which is generalized from the widely adopted CRPS. For implementation, quantile discretization is adopted to avoid complicated calculation of the integral in S-CRPS.

The contributions can be summarized as follows: (1) We define the FPT-PF (Food Preparation Time Probabilistic Forecasting) task, aiming to predict calibrated probability distribution of FPT as sharp as possible. The FPT-PF task is based on the dataset composed of PL and RL orders, which is introduced in Sec. 3; (2) For RL orders, we follow the idea usually adopted in regression methods for interval-censored data. Specifically, we make no assumptions about the unknown precise FPT, and make full use of the lower/upper bound during feature extraction, model construction and evaluation. The aggregation method for historical orders proposed in Sec. 4.1 can be referred by other similar scenarios; (3) Detailed data analysis is conducted to characterize what factors will affect the FPT of an order in Sec. 4.2. Based on the analysis, we extract a number of categories of expressive features to maximize the sharpness; (4) We propose S-QL (S-Quantile Loss) generalized from the common quantile loss. Based on the proved relationship between S-QL and S-CRPS in Sec. 4.3, the quantile discretization of S-CRPS for interval-censored data can be performed for model optimization. PF method explained in Sec. 4.4 can be well adapted to deep learning, and easily migrated to other problems; (5) Offline experiments over the large-scale real-world dataset collected from Meituan food delivery platform illustrate the effectiveness of our proposed method in Sec. 5. Besides, the online A/B test also demonstrates the gain brought to courier and customer experience.

2 RELATED WORK

In this section, we will introduce existing literatures related to the two challenges for FPT prediction mentioned in Sec. 1, including regression for interval-censored data and probabilistic forecasting.

2.1 Regression for Interval Censored Data

In the survival analysis literature, regression methods have been developed to predict time to events given interval-censored data. Mainly adopting PF, specific methods are mostly optimized via MLE (Maximum Likelihood Estimation), which maximizes the overall possibility of event occurring in the interval, and does not presume the precise time at all, such as the non-parametric MLE [21], proportional hazards model [1] and parametric models in [16]. Following the same idea, the S-CRPS is generalized from CRPS considering its superiority for PF compared with MLE [2]. Similarly, we also make no assumptions about the unknown precise FPT, and make full use of the lower/upper bound on processing RL orders in this paper.

[3]https://www.infoq.com/articles/uber-eats-time-predictions

2.2 Probabilistic Forecasting

Bayesian vs. non-Bayesian approach. Approaches of PF can broadly be distinguished as Bayesian and non-Bayesian. Bayesian approaches assume a prior and perform posterior inference based on the dataset [3, 14]. Non-Bayesian approaches which model the uncertainty straightly are more flexible in practice. We focus on non-Bayesian approaches in the paper, which our proposed method belongs to.

Parametric vs. non-parametric approach. Parametric and non-parametric approaches are two main techniques to construct probability distribution. Parametric approaches assume the shape of predictive density, e.g. Gaussian or Gamma, and focus on predicting parameters of the distribution. XGBoostLSS is proposed as an extension of XGBoost, which models all moments of a parametric distribution [18]. In addition, Duan et al. combine boosting algorithm with the natural gradient to estimate the parameters of presumed distribution [4]. Based on deep learning, Salinas et al. propose DeepAR, which parametrizes the Gaussian likelihood on training an autoregressive recurrent neural network model [20]. Without any assumption of the distribution shape, non-parametric approaches usually estimate finite points of the distribution. Quantile regression is a typical non-parametric method applying the quantile loss to common SF regression methods. Several quantile regression models are usually trained independently to calculate the distribution, which may lead to crossing quantiles [23]. Hasson et al. design a level-set PF approach, which groups the data points to different partitions and calculates the distribution within each partition [10]. Besides, the dropout-based ensemble method [15] belongs to non-parametric approaches as well. Generally, non-parametric approaches are more flexible compared with parametric approaches, but only have a finite number of predicted points rather than the distribution function. However, the effectiveness of parametric approaches depends on if the presumed distribution matches the actual one extremely.

3 PROBLEM FORMULATION

Table 1: The notations.

Notation	Defination
o	An order
D^o	Set of dishes in order o
m^o	Number of unique dishes in order o
d_i^o	i-th unique dish in order o
num_i^o	Number of dish d_i^o
t^o	Food preparation time of order o
X^o	All features for order o
X_c^o	Context features for order o
X_r^o	Restaurant features for order o
X_d^o	Dish features for dish d in order o
y^o	Precise label of FPT for order o
l^o	Lower bound of FPT for order o
u^o	Upper bound of FPT for order o
c^o	Identifier of order o (0: precise-label, 1: range-label)

As mentioned in Sec. 1, the FPT t^o of an order o refers to the time from the restaurant receiving the order to completing the dishes in the order. The order contains m^o unique dishes, denoted as $D^o = \{(d_i^o, num_i^o)|i = 1, 2, ..., m^o\}$, where num_i^o is the number of dish d_i^o. t^o is made up of the queuing time of waiting for previous orders' completion and the cooking time of all the dishes in the current order. We extract a real-world dataset from Meituan food delivery platform, containing over 442.19 million orders. The information carried by order o can be denoted as a tuple $(X^o, y^o, l^o, u^o, c^o)$, where X^o denotes the set of features, including context features X_c^o, restaurant features X_r^o and features X_d^o for each dish $d \in D^o$. Details of the features will be introduced in Sec. 4.2. As mentioned above, precise FPT of PL orders can be collected, where we set $c^o = 0$ and y^o is the precise label, meaning $t^o = y^o$. For RL orders, we can only get lower bound l^o and upper bound u^o of the FPT, meaning $l^o < t^o \leq u^o$, and we set $c^o = 1$ in this case.

The FPT-PF task aims to model FPT of each order in the form of probability distribution, treating FPT as a random variable. Specifically, given the features X^o, the CDF (Cumulative Distribution Function) or PDF (Probability Density Function) should be predicted for parametric approaches, while finite discrete points in the CDF or PDF are predicted for non-parametric approaches. Definition of notations related to order o can be found in Tab. 1 and we will omit superscripts o for succinctness without ambiguity.

4 METHOD

As mentioned in Sec. 1, we propose a non-parametric method based on deep learning to tackle the FPT-PF task, and the details will be introduced in this section.

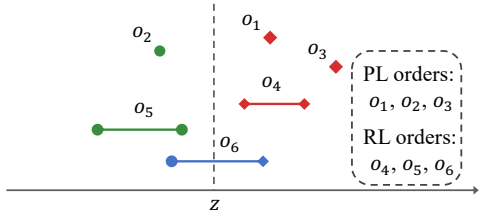
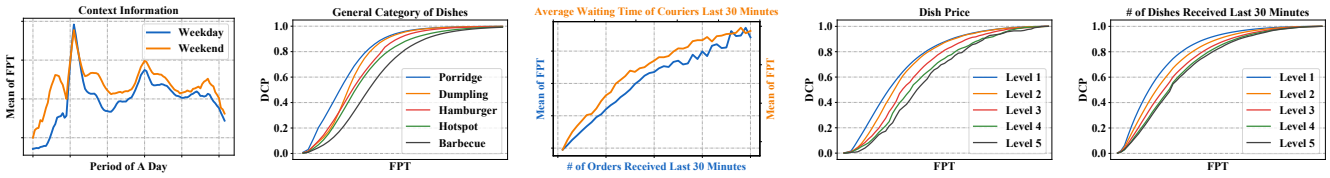


Figure 2: An example of DCP calculation. FPT of order o_1, o_3, o_4 is longer than z, o_2, o_5 is shorter, and o_6 is uncertain, so dcp_z can be calculated as $2/(2+3) = 0.4$.

4.1 Aggregation of FPT for Different Orders

In practice, we need to calculate statistics of the FPT from historical orders with the purpose of feature analysis and extraction. For common scenarios with all precise labels, the mean and variance of labels from the data are usually adopted. However, recall that we have two types of orders, i.e. PL and RL orders, mentioned in Sec. 1. Calculation of the mean/variance for RL orders is unavailable, and only considering the PL orders is obviously unreasonable. Considering PL and RL orders together, we design the DCP (Discrete Cumulative Probabilities) to aggregate FPT of a set of orders in the paper. DCP for an order set is denoted as $\text{DCP} = \{\text{dcp}_z|z \in Z\}$, where Z is a series of equidistant discrete time points $\{z_i = i \times \Delta z|1 \leq i \leq n_z\}$. Δz can be set according to statistical precision and $n_z \times \Delta z$ should be a relatively large value that FPT almost never exceeds. dcp_z is calculated as the fraction of orders whose FPT does not exceed



(a) Context. (b) General category. (c) Supply & Demand (restaurant). (d) Dish price. (e) Supply & Demand (dish).

Figure 3: Feature analysis from over 2500 restaurants with top number of orders sampled from different categories.

z, and the calculation over PL orders is intuitive. For RL orders, the calculation can only be done when $z <= l$ or $z >= u$. For the case that $l < z < u$, it is uncertain that whether the FPT is longer than z or not, so we make no assumptions and just ignore it. An example of the calculation for dcp_z is illustrated in Fig. 2 to help understand. Generally, DCP makes no assumptions about the unknown precise FPT, and make full use of the lower/upper bound for RL orders. In addition, we calculate the approximate mean as $\sum_{i=1}^{n_z} (\text{dcp}_{z_i} - \text{dcp}_{z_{i-1}}) \times (z_i + z_{i-1})/2$ from DCP, where we set $z_0 = 0$ and $\text{DCP}_{z_0} = 0$. The approximate mean can characterize the average FPT roughly, but is not numerically precise, which is only the assist for data analysis. We consider DCP is a reasonable way of data aggregation, and can be applied to other similar scenarios with interval-censored data.

Table 2: The various categories of features.

Category		Feature
Context info		Period of a day (per 10 mins) Day of a week
Restaurant	**Basic info**	ID of the restaurant General category of dishes sold
	Real-time supply and demand info	# of order received last x mins Waiting time of couriers last x mins
	Historical statistics of different granularities	# of order received Waiting time of couriers DCP of orders
Each dish in the order	**Basic info**	# and price
	Real-time supply and demand info	# of the dish in the orders received last x mins
	Historical statistics of different granularities	DCP of orders including the dish

4.2 Feature Extraction

Meaningful features are as significant to PF as SF to produce sharp distributions, which provide more information for decision making than unsharp ones. Various categories of features we derive are concluded in Tab. 2, and the analysis will be given in the following parts, illustrated in Fig. 3.

4.2.1 Context features. For on-demand food delivery service, there are obvious time periodicities. We calculate the approximate mean of FPT from orders placed during different periods of a day at weekday/weekend, and plot the results in Fig. 3(a). From the figure, there are two obvious peaks in each curve, as people mainly place orders during noon and evening. During the peak time, restaurants need more time to prepare the food due to longer queuing time. Apart from the periods, FPT changes among days of a week, especially for weekdays and weekends. At weekend, more orders are placed by the customers, especially during non-peak time. Comparison between the two curves confirms the phenomenon. Based on the above analysis, we extract two context features, i.e. period of a day (10 minutes per period) and day of a week, to capture the time periodicity for FPT prediction.

4.2.2 Restaurant features. FPT may change a lot among different restaurants as explained in Sec. 1. We build several categories of related features to depict the characteristics of each restaurant.

Basic information. We use the identifier of a restaurant to capture its uniqueness, which is intuitive. In addition, each restaurant registers a general category of dishes it mainly deals in, such as snack, dessert, dumplings and hotspot. The cooking procedure and time change a lot among different categories. We choose several typical categories of restaurants and calculate the DCP of orders placed in each category of restaurants. In Fig. 3(b), we observe that the FPT increases from porridge, dumpling, hamburger, hotspot to barbecue gradually. As a result, we consider the general category of dishes sold by the restaurant is an effective feature.

Real-time supply and demand information. For FPT of an order, the queuing time of waiting for previous orders' completion is considerable. The real-time supply and demand state of a restaurant has a great effect on orders' FPT, but is difficult to fully characterize as explained in Sec. 1. We use the number of orders received by a restaurant in the past periods (10/20/30 minutes adopted) to capture the real-time customer demand to some degree. In addition, the average waiting time of couriers in a restaurant is also calculated to reflect the supply and demand state. We analyze how FPT of the current order changes along with the real-time demand of the restaurant increasing. Fig. 3(c) shows the results, and we observe that FPT increases obviously with more orders received and longer waiting time in the past 30 minutes.

Historical statistics of different granularities. Historical statistics can usually bring obvious improvement for regression models in practice. Apart from the statistics of received orders' number and couriers' waiting time which are identical to real-time supply and demand information, we also calculate the historical DCP of orders placed in the restaurant. To capture the information in different granularities of time, we aggregate the orders placed in the past two weeks, the same weekday of the past four weeks and the current/next period each day in the past two weeks.

4.2.3 Dish features. Recall that an order is made up of several unique dishes with respective number, and the restaurant should prepare all the dishes to finish it. Apart from the general category

of dishes in restaurant features, we introduce several categories of fine-grained features related to each dish in the order to enhance the expression ability further.

Basic information. We use the number and price of each unique dish as the basic features. It is intuitive that the number of each dish has an effect on the FPT. Generally, the same dish can be cooked together, but the time can increase suddenly when the dishes can not be prepared in one go. The price of a dish is also meaningful, as higher price often means more complicated cooking procedure and longer time. Considering the FPT of each single dish is unavailable, we calculate the DCP of orders containing the dish with different price levels. The results in Fig. 3(d) meet our intuition, where larger level denotes higher price of the dish.

Real-time supply and demand information. Similar to restaurant related real-time supply and demand features mentioned above, we further count the number of the same dish in the orders received by the restaurant in past periods, as same dishes can be cooked together or maybe bring long-time queuing. For a specific order, we calculate the total number of all the same dishes in the orders received by the restaurant last 30 minutes, and plot the DCP curves with different levels in Fig. 3(e). From the results, we observe that FPT of an order increases along with the demand for same dishes increasing.

Historical statistics of different granularities. As explained above, the FPT of each single dish is unavailable. We thus calculate DCP of the orders including the current dish as the dish related features in the same granularities as the restaurant features.

4.3 S-CRPS and S-Quantile Loss

As mentioned in Sec. 1, we optimize the model through minimizing S-CRPS, and quantile discretization is adopted to simplify calculation. In this section, S-CRPS will be explained in detail. In addition, we propose S-QL and prove its relationship with S-CRPS for the first time, which serves the quantile discretization of S-CRPS for interval-censored data.

A scoring rule $\mathcal{S}$ takes a probability distribution P and one observation y as inputs, and produces a score $\mathcal{S}(P, y)$ to the forecast such that the actual distribution of the outcomes gets the best score in expectation [6]. The scoring rule is considered to be proper, when

$$\mathbb{E}_{y\sim\tilde{P}}[\mathcal{S}(\tilde{P}, y)] \leq \mathbb{E}_{y\sim\tilde{P}}[\mathcal{S}(P, y)] \quad \forall P, \tag{1}$$

where $\tilde{P}$ is the actual distribution of the observed y. Proper scoring rules encourage the model to predict calibrated distributions, naturally used as loss functions for model optimization.

CRPS proves to be a proper scoring rule [7], which is widely adopted for PF owing to its robustness compared with MLE. Two well-known equivalent forms of CRPS are as follows [8],

$$\mathrm{CRPS}(F, y) = \int_{-\infty}^{y} F(x)^2 \mathrm{d}x + \int_{y}^{+\infty} (1 - F(x))^2 \mathrm{d}x \tag{2}$$

$$= 2\int_0^1 \mathrm{QL}_q(F, y)\mathrm{d}q, \tag{3}$$

where F denotes the CDF, and y is an observation. In Equ. 2, minimizing CRPS aims to decrease $F(x)$ when $x < y$, and increase $F(x)$ when $x > y$. Ideally, CRPS can decrease to 0, when the probability density is concentrated on y completely. Equ. 3 creates a direct link of CRPS to quantile loss. Quantile loss is usually used as the loss function in quantile regression and defined as follows,

$$\mathrm{QL}_q(F, y) = \begin{cases} (y - F^{-1}(q)) \cdot q, & F^{-1}(q) < y \\ (F^{-1}(q) - y) \cdot (1 - q), & F^{-1}(q) \geq y \end{cases}, \tag{4}$$

where F^{-1} denotes the inverse function of CDF, and $F^{-1}(q)$ calculates the q-quantile. For implementation, the discretization of Equ. 3 is usually adopted to avoid complicated calculation of integral [8, 19], e.g., replacing the integral in Equ. 3 by the sum over QL_q, $q = 0.01, ..., 0.99$, named quantile discretization in this paper.

Avati et al. propose S-CRPS, generalized from CRPS to handle the interval-censored data. They argue that S-CRPS is proper and can produce sharper distributions compared with MLE [2]. For the FPT-PF task, S-CRPS is same with CRPS for PL orders, and is defined as follows for an RL order,

$$\text{S-CRPS}(F, l, u) = \int_{-\infty}^{l} F(x)^2 \mathrm{d}x + \int_{u}^{+\infty} (1 - F(x))^2 \mathrm{d}x. \tag{5}$$

For RL orders, S-CRPS makes no assumptions about the unknown precise FPT, and make full use of the lower/upper bound, as mentioned in Sec. 2.1. To our best knowledge, there is no work giving the quantile discretization of S-CRPS for interval-censored data, which is significant for FPT-PF. For the first time, we propose the S-QL generalized from the common quantile loss as follows,

$$\text{S-QL}_q(F, l, u) = \begin{cases} (l - F^{-1}(q)) \cdot q, & F^{-1}(q) < l \\ 0, & l \leq F^{-1}(q) < u \\ (F^{-1}(q) - u) \cdot (1 - q), & F^{-1}(q) \geq u \end{cases}. \tag{6}$$

In addition, we prove that Equ. 5 is equivalent to the following form in Appendix A,

$$\text{S-CRPS}(F, l, u) = 2\int_0^1 \text{S-QL}_q(F, l, u)\mathrm{d}q, \tag{7}$$

based on which the quantile discretization of S-CRPS can be performed for model optimization.

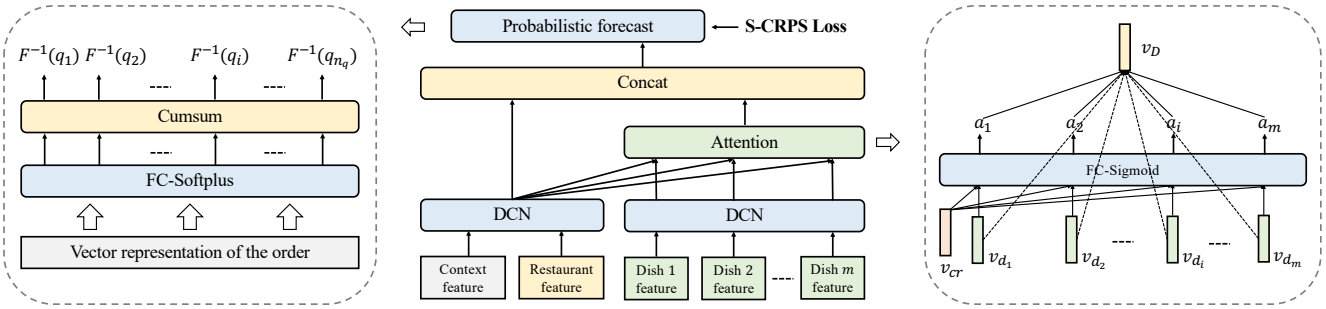
4.4 Model

We propose a non-parametric model for the FPT-PF task based on deep learning, whose overall architecture is illustrated in Fig. 4. At first, we preprocess the extracted context, restaurant and dish features. Then we adopt Attention to capture the various contributions of each dish in the order to the FPT. Finally, we predict the quantiles of FPT for a series of cumulative densities and optimize the model by minimizing quantile discretization of S-CRPS.

4.4.1 Feature preprocessing. We encode the context and restaurant features together, and then the features of each dish respectively, extracted in Sec. 4.2. Fully considering the low-order and high-order feature interactions, we adopt the DCN (Deep & Cross Network) [22] to encode the features as follows,

$$\begin{aligned} v_{cr} &= \mathrm{DCN}(X_c; X_r), \\ v_d &= \mathrm{DCN}(X_d), \quad d \in D, \end{aligned} \tag{8}$$

where X_c denotes context features, X_r denotes restaurant features and X_d denotes the features of dish d. The calculation details of DCN can be found in Appendix B.1. Through preprocessing, we get the k-dimensional vector representation of context and restaurant features as v_{cr}, features of dish d as v_d.



(a) Probabilistic forecast. (b) Architecture. (c) Attention.

Figure 4: Overall architecture of the proposed model.

4.4.2 Attention. To aggregate the vector representation of each dish in the order, we ought to consider the cooking procedure among dishes in the current order and other previous orders. The procedure is quite complicated and elusive, varying with different restaurants and time as explained in Sec. 1. For example, there is usually fixed production line for drinks like milk tea, and the parallelism of preparation is relatively high. In contrast, preparation for dishes with complicated cooking processes can be serial when supply capacity of the restaurant is insufficient. The FPT of an order mainly depends on the slowest dish in the scenarios with fully parallelism, but is the sum of preparation time for each dish when serial. As a result, the contribution of each dish to the order's FPT varies along with the context and restaurant information, which inspires us to adopt Attention [17] for aggregation, as shown in Fig. 4(c). The calculation is as follows,

$$a_i = \sigma(w_{cr} \cdot v_{cr} + w_d \cdot v_{d_i} + b), \tag{9}$$

where w_{cr} and w_d are k-dimensional parameters, b is a bias scalar and $\sigma(\cdot)$ is the Sigmoid function, $\sigma(x) = 1/(1+e^{-x})$. Sigmoid is more suitable for our problem to capture the parallelism of dish preparation, rather than Softmax, which is usually adopted to calculate the weighted average in common Attention methods. $a_i \in (0,1)$ denotes the contribution weight of dish d_i and we aggregate the dishes by the weighted sum,

$$v_D = \sum_{i=1}^{m} a_i \cdot v_{d_i}, \tag{10}$$

where v_D denotes the vector representation of the aggregation of all dishes in the order. Finally, we concatenate v_{cr} and v_D to get the vector representation v of the order and finish the feature encoding.

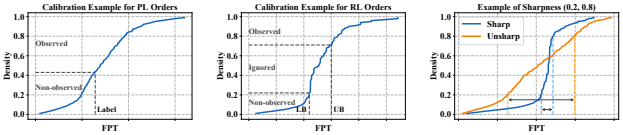
4.4.3 Probability forecast. As explained in Sec. 2, parametric and non-parametric approaches are two main techniques to construct probability distributions. Making no assumptions about the shape of distribution, non-parametric approaches are more flexible in practical scenarios. For implementation, the model is adapted to the quantile discretization of S-CRPS mentioned in Sec. 4.3. We select a set of equidistant discrete cumulative densities $Q = \{q_i = i \times \Delta q | 1 \le i \le n_q\}$, and predict the q-quantile for each $q \in Q$. The detailed structure of the neural network is shown in Fig. 4(a). To solve the problem of crossing quantiles mentioned in Sec. 2, the model must output a series of increasing values. We apply a fully connected (FC) layer with Softplus activation function, i.e. $\zeta(x) = \log(1+e^x)$, to the vector representation v of the order to get n_q positive values, and then calculate the cumulative sum as the quantiles $F^{-1}(q_i), 1 \le i \le n_q$.

4.4.4 Model Training. Based on the quantile discretization of Equ. 3 and Equ. 7, the loss function for an order is calculated as follows,

$$\mathcal{L}(\Theta) = \begin{cases} 2\Delta q \cdot \sum_{i=1}^{n_q} \mathrm{QL}_{q_i}(F, y), & c = 0 \\ 2\Delta q \cdot \sum_{i=1}^{n_q} \text{S-QL}_{q_i}(F, l, u), & c = 1 \end{cases}, \tag{11}$$

where Θ denotes the parameter set of the neural network model, and we set $\Delta q = 0.01$ and $n_q = 99$ in the paper. For model training, the Adam optimizer is adopted to minimize the loss function. In addition, we adopt Batch Normalization [12] to accelerate convergence and improve stability, which proves to be very effective.

The proposed non-parametric method can be easily migrated to other scenarios, through a simple modification of the output and loss function for SF based on deep learning. Besides, we also discuss another non-parametric PF approach based on time discretization in Appendix B.2, which has ever been adopted practically.



(a) Precise-label order. (b) Range-label order. (c) Sharpness.

Figure 5: Examples for evaluation metrics.

5 EXPERIMENTS

In this section, we evaluate and analyze the performance of our proposed method. First, we explain the evaluation framework for PF and specific metrics adopted in the paper. Then we conduct detailed offline experiments to illustrate the effectiveness. Finally, we introduce the brought gain of courier and customer experience through online A/B test.

5.1 Metrics

Evaluation for PF has more challenges than that for SF, and acknowledged target of PF is maximizing sharpness of the probability distribution, subject to calibration [23]. Uncalibrated predictions regardless of sharpness are wrong, and calibrated but non-sharp ones are correct but less useful [2]. We follow most existing researches, evaluating the performance based on calibration, sharpness and regression accuracy, i.e. scoring rules and quantile loss. Specific metrics will be given in the following parts and Fig. 5 shows three examples to help understand.

5.1.1 Calibration. Calibration refers to the statistical consistency of predicted distributions and observations as mentioned in Sec. 1. In this paper, we follow the same way used in [2] to evaluate how well calibrated a predicted distribution. We calculate the frequency of observations which do not exceed the predicted quantile for each cumulative density. For a specific cumulative density q, the q-quantile of each order is predicted by the model as introduced in Sec. 4.4. For a PL order, the order is observed when $y \leq q$-quantile, not observed otherwise. For an RL order, the order is observed when $u \leq q$-quantile, and not observed when q-quantile $\leq l$. If q-quantile $\in (l, u)$, we just ignore the order, as we can not confirm if the FPT is longer than the q-quantile or not. Fig. 5(a) and Fig. 5(b) illustrate two predicted distributions of our proposed method to help understand. Finally, the observed frequency is calculated by the division of observed number and the number of all considered orders (without the ignored ones). When the predicted distribution is well calibrated, the observed frequency for cumulative density q should be close to q, and we evaluate the cumulative density 0.1 to 0.9 in the paper. The evaluation of calibration for RL orders also keeps the principle that making no assumptions about the unknown precise FPT, and making full use of the lower/upper bound.

5.1.2 Sharpness. Sharpness characterizes concentration of the predicted distribution, as sharp distribution provides more information for decision making. Sharpness is just a single property of the predicted distribution, without consideration of the observation. We use the average width of central prediction intervals to evaluate the sharpness mentioned in [5]. Formally, the average width of interval (a, b) is calculated by b-quantile minus a-quantile of the predicted distribution. Specifically, the two central intervals $(0.4, 0.6)$ and $(0.2, 0.8)$ are evaluated in the paper. We show two predicted distributions of our proposed method and the width of interval $(0.2, 0.8)$ to help understand in Fig. 5(c).

5.1.3 Regression accuracy. Treating the observed label as a Dirac distribution centered over the FPT, CRPS characterizes the L^2 divergence between the predicted distribution and the actual one [4]. Through minimizing CRPS/S-CRPS, calibration and sharpness are optimized at the same time. As a result, S-CRPS is adopted as an important evaluation metric, which can capture accuracy of the predicted distribution relative to observations. Considering computational complexity and comparability between different approaches, we calculate the quantile discretization of S-CRPS, i.e. the loss function Equ. 11. For commonly understood regression accuracy, we consider the 0.5-Quantile Loss, i.e. $\mathrm{QL}_{0.5}$ for PL orders and S-$\mathrm{QL}_{0.5}$ for RL orders as well. The 0.5-Quantile Loss is the middle unbiased term of the loss function Equ. 11, and $\mathrm{QL}_{0.5}$ is half of the well-known MAE (Mean Absolute Error), usually used in SF for accuracy evaluation.

5.2 Offline Experiments

We conduct extensive offline experiments based on the collected large-scale real-world dataset from Meituan food delivery platform mentioned in Sec. 3. The experimental results are shown in Tab. 3 and Tab. 4. See the results in the third row, our proposed method is well calibrated, as observed frequencies are close to the corresponding cumulative densities. In addition, calibration for larger cumulative densities (0.6 to 0.9) is better than that for smaller ones (0.1 to 0.5). For sharpness and regression accuracy, the average width of interval (0.2, 0.8) is larger than that of interval (0.4, 0.6), 349.42 and 105.80 respectively, which is intuitive. Meanwhile, the S-CRPS (127.67/46.29) and 0.5-Quantile Loss (90.30/29.77) for PL/RL orders are given together.

5.2.1 Comparison with parametric approach. Explained in Sec. 2.2, parametric approaches assume the shape of predictive density. In this part, two parametric approaches are implemented, assuming FPT follows the Gaussian/Gamma distribution, and compared with our proposed non-parametric method. Gaussian distribution is often used in the natural and social sciences to represent random variables with unknown distributions. The PDF of Gaussian distribution is

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}, \tag{12}$$

where parameter μ denotes the mean/expectation and σ is the standard deviation. Gamma distribution is frequently used to model the length of waiting time, which is suitable for FPT prediction. The PDF of Gamma distribution is

$$f(x) = \frac{x^{\alpha-1} e^{-\beta x} \beta^{\alpha}}{\Gamma(\alpha)}, \tag{13}$$

where α denotes the shape parameter, β denotes the rate parameter, and $\Gamma(\cdot)$ is the gamma function. For implementation, we adopt the same model structure and just replace the final layer of the neural network in Fig. 4. The model outputs parameters of the Gaussian/Gamma distribution activated by Softplus function, and the same loss Equ. 11 is used for training. From the experimental results listed in the first two rows in Tab. 3 and Tab. 4, we observe our method is more calibrated overall except for the cumulative density 0.2 and 0.3. For sharpness and regression accuracy, our method also outperforms the Gaussian/Gamma distribution, under the settings of same features and model structure. How the assumed distribution matches actuality has a huge effect on the performance of a parametric approach, and our proposed non-parametric approaches are more flexible and can guarantee the effectiveness.

5.2.2 Comparison with spot forecasting. As explained in Sec. 1, SF methods focus on the conditional expectation/median of the response variable, treating higher moments as fixed values. In this part, we compare the regression accuracy between our proposed method and SF method. The SF model outputs a positive scalar, and is optimized through minimizing the 0.5-Quantile Loss, i.e. the middle unbiased term of the quantile discretization of S-CRPS. Considering fairness of comparison, we also keep the features and model structure to be consistent. The comparative results are shown

Table 3: Calibration (observed frequencies for cumulative density 0.1 to 0.9).

	Cumulative Density								
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Gaussian distribution	0.0619	**0.1938**	0.3309	0.4465	0.5453	0.6334	0.7153	0.7944	0.8740
Gamma distribution	0.0859	0.1927	**0.3045**	0.4144	0.5199	0.6201	0.7152	0.8072	0.8983
Our method	**0.0868**	0.2245	0.3132	**0.4137**	**0.5155**	**0.6096**	**0.7068**	**0.8048**	**0.9015**
... w/o context features	0.0915	0.2317	0.3177	0.4165	0.5149	0.6108	0.7094	0.8053	0.9004
... w/o restaurant features	0.0818	0.2063	0.2928	0.3968	0.4931	0.5930	0.6972	0.8004	0.8992
... w/o dish features	0.0945	0.2170	0.3069	0.4054	0.5068	0.6004	0.6991	0.7967	0.8970
... w/o supply and demand	0.0896	0.2255	0.3134	0.4099	0.5119	0.6097	0.7029	0.8003	0.8978
... w/o historical statistics	0.1011	0.2198	0.3133	0.4116	0.5117	0.6114	0.7074	0.8051	0.9023
(0min, 10min]	0.0900	0.2203	0.3097	0.4097	0.5147	0.6092	0.7031	0.8031	0.8995
(10min, 20min]	0.0831	0.2297	0.3180	0.4189	0.5165	0.6098	0.7118	0.8072	0.9041
(20min, +∞)	0.0579	0.2479	0.3255	0.4298	0.5227	0.6170	0.7164	0.8074	0.9080

Table 4: Sharpness and regression accuracy.

	Sharpness		S-CRPS		0.5-Quantile Loss	
	(0.2, 0.8)	(0.4, 0.6)	Precise-label	Range-label	Precise-label	Range-label
Gaussian distribution	356.38	107.28	130.61	47.06	91.64	31.52
Gamma distribution	365.49	109.65	131.32	46.55	92.25	29.98
Our method	**349.42**	**105.80**	**127.67**	**46.29**	**90.30**	**29.77**
... w/o context features	350.02	105.86	128.05	46.82	90.52	30.26
... w/o restaurant features	384.69	114.96	136.67	51.36	97.13	32.88
... w/o dish features	376.17	112.88	143.50	48.40	100.78	30.45
... w/o supply and demand	355.74	109.97	130.39	47.42	92.28	30.72
... w/o historical statistics	394.43	117.92	150.25	50.27	104.90	31.51
(0min, 10min]	305.46	93.09	111.32	41.98	78.56	27.01
(10min, 20min]	399.29	120.10	148.70	49.86	105.45	32.07
(20min, +∞)	553.09	167.06	234.23	61.55	165.46	39.23

Table 5: Regression accuracy (0.5-Quantitle Loss).

	(0min, 10min]		(10min, 20min]		(20min, +∞)		All	
	PL	RL	PL	RL	PL	RL	PL	RL
Spot forecast	81.44	27.65	110.07	32.10	171.39	**39.16**	93.89	30.10
Our method	**78.56**	**27.01**	**105.45**	**32.07**	**165.46**	39.23	**90.30**	**29.77**

in Tab. 5. Apart from the overall results, we also analyze the performance for restaurants with different levels of FPT. We divide restaurants into three groups (no more than 10mins/20mins and the remaining) according to the approximate mean of FPT for historical orders. From the results, we observe that our method outperforms the SF method 3.59/0.33 for PL/RL orders over the 0.5-Quantile Loss overall, and brings bigger improvement for restaurants with longer FPT. Through analysis, we conclude that the improvement is from the better process for RL orders of our method. For example, when the predicted value locates in the range $(l, u]$, the loss for SF method decreases to 0, and is equivalent to ignoring the sample. However, our method also optimizes other terms, other than the 0.5-Quantile Loss, which can extract more meaningful information from the data.

5.2.3 Feature ablation. To illustrate the effectiveness of each category of features extracted in Sec. 4.2, we conduct feature ablation experiments, and the results are shown in the middle five rows of Tab. 3 and Tab. 4. Obviously, all the experimental results are calibrated, as ablating features has little effect on calibration evaluated on the global dataset, but has influence in local cases. Just like a conditional distribution $\mathbb{P}(y|x)$ and its marginal distribution $\mathbb{P}(y) = \int \mathbb{P}(y|x)\mathbb{P}(x)\mathrm{d}x$ can be both calibrated evaluated over the global data, while the marginal one loses calibration when evaluated on the data x is limited. Enumerating all the local cases is impossible, so PF aims to maximize sharpness subject to calibration, as marginal distribution lacks concentration and accuracy although global calibrated. As a result, we focus on the sharpness and regression accuracy to illustrate the effectiveness of features. From Tab. 4, we observe that all the features are meaningful for FPT-PF. Context features can reflect time periodicity, but just bring slight improvements, i.e. 0.38/0.53 over S-CRPS for PL/RL orders. The reason is that the historical statistics contain the periodic information as well, which are calculated in the granularity of period and weekday. The restaurant and dish features both bring considerable improvement, 9.00/5.07 and 15.83/2.11 over S-CRPS respectively. Apart from the above three main categories of features, we also ablate the real-time supply and demand information and historical statistics respectively. From the results, real-time supply and demand information can bring the improvement of 2.72/1.13 over

S-CRPS. Historical statistics are extremely significant, which bring the biggest improvement of 22.58/3.98 over S-CRPS.

5.2.4 Evaluation on restaurants with different levels of FPT. Inspired by the analysis in Sec. 5.2.3, we are willing to evaluate the performance of our proposed method in some local cases. We choose the restaurants with different levels of FPT mentioned in Sec. 5.2.2, as the ability to distinguish different restaurants is meaningful for on-demand food delivery service. The evaluation results are listed in the last three rows of Tab. 3 and Tab. 4. Longer FPT often means larger uncertainty meanwhile, as the preparation procedure is more likely to be affected by chance factors, e.g. other orders online and dining in. As a result, there are more challenges for modeling the restaurants with longer FPT. For calibration, the predicted distributions for restaurants with different levels of FPT are all calibrated overall, except for the cumulative density 0.1/0.2 of restaurants with longest FPT. For sharpness and regression accuracy, the restaurants with shorter FPT get more sharp and accurate predictions as well. The S-CRPS achieves 111.32/41.98 for PL/RL orders in restaurants with shortest FPT, but 234.23/61.55 for restaurants with longest FPT. All the results meet our intuition, and show that our method has the ability to distinguish restaurants with different levels of FPT.

In conclusion, our proposed method not only provides calibrated distribution as sharp as possible for better decision making, but also produces more accurate predictions compared with SF method.

5.3 Online A/B Test

Motivated by the encouraging offline evaluation results, we deploy the proposed method in Meituan food delivery platform and test its performance online. We conduct the online A/B test, and each experimental city is split into grids in 0.5/0.5 ratio independently. Compared with the original online SF model trained only with PL orders, which is based on DeepFM [9], the MAPE (Mean Absolute Percentage Error) of our method based on 0.5-quantile decreases 0.2619 evaluated over PL orders. The well processing for RL orders, extracted features and constructed model both contribute to the improvement. Apart from the accuracy improvement, our proposed method also provides more information for better decision making, which reduces 2.17%∼4.57% of the couriers' waiting time during picking up food and gains the 0.14∼0.23pp (percentage point) improvement for order punctuality. In a word, our proposed method brings the gain of courier and customer experience jointly.

6 CONCLUSION

This paper proposes a non-parametric PF method based on deep learning to tackle the FPT-PF task. Apart from data with precise label, we make full use of the lower/upper bound of RL orders, designing DCP for feature analysis/extraction and the quantile discretization of S-CRPS for model optimization specifically. Detailed data analysis is conducted to build meaningful features and the relationship between S-QL and S-CRPS for RL orders is first proposed and proved in the paper. The offline experiments show the effectiveness compared with two parametric approaches and a SF method, and the online A/B test illustrates the gain of courier and customer experience brought by the proposed method as well.

ACKNOWLEDGMENTS

This work is supported by Meituan, and partially supported by National Key Research and Development Program No. 2020YFB2103801, and NSFC No. 62032003.

REFERENCES

[1] Clifford Anderson-Bergman. 2017. icenReg: regression models for interval censored data in R. *Journal of Statistical Software* 81, 1 (2017), 1–23.
[2] Anand Avati, Tony Duan, Sharon Zhou, Kenneth Jung, Nigam H Shah, and Andrew Y Ng. 2020. Countdown regression: sharp and calibrated survival predictions. In *Uncertainty in Artificial Intelligence*. PMLR, 145–155.
[3] Hugh A Chipman, Edward I George, and Robert E McCulloch. 2010. BART: Bayesian additive regression trees. *The Annals of Applied Statistics* 4, 1 (2010), 266–298.
[4] Tony Duan, Avati Anand, Daisy Yi Ding, Khanh K Thai, Sanjay Basu, Andrew Ng, and Alejandro Schuler. 2020. Ngboost: Natural gradient boosting for probabilistic prediction. In *International Conference on Machine Learning*. PMLR, 2690–2700.
[5] Tilmann Gneiting and Matthias Katzfuss. 2014. Probabilistic forecasting. *Annual Review of Statistics and Its Application* 1 (2014), 125–151.
[6] Tilmann Gneiting and Adrian E Raftery. 2007. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association* 102, 477 (2007), 359–378.
[7] Tilmann Gneiting, Adrian E Raftery, Anton H Westveld III, and Tom Goldman. 2005. Calibrated probabilistic forecasting using ensemble model output statistics and minimum CRPS estimation. *Monthly Weather Review* 133, 5 (2005), 1098–1118.
[8] Tilmann Gneiting and Roopesh Ranjan. 2011. Comparing density forecasts using threshold-and quantile-weighted scoring rules. *Journal of Business & Economic Statistics* 29, 3 (2011), 411–422.
[9] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. 1725–1731.
[10] Hilaf Hasson, Bernie Wang, Tim Januschowski, and Jan Gasthaus. 2021. Probabilistic Forecasting: A Level-Set Approach. *Advances in Neural Information Processing Systems* 34 (2021).
[11] Torsten Hothorn, Thomas Kneib, and Peter Bühlmann. 2014. Conditional transformation models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 76, 1 (2014), 3–27.
[12] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*. PMLR, 448–456.
[13] Yan Y Kagan and David D Jackson. 2000. Probabilistic forecasting of earthquakes. *Geophysical Journal International* 143, 2 (2000), 438–453.
[14] Balaji Lakshminarayanan, Daniel M Roy, and Yee Whye Teh. 2016. Mondrian forests for large-scale regression when uncertainty matters. In *Artificial intelligence and statistics*. PMLR, 1478–1487.
[15] HyunYong Lee, Nac-Woo Kim, Jun-Gi Lee, and Byung-Tak Lee. 2019. Uncertainty-aware Deep Learning Forecast using Dropout-based Ensemble Method. In *2019 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 1120–1125.
[16] JK Lindsey. 1998. A study of interval censoring in parametric regression models. *Lifetime data analysis* 4, 4 (1998), 329–354.
[17] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 1412–1421.
[18] Alexander März. 2019. XGBoostLSS–An extension of XGBoost to probabilistic forecasting. *arXiv preprint arXiv:1907.03178* (2019).
[19] Jakub Nowotarski and Rafał Weron. 2018. Recent advances in electricity price forecasting: A review of probabilistic forecasting. *Renewable and Sustainable Energy Reviews* 81 (2018), 1548–1568.
[20] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. 2020. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting* 36, 3 (2020), 1181–1191.
[21] Bruce W Turnbull. 1976. The empirical distribution function with arbitrarily grouped, censored and truncated data. *Journal of the Royal Statistical Society: Series B (Methodological)* 38, 3 (1976), 290–295.
[22] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*. 1–7.
[23] Yao Zhang, Jianxue Wang, and Xifan Wang. 2014. Review on probabilistic forecasting of wind power generation. *Renewable and Sustainable Energy Reviews* 32 (2014), 255–270.
[24] Lin Zhu, Wei Yu, Kairong Zhou, Xing Wang, Wenxing Feng, Pengyu Wang, Ning Chen, and Pei Lee. 2020. Order Fulfillment Cycle Time Estimation for On-Demand Food Delivery. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2571–2580.

A MATHEMATICAL PROOF OF EQU. 7

In this section, we introduce the mathematical proof of Equ. 7. Given an RL order, the labels are denoted as (l, y, u), where y is the assumed precise label, and is not available in practice. l/u is the lower/upper bound, thus $l < y \leq u$, and F is the predicted CDF, then $F(l) \leq F(y) \leq F(u)$. We know

$$\text{CRPS}(F, y) - \text{S-CRPS}(F, l, u) = \int_l^y F(x)^2 \mathrm{d}x + \int_y^u (1 - F(x))^2 \mathrm{d}x. \tag{14}$$

For brevity, we define that

$$\Delta\text{QL}_q = \text{QL}_q(F, y) - \text{S-QL}_q(F, l, u),$$

then

$$2\int_0^1 \text{QL}_q(F, y)\mathrm{d}q - 2\int_0^1 \text{S-QL}_q(F, l, u)\mathrm{d}q = 2\int_0^1 \Delta\text{QL}_q \mathrm{d}q. \tag{15}$$

Using the formula for integration by parts, we can get the following two equations:

$$\int F^{-1}(q)\mathrm{d}q = qF^{-1}(q) - \int F(x)\mathrm{d}x,$$
$$2\int qF^{-1}(q)\mathrm{d}q = q^2F^{-1}(q) - \int F(x)^2\mathrm{d}x,$$

where $x = F^{-1}(q)$. The Equ. 15 can be calculated separately as follows,

$$2\int_0^{F(l)} \Delta\text{QL}_q \mathrm{d}q = (y - l)F(l)^2,$$
$$2\int_{F(l)}^{F(y)} \Delta\text{QL}_q \mathrm{d}q = (l - y)F(l)^2 + \int_l^y F(x)^2\mathrm{d}x,$$
$$2\int_{F(y)}^{F(u)} \Delta\text{QL}_q \mathrm{d}q = (y - u)(1 - F(u))^2 + \int_y^u (1 - F(x))^2\mathrm{d}x,$$
$$2\int_{F(u)}^{1} \Delta\text{QL}_q \mathrm{d}q = (u - y)(1 - F(u))^2.$$

Through addition of the above four equations, we can prove that Equ. 14 is equivalent to Equ. 15, and Equ. 7 is proved further.

B MODEL IMPLEMENTATION DETAILS

B.1 Feature Preprocessing

In Sec. 4.4, we refer to the DCN model [22] to encode each category of features to the vector representation. The origin DCN model is consist of two components, i.e. the cross network and deep network, as illustrated in Fig. 6. The i-th cross layer is calculated as follows,

$$x_i = x_0 x_{i-1}^{\mathrm{T}} w_{i-1} + b_{i-1} + x_{i-1}, \tag{16}$$

where x_0 denotes the origin features, concatenated by the dense features and the embeddings of sparse features. x_i is the output of the i-th cross layer, and w_{i-1} and b_{i-1} are parameters. We use a three-layer cross network, i.e. $v_{cross} = x_3$. The deep network is the common Multilayer Perceptron, calculated as follows,

$$v_{deep} = \text{DNN}^3(x_0), \tag{17}$$

where DNN^k means a neural network with k FC layers. The FC layer is calculated by

$$\text{FC}_f(x) = f(Wx + b), \tag{18}$$

where $f(\cdot)$ denotes the non-linear activation function, Leaky Rectified Linear Unit is used here and W and b are the parameters. Based

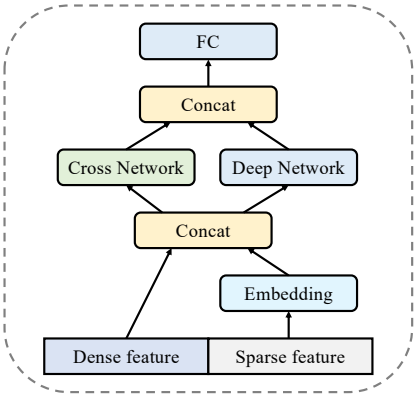


Figure 6: Structure of DCN.

on the outputs of cross network and deep network, we get the final result as

$$v = \text{FC}([v_{cross}; v_{deep}]). \tag{19}$$

B.2 Time Discretization Based Approach

Apart from the quantile discretization, time discretization of S-CRPS is also feasible under certain circumstances, and similar idea has been mentioned in [8]. Specifically, we discretize the FPT similar to the calculation of DCP in Sec 4.1, and predict $\{F(z) = \mathbb{P}(t \leq z) | z \in Z\}$. Implementation of the neural network model is illustrated in Fig. 7. Based on the vector representation v of the order, we use a

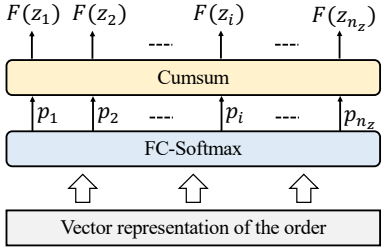


Figure 7: Structure of model based on time discretization.

FC layer activated by Softmax to produce the probability densities as follows:

$$p_i = \mathbb{P}(z_{i-1} < t \leq z_i), 1 \leq i \leq n_z. \tag{20}$$

The cumulative sum of the probability densities, denoted as $F(z_i)$, $1 \leq i \leq n_z$, is the predicted cumulative densities, based on which S-CRPS can be calculated approximately. The loss for an order is calculated as follows,

$$\mathcal{L}(\Theta) = \begin{cases} \sum_{i=1}^{I(y)-1} F(z_i)^2 + \sum_{i=I(y)}^{n_z} (1 - F(z_i))^2, & c = 0 \\ \sum_{i=1}^{I(l)-1} F(z_i)^2 + \sum_{i=I(u)}^{n_z} (1 - F(z_i))^2, & c = 1 \end{cases}, \tag{21}$$

where we omit the constant term Δz and $I(\cdot)$ denotes the index meeting $z_{I(x)-1} < x \leq z_{I(x)}$. The method based on time discretization outputs the cumulative densities of some discrete FPT, while

quantile discretization produces the quantiles for some cumulative densities. Besides, there are two drawbacks in the use of time discretization: (1) A global maximum bound of FPT must be set in advance, and time discretization can not be adopted in scenarios where minimum/maximum bound of the predictive value is unavailable; (2) As shown in Equ. 21, values approximate to some z are used for calculation, rather than the precise $y/l/u$. Therefore, some precise information contained in the labels is lost in time discretization. Relatively, the method based on quantile discretization is more flexible, universal and effective, compared with the time discretization based one.

B.3 Parameter Selection

Tab. 6 shows the hyper-parameters setup of our proposed model, and some parameters can be tuned according to the practical situation.

Table 6: The hyper parameters setup.

Hyper parameters	Value
Dimension of sparse feature embeddings	8
Dimension of feature's vector representation	$k = 128$
# of nodes in FC hidden layers	128
Adam optimizer	$\beta_1 = 0.9, \beta_2 = 0.999$
Learning rate	0.01