Published in IEEE Transactions on Intelligent Transportation Systems (2023) 24 (12), 14556-14569. DOI: 10.1109/TITS.2023.3297948

A Poisson-Based Distribution Learning Framework for Short-Term Prediction of Food Delivery Demand Ranges

Jian Liang, Jintao Ke, Hai Wang, Hongbo Ye, and Jinjun Tang

***Abstract*— The COVID-19 pandemic has caused a dramatic change in the demand composition of restaurants and, at the same time, catalyzed on-demand food delivery (OFD) services—such as DoorDash, Grubhub, and Uber Eats—to a large extent. With massive amounts of data on customers, drivers, and merchants, OFD platforms can achieve higher efficiency with better strategic and operational decisions; these include dynamic pricing, order bundling and dispatching, and driver relocation. Some of these decisions, and especially proactive decisions in real time, rely on accurate and reliable short-term predictions of demand ranges or distributions. In this paper, we develop a Poisson-based distribution prediction (PDP) framework equipped with a double-hurdle mechanism to forecast the range and distribution of potential customer demand. Specifically, a multi-objective function is designed to learn the likelihood of zero demand and approximate true demand and label distribution. An uncertainty-based multi-task learning technique is further employed to dynamically assign weights to different objective functions. The proposed model, evaluated by numerical experiments based on a real-world dataset collected from an OFD platform in Singapore, is shown to outperform several benchmarks by achieving more reliable demand range forecasting.**

***Index Terms*— Short-term demand forecasting, demand distribution, label distribution learning, on-demand food delivery, Poisson distribution.**

I. Introduction

THE outbreak of the COVID-19 pandemic has brought about significant shifts in the demand dynamics of restaurants, concurrently fueling the growth of on-demand food delivery (OFD) services. OFD platforms, such as Uber Eats, Grubhub, and DoorDash, stand out by serving people's demand for online food ordering and delivery. These OFD platforms provide intermediate services that connect customers, drivers, and restaurants and build their revenue by charging commissions to restaurants and/or customers [1]. The global market size of the OFD industry is estimated to surpass $200 billion by 2025 [2], and is a representative transportation-enabled urban service [3]. Because the food delivery industry deals with large volumes of perishable products, accurate and robust demand forecasting is critical for the strategic and operational decisions of OFD platforms, such as dynamic pricing, order bundling and dispatching, driver routing, and relocations.

Although demand forecasting has been extensively studied in diverse contexts, such as ride-hailing [4], [5], [6] and taxi [7], demand forecasting for OFD platforms deserves additional attention because of these services' unique features. First, OFD platforms may be more concerned with the range of possible demand rather than an estimated point value. Since food is perishable, an estimation of demand range with upper and lower bounds can help the platform make more robust operational decisions. For example, by knowing the upper bound of customer demand in the next 10 minutes within a region, the platform can infer the maximum waiting time for customers and design proactive operational strategies—for instance, dispatching more drivers from other regions to this region and adjusting the delivery price/wage to balance supply and demand. Therefore, robust optimization requires information on the upper and lower bounds of customer demand.

Second, unlike some travel demand patterns that are relatively stable over a day, such as consistent demand for public transportation, OFD demand exhibits a strong intermittent and temporal pattern—i.e., the number of delivery orders is very imbalanced in different time periods. As shown in Figure 1 based on a real-world dataset collected from an OFD platform in Singapore, most delivery orders are placed during lunchtime (around 12:00 pm) and dinnertime (around 6:00 pm), and demand at other times is very low. Despite high demand in peak hours, the occurrences of high demand across the entire dataset are particularly low, showing a clear long-tail distribution. This renders OFD demand prediction more challenging.

Third, customers' willingness to order food online with delivery is heavily affected by weather and other real-time conditions. For example, when it is cold or raining, customers

Manuscript received 23 November 2022; revised 26 June 2023; accepted 12 July 2023. Date of publication 3 August 2023; date of current version 29 November 2023. This work was supported in part by the National Natural Science Foundation of China under Project 72201223, in part by the Hong Kong Research Grants Council under Project HKU27203323, and in part by The Hong Kong Polytechnic University under Grant P0036644. The work of Hai Wang was supported by the Lee Kong Chian (LKC) Fellowship awarded through Singapore Management University. The Associate Editor for this article was H. Han. *(Corresponding author: Jintao Ke.)*

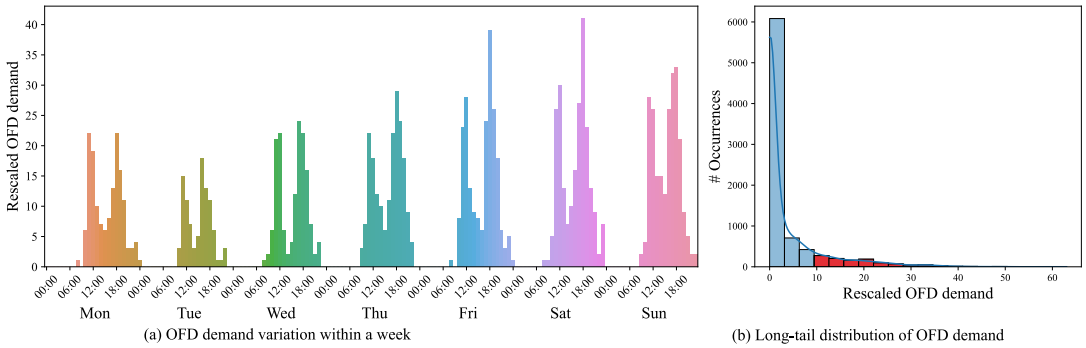
Jian Liang and Jintao Ke are with the Department of Civil Engineering, The University of Hong Kong, Hong Kong (e-mail: liang97@connect.hku.hk; kejintao@hku.hk).

Hai Wang is with the School of Computing and Information Systems, Singapore Management University, Singapore 188065 (e-mail: haiwang@smu.edu.sg).

Hongbo Ye is with the Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University, Hong Kong (e-mail: hongbo.ye@polyu.edu.hk).

Jinjun Tang is with the School of Traffic and Transportation Engineering, Central South University, Changsha 410075, China (e-mail: jinjuntang@csu.edu.cn).

Digital Object Identifier 10.1109/TITS.2023.3297948



(a) OFD demand variation within a week (b) Long-tail distribution of OFD demand

Fig. 1. OFD demand variation on an OFD platform in Singapore within a week and the occurrence distribution of OFD demand in the whole dataset (Y-axis of (a) and X-axis of (b) are rescaled as required by the industry collaborator).

may prefer ordering food for delivery rather than dining in a restaurant. When developing a demand forecasting model for OFD services, therefore, data on weather and other real-time information should be an indispensable input.

Most of the existing demand forecasting methods focus on a scalar or point prediction—namely, the prediction output is a single value, which has no prediction for the range or distribution of demand. A good solution to this problem could be label distribution learning (LDL) [8], [9]. LDL converts the ground truth to a discrete distribution and trains a model by fitting the entire distribution. LDL has two major advantages. On the one hand, the output of LDL is a distribution, which contains information on the reliability of the prediction as well as the upper and lower bounds of the predicted demand. Since demand uncertainty is a critical concern for many design and operational decisions in the general context of on-demand transportation and delivery, such as fleet sizing [10], [11], spatial and temporal pricing [12], order dispatching [13], [14], routing and scheduling [15], reward mechanism [16], demand-supply coordination [17], [18], ride-pooling [19], [20], supply management [21], [22], [23], and driver relocation [24], [25], the prediction of demand ranges is of great importance to OFD platforms for more robust and reliable decisions. In addition, unlike some traditional demand distribution estimation methods that assume that demand within the same time period or location is distributed with the same parameters, LDL easily derives a unique distribution of OFD demand for each time period and each location in real time.

On the other hand, the OFD demand samples exhibit a long-tailed distribution, where demands in peak hours have only a few samples but the others are associated with massive sample points. In this case, most scalar prediction models are easily biased towards low demand with massive training data, resulting in poor performance on high demand during peak hours with limited data. LDL puts a distribution over an ambiguous label, i.e., label smoothing or data augmentation on the label side, which could beneficially prevent overfitting and promote generalization [26]. This helps improve regression performance and may be very helpful in mitigating the long-tail effect. In literature, LDL techniques are mostly used for facial age estimation and other computer vision tasks and have seldom been used for customer demand prediction; this includes OFD demand prediction. A method based on and adapted from LDL techniques is certainly worth investigating and would potentially improve short-term demand range forecasting for OFD services.

However, LDL generally assumes that the labels of instances obey Gaussian distributions with the same standard deviation, which is not realistic because the heterogeneity among instances is not negligible. In addition, the standard deviation of the Gaussian distributions must be predefined as a hyperparameter to train the models, which requires additional model tuning and the resolution of potential overfitting issues. To tackle such issues related to LDL methods, we define the label distributions of instances based on a Poisson distribution, the only parameter of which is calibrated by ground truth; thus, no hyperparameter of the Poisson distribution is needed for model training. In addition, the Poisson distribution is typically used to characterize the probability of a given number of events occurring within a fixed period of time and region, and it can naturally characterize the arrivals of customer orders for food delivery services and thus be suitable for OFD demand forecasting.

In this paper, we develop a novel Poisson-based distribution prediction (PDP) framework integrated with a double-hurdle mechanism and uncertainty-based multi-task learning techniques to address the challenges described above. Given the characteristics of customers' arrivals for placing food delivery orders, we assume that the OFD demand in each region and each time period follows a Poisson distribution whose parameter is calibrated by the ground truth, and thus covert label scalar into label distribution. We then develop a neural network with multiple outputs to learn the likelihood of demand being zero and approximate the label distribution. A multi-objective function is designed to achieve three goals: (1) minimizing the Binary Cross Entropy (BCE) between the binarized demand and the predicted probability of the demand being non-zero, (2) minimizing the Kullbac-kLeibler (KL) divergence between the distribution drawn from the ground truth and the distribution predicted by the neural network, and (3) minimizing the difference between the ground truth value and the peak value (i.e., the value with highest probability density) of the predicted distribution. An uncertainty-based multi-task learning technique is applied to organize different

learning objectives. By doing this, the algorithm can generate a demand distribution that is as close as possible to the real demand in terms of both the peak value and the distribution itself. We also incorporate real-time information in the prediction, which includes temperature and rainfall. The main contributions of this paper are listed as follows:

- We introduce a label distribution learning technique in OFD demand forecasting. To the best of our knowledge, this is the first attempt to develop a distribution prediction method for OFD demand range forecasting.
- We develop a novel Poisson-based distribution prediction framework equipped with a double-hurdle mechanism to solve the data imbalance problem. Additionally, the PDP framework incorporates an uncertainty-based multi-task learning technique to coordinate different learning objectives adaptively.
- We conduct extensive experiments based on a real dataset from an on-demand food delivery platform in Singapore. Experiments show that PDP-based methods outperform several benchmarks by achieving more reliable demand range forecasting. Moreover, further ablation experiments demonstrate the efficient design of the proposed PDP framework.

The remainder of the paper is organized as follows. In Section II we review the relevant literature. In Section III we formally define the OFD demand distribution prediction problem and propose the PDP framework. In Section IV we conduct a set of numerical experiments and discuss the ablation experiments. Section V concludes with recommendations for future studies.

II. Literature Review

Our work is closely related to the literature on short-term demand forecasting in urban transportation systems, which includes a series of statistical and machine learning algorithms to capture the spatial and temporal correlations of transportation demand. It is also related to the literature on label distribution learning, which is mainly used for facial age estimation, gesture detection, and pre-release movie prediction in the field of computer vision.

A. Demand Forecasting for Urban Transportation Systems

Passenger demand forecasting plays an important role in alleviating the imbalance between demand and supply in urban transportation systems and has received much attention in recent decades. Earlier studies focused on various time-series forecasting models, including auto-regressive integrated moving average (ARIMA) [27]; Kalman filtering models [28]; fuzzy neural networks [29]; and recurrent neural networks (RNN) [30], [31]. Li et al. [32] build a multitask learning framework based on a long short-term memory network (LSTM) for multimodal demand co-prediction. These methods do not consider spatial correlations.

Due to their ability to capture both temporal and spatial dependencies simultaneously, grid-based deep learning neural networks (DNN) have quickly become popular in recent years [4], [33]. Demands scattered in different grid cells can be treated as images, after which convolutional operations are used to characterize spatial correlations [34], [35], [36]; and RNN [7], [37] or temporal convolutional networks (TCN) [38], [39] are used to capture temporal dependencies. On the basis of grid division using regular hexagons, Ke et al. [40] develop a hexagon-based CNN to improve demand prediction performance for on-demand ride-hailing services. Despite its superior performance, Grid-based DNN only captures local spatial correlations between adjacent areas and fails to examine non-Euclidean pair-wise correlations between distant areas [6].

Motivated by the great success of graph neural networks on non-Euclidean data, graph-based methods, such as graph convolutional networks [41] and graph attention networks [42], have recently been introduced to demand forecasting. To address the challenge of heterogeneous spatial dependencies between areas, a variety of spatiotemporal graph neural networks [43] have been developed. Tang et al. [44] integrate a spatiotemporal graph convolutional network with a community detection algorithm to predict regional-level passenger demand. Ke et al. [45] propose a multi-graph-based approach to predict demand for different service modes. Another representative graph-based method is the dynamic graph [46], which can automatically deduce hidden interdependencies between nodes from data. For instance, Zhang et al. [47] design a dynamic node-edge attention network to capture the temporal evolution of node topologies. Bai et al. [48] propose an adaptive graph convolutional recurrent network for traffic forecasting, which includes a node adaptive parameter learning module for learning node-specific patterns and a data adaptive graph generation module for automatically inferring node embedding among different series of traffic.

Although the algorithms developed in previous studies have shown promising results in terms of demand forecasting, little effort has been devoted to demand distribution prediction. Only a few studies [49], [50] investigate the problem of on-demand food delivery service demand prediction, and none of them pay attention to demand distribution forecasting.

B. Label Distribution Learning

Label distribution learning enhances regression performance through data augmentation on the label side [26], which has been demonstrated to be efficient in a variety of tasks. These include facial age estimation [51], head pose estimation [52], and pre-release prediction of movies [53]. LDL defines the label distribution of an instance as a vector, where each probability in the vector represents the likelihood that the instance is equal to each label [9]. Geng et al. [54] propose an LDL learning algorithm for facial age estimation based on a maximum entropy model. To resolve the inconsistency between the training stage and the evaluation stage, Gao et al. [55] combine LDL and expectation regression into an end-to-end learning framework.

These LDL methods are majorly used for computer vision tasks, such as facial age estimation and head pose estimation. They generally convert the label into probability distribution by assuming that facial age or head pose in each image

follows a Gaussian distribution centered on the label with the same standard deviation. Although the assumption of Gaussian distribution is valid for face age estimation and head pose estimation, it may not be suitable for OFD demand estimation. In contrast, the arrivals of customer orders for food delivery services are better characterized by a Poisson distribution. Additionally, considering the influence of individual differences such as gender, ethnicity, and photography habit, it is unreasonable to assume that the standard deviation of the age/head pose label distribution of all samples is the same. Therefore, necessary adjustments should be made to adapt LDL techniques to the task of short-term OFD demand distribution estimation.

III. Methodology

In this section, we first describe the notation and define the problem of demand distribution learning. Then, we propose and discuss the Poisson-based distribution prediction (PDP) framework in depth. Finally, we compare the scalar prediction model, the Gaussian-based LDL method, and the proposed PDP framework.

A. Notation and Problem Definition

The goal of this paper is to predict the short-term demand distribution within a specific region based on historical demand observations and real-time information such as rainfall and temperature. We use non-bold lowercase letters to denote scalars, bold lowercase letters to denote vectors, and bold capital letters to denote matrices.

Definition 1: Feature matrix $\boldsymbol{X} = \{\boldsymbol{d}, \boldsymbol{r}, \boldsymbol{\tau}, \boldsymbol{s}\}$. The feature matrix represents the model's inputs, which contain three parts: historical demand sequence ($\boldsymbol{d}$), historical weather sequences ($\boldsymbol{r}$ and $\boldsymbol{\tau}$), and static features ($\boldsymbol{s}$). We denote the demand in time period t as d_t, which refers to the number of OFD orders placed in time period t. Considering the strong periodicity (both daily and weekly) of OFD demand, similar to [33], we truncate three time-series segments of length t_r, t_d, and t_w along the time axis as inputs of recent, daily-period, and weekly-period components, respectively. We concatenate the three segments into a vector $\boldsymbol{d} = \{d_{t-t_w*week}, \cdots, d_{t-1*week}, d_{t-t_d*day}, \cdots, d_{t-1*day}, d_{t-t_r}, \cdots, d_{t-1}\}$. The weather information considered in this paper includes rainfall and temperature, both of which are also time series data. Following the same procedure as for the OFD demand sequence, we define and obtain rainfall vector $\boldsymbol{r}$ and temperature vector $\boldsymbol{\tau}$. Finally, we incorporate some static features $\boldsymbol{s}$ to assist prediction, including day of week, hour of day, and location ID.

Definition 2: Proxy label distribution $\boldsymbol{p}$. For an ordinary demand forecasting problem, the ground-truth demand y (i.e., label) is a scalar and the aim is to minimize the difference (i.e., mean square error) between the predicted demand $\hat{y}$ and label y. For the problem of demand distribution forecasting, instead of taking a scalar as the label, we discretize the range of possible demand values into segments and generate a discrete probability distribution for the label value within a range. Since OFD demand is an integer value, we then define the ordered label set as:

$$\begin{aligned} \boldsymbol{l} &= \{1, 2, \cdots, |\boldsymbol{l}|\}, \\ |\boldsymbol{l}| &= (1+e) * y_{\max}, \end{aligned} \tag{1}$$

where $|\boldsymbol{l}|$ denotes the length of ordered label set, y_{max} denotes the largest label value in the dataset, and e is the hyperparameter that controls the range of possible demand value. Then, a probability mass function (pmf) is chosen to describe the proxy label distribution $\boldsymbol{p} \in \mathbb{R}^{|\boldsymbol{l}|}$ over the $|\boldsymbol{l}|$ segments.

Table I summarizes the definitions of notation.

TABLE I
Notation

Notation	Definition
$\boldsymbol{d}$	Historical demand sequence
$\boldsymbol{r}$	Historical rainfall sequence
$\boldsymbol{\tau}$	Historical temperature sequence
$\boldsymbol{s}$	Static features
$\boldsymbol{l}$	Ordered label set
$\boldsymbol{X}$	Model inputs
$\boldsymbol{p}$	Proxy label distribution
$\hat{\boldsymbol{p}}$	Predicted label distribution
y	Ground-truth demand
$\hat{y}$	Predicted demand
y_b	$y_b = 0$ if $y = 0$, and $y_b = 1$ if $y > 0$
$\hat{y}_b$	Probability that demand is predicted to be non-zero
$\boldsymbol{\mathcal{W}}$	All parameters in the model, including $\boldsymbol{W}$, $\boldsymbol{b}$, $\boldsymbol{w_s}$, b_s, and $\boldsymbol{\theta}$

Ideally, the proxy label distribution $\boldsymbol{p}$ generated from the ground truth y should be unimodal and the probability of the integer closest to y in distribution $\boldsymbol{p}$ should be the largest. Conceptually, this not only ensures that the ground truth is the peak of the label distribution but also causes the probability of demand taking other values to decrease with the distance from y. Although there are other probability distributions that meet the requirements, we choose the Poisson distribution for three reasons:

- The Poisson distribution normally describes the number of events that occur in a given time period, such as the number of telephone calls per minute. OFD demand refers to the number of delivery orders within a fixed-length time interval and within a given region, which is well described by a Poisson distribution.
- The Poisson distribution has only one parameter λ (the expected number of event occurrences in the period), and a Poisson distribution has the largest probability density when the random variable takes a value close to λ. Thus, a natural way to derive a label distribution is to set λ as the ground truth. Other than λ (which is also the expectation), we do not need any other parameters, such as the standard deviation in Gaussian distribution.
- When the total number of event occurrences follows a Poisson distribution, the inter-arrival time between successive events follows exponential distribution, which has a memoryless property and improves tractability when used in any follow-up analytical model [56].

Based on the assumption of Poisson distribution, we can calculate the probability for each label in $\boldsymbol{l}$ using the probability density function of Poisson distribution. To ensure that

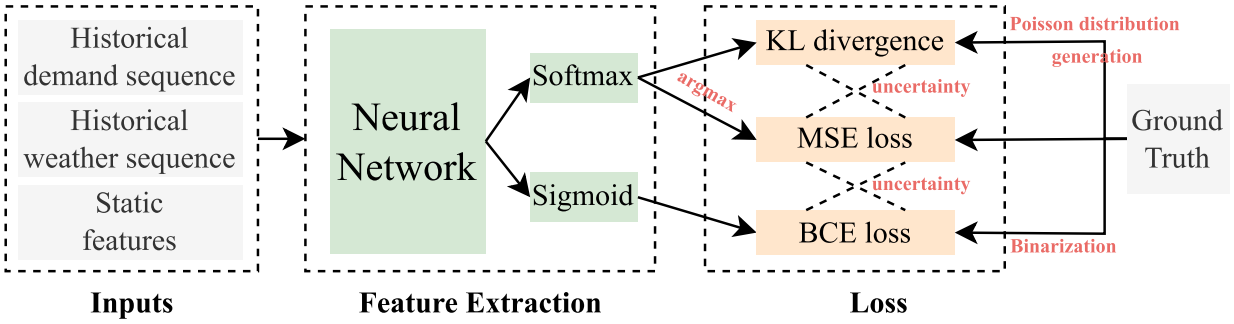


Fig. 2. Poisson-based distribution prediction framework.

the sum of the probabilities of all labels in $\boldsymbol{l}$ is 1, we normalize each probability to get the proxy label distribution $\boldsymbol{p}$:

$$\boldsymbol{p} = \left\{p_1, p_2, \cdots, p_k, \cdots, p_{|\boldsymbol{l}|}\right\}, \tag{2}$$

where p_k is the probability of label k in $\boldsymbol{l}$ and calculated as:

$$p_k = \frac{\frac{y^k}{k!}e^{-y}}{\sum_{i=1}^{|\boldsymbol{l}|}\frac{y^i}{i!}e^{-y}} = \frac{y^k}{k!\sum_i^{|\boldsymbol{l}|}\frac{y^i}{i!}}. \tag{3}$$

Problem Definition: we formulate the problem as learning a mapping function $\mathcal{F}$ to predict the proxy label distribution $\boldsymbol{p}$ based on the feature matrix $\boldsymbol{X}$:

$$\boldsymbol{p} = \mathcal{F}(\boldsymbol{X}). \tag{4}$$

B. Poisson-Based Distribution Prediction Framework

In this section, we propose the PDP framework. As shown in Figure 2, the framework contains two modules: a feature extraction module and a loss module. The feature extraction module aims to learn valuable information from historical demand sequences, historical weather sequences, and static features. It generates two outputs: the probability of the predicted demand being zero and the predicted demand distribution. Introducing the probability of the predicted demand being zero is inspired by the traditional double-hurdle model [57], which is often used to deal with data with a large number of zero values and is especially applicable to our case. This statistical model involves two steps: a binary model estimates the probability of the outcome being zero or non-zero, followed by a separate model to estimate the non-zero observations. Similarly, in our approach, if the probability of the predicted demand being zero exceeds 0.5, we predict the demand as zero. Otherwise, we utilize the demand value with the highest probability density derived from the predicted demand distribution, as our final prediction. This approach is referred to as the double-hurdle mechanism in our study. Subsequently, the loss module calculates the difference between the ground truth and the outputs generated by the feature extraction module.

1) Feature Extraction Module: Given a sample $(\boldsymbol{X}, y, \boldsymbol{p})$, the feature extraction module takes $\boldsymbol{X}$ as input and uses a neural network to extract features. Since the neural network is essentially a function, it can be denoted as $f(\cdot, \theta)$ where θ represents the parameters of the neural network. The output of the neural network is represented as $f(\boldsymbol{X}; \theta)$. Based on the features extracted by $f(\cdot, \theta)$, we first use a fully connected layer with a sigmoid activation to predict the likelihood that the OFD demand is equal to non-zero, that is,

$$\hat{y}_b = \frac{1}{1 + e^{-\boldsymbol{w}_s{}^T f^{\boldsymbol{\theta}}(\boldsymbol{X}) + b_s}}. \tag{5}$$

To keep the dimensions of the predicted label distribution consistent with the dimensions of proxy label distribution, $f(\boldsymbol{X}; \theta)$ is fed to a fully connected layer with $|\boldsymbol{l}|$ neurons. Setting a separate fully-connected layer instead of including it in the neural network $f(\cdot, \theta)$ can remove the restriction on the structure of $f(\cdot, \theta)$, making our model more generalized. Then, a softmax function is applied to convert the outputs from the fully connected layer, denoted as $\boldsymbol{z}$, into a probability distribution $\hat{\boldsymbol{p}}$; that is,

$$\begin{aligned} \boldsymbol{z} &= \left[z_1, z_2, \cdots, z_k, \cdots, z_{|\boldsymbol{l}|}\right]^T = \boldsymbol{W}^T f^{\boldsymbol{\theta}}(\boldsymbol{X}) + \boldsymbol{b}, \\ \hat{p}^k &= \frac{e^{z_k}}{\sum_{i=1}^{|\boldsymbol{l}|} e^{z_i}}, \\ \hat{\boldsymbol{p}} &= \left\{\hat{p}^1, \hat{p}^2, \cdots, \hat{p}^k, \cdots \hat{p}^{|\boldsymbol{l}|}\right\}, \end{aligned} \tag{6}$$

where both $\boldsymbol{W} = \left[w_1, w_2, \cdots, w_k, \cdots w_{|\boldsymbol{l}|}\right]^T \in \mathbb{R}^{?\times|\boldsymbol{l}|}$ and $\boldsymbol{b} \in \mathbb{R}^{|\boldsymbol{l}|}$ are learnable parameters in the fully connected layer. Since the proxy label distribution is generated from a Poisson distribution parameterized by the ground truth, which is only a proxy for the real distribution (which is unknown to us), there could be an inconsistency between the training objective and evaluation metric. To reduce this inconsistency, we jointly learn the demand distribution and peak demand scalar in an end-to-end manner. Since the output from the softmax function is the predicted distribution $\hat{\boldsymbol{p}}$, we can infer the estimated peak demand $\hat{y}$ from the distribution $\hat{\boldsymbol{p}}$. Given that the proxy label distribution $\boldsymbol{p}$ reaches its peak at the value of ground truth y, applying the argmax function is a natural choice to infer the estimated peak demand scalar $\hat{y}$ from the predicted label distribution $\hat{\boldsymbol{p}}$, i.e., the predicted demand is the demand that corresponds to the largest probability in the prediction distribution:

$$\hat{y} = \text{argmax}_{i=1,2,\cdots,|\boldsymbol{l}|}\, \hat{p}^i. \tag{7}$$

However, the argmax function is non-differentiable; thus if argmax is directly used to obtain predicted demand, the gradients cannot be backpropagated from the loss function to the trainable parameters. To address this problem, we then use

the soft-argmax operation [58], which is a smoothed version of argmax that preserves end-to-end differentiability as follows:

$$\begin{aligned}\hat{y} &= \text{argmax}_{i=1,2,\cdots,|\boldsymbol{l}|}\, \hat{p}^i \approx \sum_{j=1}^{|\boldsymbol{l}|} j \cdot \text{softmax}\left(\beta \hat{p}^j\right) \\ &= \sum_{j=1}^{|\boldsymbol{l}|} \frac{j e^{\beta \hat{p}^j}}{\sum_{i=1}^{|\boldsymbol{l}|} e^{\beta \hat{p}^i}},\end{aligned} \tag{8}$$

where $\beta \geq 1$ is a hyperparameter.

2) Loss Module: The loss function used to evaluate the prediction performance consists of three parts: distribution loss, regression loss, and classification loss. As depicted in Figure 2, the KL divergence is used to measure the difference between proxy label distribution and predicted label distribution. We employ two commonly used metrics, i.e., mean squared error (MSE) and Binary Cross Entropy (BCE), to measure the regression loss and classification loss, respectively. Formally,

$$\begin{aligned}\mathcal{L}_{kl} &= \sum_{k=1}^{|\boldsymbol{l}|} p^k \cdot \ln \frac{p^k}{\hat{p}^k}, \\ \mathcal{L}_{mse} &= (y - \hat{y})^2, \\ \mathcal{L}_{bce} &= -(y_b \cdot \log \hat{y}_b + (1 - y_b) \cdot \log (1 - \hat{y}_b)).\end{aligned} \tag{9}$$

Integrating multiple loss functions within a composite loss function can pose several challenges. Each loss function may have a different scale or magnitude. Determining the appropriate weights for each loss becomes a challenge, often requiring extensive experimentation to achieve the desired balance. To tackle this issue, we apply an advanced multi-task learning technique that learns optimal loss weights based on the homoscedastic uncertainty of different learning objectives [59]. This multi-task learning technique uses a Gaussian distribution to model the homoscedastic uncertainty associated with each task. The homoscedastic uncertainty, which remains constant across all input data but varies between different tasks, is then used to determine the weight assigned to each task's loss. In our study, three tasks are involved, corresponding to three loss functions, namely, $\mathcal{L}_{kl}$, $\mathcal{L}_{mse}$, and $\mathcal{L}_{bce}$. The final loss function can be represented as the negative log-likelihood of joint probabilities for different tasks, namely,

$$\begin{aligned}&\mathcal{L}\left(\boldsymbol{\mathcal{W}}, \sigma_1, \sigma_2, \sigma_3\right) \\ &= -\log p\left(\hat{\boldsymbol{p}}, \hat{y}, \hat{y}_b | f^{\boldsymbol{\mathcal{W}}}(\boldsymbol{X})\right) \\ &= -\log\left[\mathcal{N}\left(\hat{\boldsymbol{p}}; f^{\boldsymbol{\mathcal{W}}}, \sigma_1^2\right) \mathcal{N}\left(\hat{y}; f^{\boldsymbol{\mathcal{W}}}, \sigma_2^2\right) \mathcal{N}\left(\hat{y}_b; f^{\boldsymbol{\mathcal{W}}}, \sigma_3^2\right)\right] \\ &\propto \frac{1}{2\sigma_1^2}\mathcal{L}_{kl} + \frac{1}{2\sigma_2^2}\mathcal{L}_{mse} + \frac{1}{2\sigma_3^2}\mathcal{L}_{bce} + \log\left(\sigma_1 \sigma_2 \sigma_3\right),\end{aligned} \tag{10}$$

where $\boldsymbol{\mathcal{W}}$ denotes all parameters in the model, $f^{\boldsymbol{\mathcal{W}}}(\cdot)$ denotes the entire feature extraction module, and σ_i represents the model's observation noise parameter, measuring the uncertainty in the i-th task. The coefficient term $1/2\sigma_i^2$ can be interpreted as the relative weight assigned to the i-th task, indicating its importance in the overall loss function. Besides, the term $\log \sigma_1 \sigma_2 \sigma_3$ serves as a regularizer, preventing the value of σ_i from excessively increasing. By leveraging this uncertainty, the technique dynamically adjusts the loss weights, giving more importance to tasks with lower uncertainty and effectively balancing the learning process across multiple tasks.

Algorithm 1 calculation of Upper and Lower Bounds

Input: $\hat{\boldsymbol{p}}$, $\hat{y}$, P_{thre}
Output: predicted upper and lower bounds a and b
Initialize $P_{cum} = \hat{p}^{\hat{y}}, b = \hat{y} - 1, a = \hat{y} + 1$;
while $P_{cum} < P_{thre}$ **do**
  $P_{cum} = P_{cum} + \hat{p}^b + \hat{p}^a$;
  $b = \max\{b - 1, 0\}$;
  $a = \min\{a + 1, |\boldsymbol{l}|\}$;
end

3) Range Estimation: Since many strategic and operational decisions by OFD platforms, such as robust dynamic pricing, require knowing the upper and lower bounds (denoted as a and b) of the predicted demand, we define another metric that quantifies the model's performance in predicting such upper and lower bounds. Specifically, we need to find a way to infer reasonable upper and lower bounds using the predicted probability distribution $\hat{\boldsymbol{p}} = \left\{\hat{p}^1, \hat{p}^2, \cdots, \hat{p}^k, \cdots \hat{p}^{|\boldsymbol{l}|}\right\}$ (note that $\hat{p}^k$ represents the probability that the model believes the predicted demand is k), and then propose a metric to evaluate the accuracy of the predicted upper and lower bounds. The method of choosing the upper and lower bounds is explained in detail in Algorithm 1; simply put, starting from the most probable predicted value (i.e., the $\hat{y}$ value), we move one unit to the left and right simultaneously and accumulate the corresponding probabilities until the accumulated probability P_{cum} exceeds a preset threshold P_{thre}. The cumulative probability P_{cum} represents the probability that the model believes that the predicted demand is within the range $[b, a]$, and the preset threshold P_{thre} represents the operator's requirement for confidence.

4) Relationship Between Different Methods: In this subsection, we discuss the relationship between PDP, LDL, and the scalar prediction. Figure 3 compares the predictions and labels of the three methods. These three methods are learned by minimizing the difference between predictions and labels. Assume that the OFD demand follows a latent distribution, whose parameters vary over time and space. A scalar prediction model assumes that the expectation of the latent distribution is the observed demand (i.e., the label) and is learned by minimizing the difference (e.g., mean square error) between the label and the predicted demand. LDL is a neural network-based learning paradigm that assumes that the latent distribution is a Gaussian distribution with a mean as the observed demand and variance as a fixed hyperparameter. In detail, LDL first converts the observed demand into a discrete probability distribution (label distribution) based on the Gaussian distribution assumption, and the demand is then learned by minimizing the KL divergence between the label distribution and the neural network output. The proposed PDP is an improved version of LDL that takes into account the

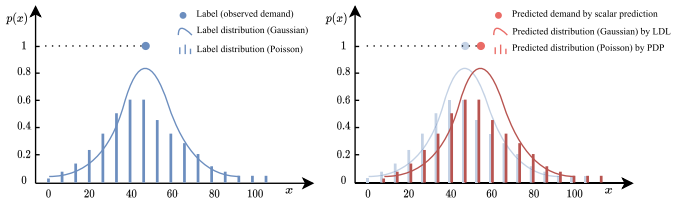


Fig. 3. Comparison of the predictions and labels.

characteristics of OFD demand. It assumes that the latent distribution is Poisson distribution parameterized by the observed demand. PDP also considers the argmax over the predicted distribution and the predicted demand is learned by balancing the tradeoff between the KL divergence and mean square error. Hence, the difference between the three methods is based on different inherent assumptions and different objective functions.

IV. Numerical Experiments

In this section, we first present the datasets and experiment details. Then we compare the proposed PDP framework with several benchmark methods. Lastly, we evaluate the effectiveness of our PDP framework using a set of ablation experiments.

A. Datasets

The datasets used in this paper are obtained from a crowd-sourcing delivery platform in Asia.[1] It includes real order information over a period of 10 months from November 8, 2020 to October 21, 2021. Each order record includes the requesting time and pickup location ID. The location ID has five possible values that represent the five Community Development Councils (CDC) of Singapore: the North East CDC, North West CDC, South East CDC, South West CDC, and Central Singapore CDC. To identify the impact of weather conditions on OFD demand forecasting, we further crawl temperature and rainfall data from the Singapore Government Technology Agency[2] (collected at hourly intervals). To ensure consistency, we average the temperature and rainfall values from the weather stations associated with each CDC. These averaged values are then considered as the temperature and rainfall for the orders within the respective CDC during the corresponding time interval. The temperature range observed in the data is between $22.1°C$ and $35.9°C$, while the rainfall range spans from $0mm$ to $62.5mm$. By aggregating the number of orders from the same location ID within each hour, we get OFD demand. Each OFD demand is described by six columns, including demand value, average temperature, average rainfall, location ID, day of week, and hour of day. Finally, using the time-series generation method described in Section III-A, we generate demand, rainfall, and temperature series. We select data from September 24, 2021 to October 21, 2021 as the test set and data from September 10, 2021 to September 23, 2021 as the validation set. The remaining data are used as the training set.

[1]Some data is rescaled as required by the industry collaborator.
[2]https://data.gov.sg/

B. Evaluation Metrics

Three common regression evaluation metrics—mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE)—are used to evaluate the predictive accuracy for the scalar demand. The formulas of MAE, RMSE, and MAPE are given as follows:

$$\begin{aligned}
\mathrm{MAE} &= \frac{1}{N}\sum_{i=1}^{N}\left|y_i-\hat{y}_i\right|, \\
\mathrm{RMSE} &= \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_i-\hat{y}_i\right)^2}, \\
\mathrm{MAPE} &= \frac{1}{N}\sum_{i=1}^{N}\frac{\left|y_i-\hat{y}_i\right|}{y_i},
\end{aligned} \tag{11}$$

where N is size of the test set.

Additionally, three commonly used metrics for interval prediction are used to evaluate the performance of the models in predicting demand ranges, including Prediction Interval Coverage Probability (PICP), Prediction Interval Normalized Average Width (PINAW), and Interval Score (IS) [60], [61]. PICP measures the accuracy in range, i.e., how many samples in the test set fall within their corresponding predicted demand ranges. PINAW, on the other hand, evaluates the range width and is normalized by dividing the possible range (i.e., $|\boldsymbol{l}|$) of the real demand.

$$\begin{aligned}
\mathrm{PICP} &= \frac{1}{N}\sum_{i=1}^{N}\delta\left(y_i,[b_i,a_i]\right), \\
\delta\left(y_i,[b_i,a_i]\right) &= \begin{cases} 1, & y_i\in[b_i,a_i], \\ 0, & \text{otherwise} \end{cases} \\
\mathrm{PINAW} &= \frac{1}{N}\sum_{i=1}^{N}\frac{a_i-b_i}{|\boldsymbol{l}|}.
\end{aligned} \tag{12}$$

A higher PICP indicates that more observed labels in the test set are located in the predicted range. PINAW measures the width of the predicted range of demand. We expect a higher PICP and a lower PINAW, so that we can more accurately predict stochastic demand with a more narrow predicted range. IS takes into account the balance between a more accurate prediction and a more narrow range, thereby providing a more comprehensive assessment of range prediction [62], [63]. The formula of IS is given by:

$$\mathrm{IS} = \mathrm{PINAW} + \frac{2\eta}{N\cdot P_{thre}}\sum_{i=1}^{N}(y_i-a_i)+(b_i-y_i), \tag{13}$$

where η is a hyperparameter that determines how much penalty is imposed on the estimated range that does not contain the ground truth. The indicator function $\delta(a)$ returns 1 if $a \geq 0$, otherwise return 0. Also, the term $\frac{2}{P_{thre}}$ is used to give more penalty when confidence is lower. We can find from Equation 13 that IS is equal to PINAW when the observed labels are within the estimated range. But when the estimated range cannot cover the observed labels, IS combines PINAW

with a penalty term, the size of which depends on how much the label exceeds the estimated range. By doing so, IS is able to balance the trade-off between accuracy and range width.

C. Benchmarks and Implementation Details

Our PDP framework can be integrated with any neural network used for feature extraction (see Figure 2). In this paper, we choose three classical neural network architectures—MLP, TCN, and LSTM—for comparison. To demonstrate the effectiveness of the proposed PDP framework, we compare the three PDP-based models with the following eight benchmarks:

- **HA**: Historical average (HA) is a basic time-series forecasting method, which uses the average of historical observations within a moving period as prediction results. We take the average of the components in the demand series $\boldsymbol{d}$ as the output of HA.
- **ARIMA**: Autoregressive integrated moving average is one of the most widely used statistical methods in time-series forecasting. It combines autoregressive components and the moving average method. We use the statsmodel package[3] (version 0.12.2) in Python to implement ARIMA, in which the order of autoregressive is set to 5, the degree of difference to 1, and the moving average to 0.
- **RF**: Random forest (RF) is a classic machine learning method that uses the bootstrap sampling method to assemble different decision trees. RF is implemented using the scikit-learn package[4] (version 0.24.2) in Python with default settings.
- **XGBoost**: eXtreme gradient boosting (XGBoost) is a powerful ensemble method and has been successfully used in a wide range of applications. XGBoost is implemented using the xgboost package[5] (version 1.4.2) in Python with default settings.
- **MLP**: Multi-layer perceptron (MLP) is a basic feedforward neural network that consists of an input layer, one or more hidden layers, and an output layer. We use one hidden layer and the number of hidden units is set to 64.
- **TCN**: Temporal Convolutional Network (TCN) captures long-term temporal dependencies through dilated convolution operations and achieves satisfactory results on some time series prediction tasks. A two-layer TCN with a kernel size of 2 is adopted.
- **LSTM**: Long short-term memory (LSTM) is a special recurrent neural network with complex gating mechanisms to capture long-term dependencies. There is only one layer in LSTM with 64 hidden units.
- **QReg**: Quantile Regression (QReg) studies the relationship between the independent variable and the conditional quantile of the dependent variable, which can further infer the conditional probability distribution of the dependent variable. QR is implemented using the statsmodels package in Python with default settings.

[3]https://www.statsmodels.org/stable/
[4]https://scikit-learn.org/stable/
[5]https://xgboost.readthedocs.io/en/latest/

TABLE II
OVERALL COMPARISON OF PDP-BASED METHODS AND BENCHMARK METHODS

Method	MAE	RMSE	MAPE (%)
HA	2.566	5.663	82.531
ARIMA	1.209	2.809	43.735
RF	0.837	1.936	28.598
XGBoost	0.812	1.857	27.811
MLP	0.885	2.019	30.899
TCN	0.821	1.772	26.816
LSTM	0.796	1.973	21.574
PDP_MLP	0.717	1.728	18.973
PDP_TCN	**0.705**	**1.714**	**18.694**
PDP_LSTM	0.731	1.815	19.607

We integrate the three neural networks (MLP, TCN, and LSTM) with the PDP framework to obtain three PDP-based prediction models: PDP_MLP, PDP_TCN, and PDP_LSTM, respectively. All three PDP models are implemented using PyTorch[6] (version 1.9.1) on a server with two NVIDA RTX 3090. The learning rate are set to 0.01 for all three PDP models. These neural networks are trained using the Adam optimizer with an early stopping mechanism to determine the number of epochs when abort training. The early stopping patience is set to 10 and the batch size is set to 64. MLP, TCN, and LSTM are trained under the same environment and setting. In this paper, e is set to 0.1, β is set to 100, η is set to 10, and t_r, t_d, and t_w are set to 12, 7, and 3, respectively.

D. Model Comparison

We first compare the proposed PDP-based neural networks with seven scalar prediction methods, which are shown in Table II. Despite good interpretability, classical time-series forecasting methods (i.e., HA and ARIMA) perform poorly with a low predictive accuracy. In contrast, machine learning methods (e.g., RF and XGBoost) and neural networks (e.g., MLP, TCN, and LSTM) outperform HA and ARIMA in terms of MAE, RMSE, and MAPE due to better prediction capability. For the three neural network methods, TCN and LSTM achieve better prediction performance than MLP because they are more suitable for exploring correlations of features in time sequences. Compared with MLP, TCN, and LSTM, we find that their corresponding PDP-based model (PDP_MLP, PDP_TCN, and PDP_LSTM, respectively) achieve a significant reduction in MAE, RMSE, and MAPE. This indicates that, although a neural network integrated with the PDP framework is designed to forecast the distributions of future demand, it can also achieve higher accuracy in predicting the actual scalar demand than its counterpart. This is because PDP-based methods try to learn the latent distribution by involving a KL divergence term in the loss function, while scalar prediction models focuses on predicting the expectation of the labels. Learning the entire latent distribution is equivalent to imposing data augmentation on the label side, helping to provide a more robust demand range and enhance prediction performance. In addition, the integration of the double-hurdle mechanism

[6]https://pytorch.org/

TABLE III
OVERALL COMPARISON OF PDP-BASED METHODS AND BENCHMARK METHODS

Method	MPICP	MPINAW	MIS
PDP_MLP	0.972	0.084	0.093
PDP_TCN	0.972	0.081	0.090
PDP_LSTM	**0.973**	0.075	**0.084**
QReg	0.413	**0.068**	0.224

may help alleviate the problem of data imbalance, thereby improving prediction accuracy.

Next, we compare the proposed PDP-based neural networks with a classical method for range estimation, quantitle regression. Quantitle regression obtains the lower and upper bounds of the estimated range by taking two quantiles as labels; for example, an estimated range at a 90% confidence level takes the prediction result of the 5% quantile regression as the lower bound, and the prediction result of the 95% quantile regression as the upper bound. The PDP-based method can quickly derive the estimated range under any confidence level[7] after obtaining the predicted demand distribution, while the quantile regression method has to be retrained twice each time to obtain the estimated range under each confidence level.

The calculation of PICP, PINAW and IS depends on the estimated range, which is determined by a predefined confidence level. That is, each confidence level corresponds to a PICP, PINAW and IS. We compute PICP, PINAW and IS at different confidence levels—55%, 60%, $\cdots$, 90%, 95%—and then average them to get mean PICP (MPINP), mean PINAW (MPINAW), and mean IS (MIS), which are presented in Table III. We can observe that PDP-based methods achieve much higher accuracy in range (reflected by much larger MPICP) by sacrificing a little range width (reflected by slightly smaller MPINAW). With a smaller MIS, which is a more comprehensive assessment factor considering accuracy and range width simultaneously, the PDP-based models are shown to have a better performance than the quantile regression model. Figure 4 shows the comparison of estimated ranges, predicted demands and ground-truth demands for the three PDP-based neural networks and quantile regression (QReg) at the 80% confidence level. It can be observed that the demand range estimated by QReg fails to capture demand in peak hours and cannot well identify the lower bound of demand range. In contrast, the demand ranges estimated by the three PDP-based methods have higher accuracy in range and the lower bounds do not always remain 0, which is more reasonable.

E. Ablation Study

In this subsection, we verify the effectiveness of the proposed framework in terms of the integration of weather information, double-hurdle mechanism, inference strategy (i.e., the way we derive predicted demand from the predicted label distribution), and probability distribution type selection through a variety of ablation experiments.

[7]Here we treat P_{thre} as the equivalent of confidence level.

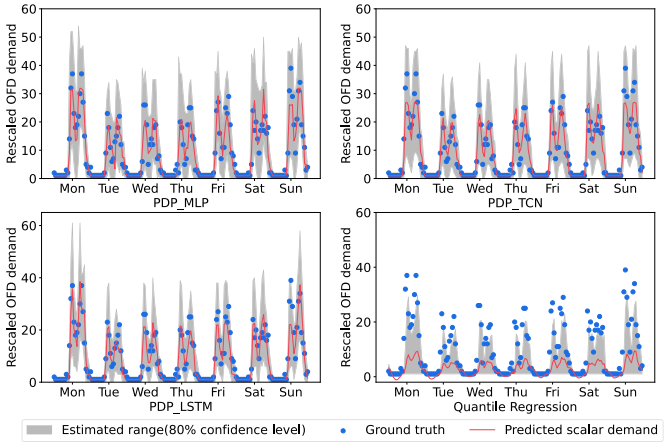


Fig. 4. Comparison of estimated ranges of PDP-based neural networks and Quantile regression over a test week.

TABLE IV
OVERALL COMPARISON OF PDP-BASED METHODS AND THEIR VARIANTS WITH HISTORICAL WEATHER SEQUENCES MISSING

Method	MAE	RMSE	MAPE	MPICP	MPINAW	MIS
PDP_MLP	0.717	1.728	18.973	0.972	0.084	0.093
NwMLP	0.753	1.815	22.241	0.991	0.095	0.098
PDP_TCN	**0.705**	**1.714**	**18.694**	0.972	0.081	0.090
NwTCN	0.785	1.942	22.046	**0.992**	0.096	0.099
PDP_LSTM	0.731	1.815	19.607	0.973	**0.075**	**0.084**
NwLSTM	0.756	1.882	21.218	0.989	0.090	0.096

1) Impact of Weather Information: To examine the impact of weather information on model performance, we compare three PDP-based methods and their variants. These variants include NwMLP, NwTCN, and NwLSTM, which are obtained by removing historical weather sequences from the model input. Table IV presents a comparison of PDP-based neural networks and their variants, those without historical weather sequences as input, across various metrics. The results indicate that the removal of weather information from model inputs results in a decline in the scalar prediction performance. Additionally, in the absence of weather information, we observe a notable increase in MPINAW accompanied by a slight increase in MPICP. As a consequence, the overall MIS score also increased, indicating a reduction in the comprehensive performance of range estimation. Figure 5 further illustrates the comparisons of estimated ranges at the 80% confidence level. Notably, when weather information is absent from the input, the model exhibits a considerable decrease in its capability to capture demand during peak hours.

2) Effect of Double-Hurdle Mechanism: As previously mentioned, the primary challenge in OFD demand forecasting is the issue of data imbalance. The majority of customers' OFD demand is concentrated during lunch and dinner time, resulting in high demand during peak hours and zero demand during many other time periods. To address this challenge, in addition to employing the LDL technique for label augmentation, we also incorporate a double-hurdle mechanism into the model to effectively identify a large number of cases where demand is zero. To evaluate the effectiveness of this double-hurdle mechanism, we first compare the proposed

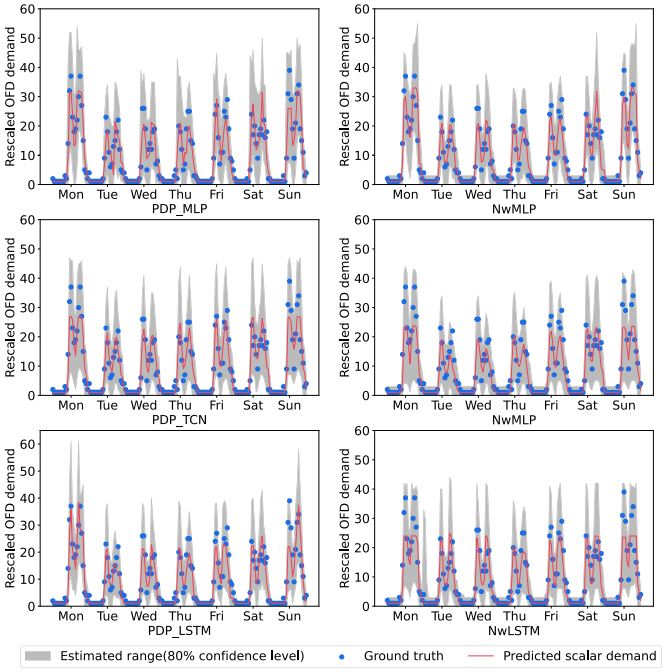


Fig. 5. Comparison of estimated ranges of PDP-based neural networks and their variants with historical weather sequences missing over a test week.

models with their variants where the double-hurdle mechanism is removed (NdMLP, NdTCN, and NdLSTM), as shown in Table V. The variants (NdMLP, NdTCN, NdLSTM) without the double-hurdle mechanism are essentially a combination of ordinary neural networks and the PDP framework.[8] To assess the impact of the proposed PDP framework on OFD demand forecasting performance, we first compare the three regression evaluation metrics of NdMLP/NdTCN/NdLSTM in Table V with those of MLP/TCN/LSTM in Table II. The results clearly demonstrate the substantial improvement achieved by incorporating the PDP framework. For example, NdMLP exhibits a 15% lower MAE, 12% lower RMSE, and 30% lower MAPE in comparison to the MLP model. This significant enhancement in performance highlights the effectiveness and potential of the PDP framework in improving OFD demand forecasting accuracy.

Our results in Table V also show a significant decline in performance for all three neural networks in terms of scalar prediction and range estimation when the double-hurdle mechanism is removed from the input, which provides evidence supporting the effectiveness of the double-hurdle mechanism in improving model performance. To further assess the effectiveness of the proposed method in addressing the data imbalance issue in Figure 6, we compare all six metrics of the proposed methods and their variants during peak hours (11 am to 1 pm and 5 pm to 7 pm) as well as off-peak hours. As shown in Figure 6, the prediction error of the model during peak hours is notably higher than that during off-peak hours. This observation confirms our previous statement that the long-tailed distribution makes peak demand challenging to

[8]For variants without the double-hurdled mechanism, we add 1 to all demands during model training and subtract 1 during testing.

TABLE V
OVERALL COMPARISON OF PDP-BASED METHODS AND THEIR VARIANTS WITH DOUBLE-HURDLE MECHANISM MISSING

Method	MAE	RMSE	MAPE	MPICP	MPINAW	MIS
PDP_MLP	0.717	1.728	18.973	0.972	0.084	0.093
NdMLP	0.749	1.779	21.484	0.990	0.101	0.106
PDP_TCN	**0.705**	**1.714**	**18.694**	0.972	0.081	0.090
NdTCN	0.748	1.742	21.565	**0.991**	0.096	0.099
PDP_LSTM	0.731	1.815	19.607	0.973	**0.075**	**0.084**
NdLSTM	0.769	1.868	21.616	0.990	0.096	0.101

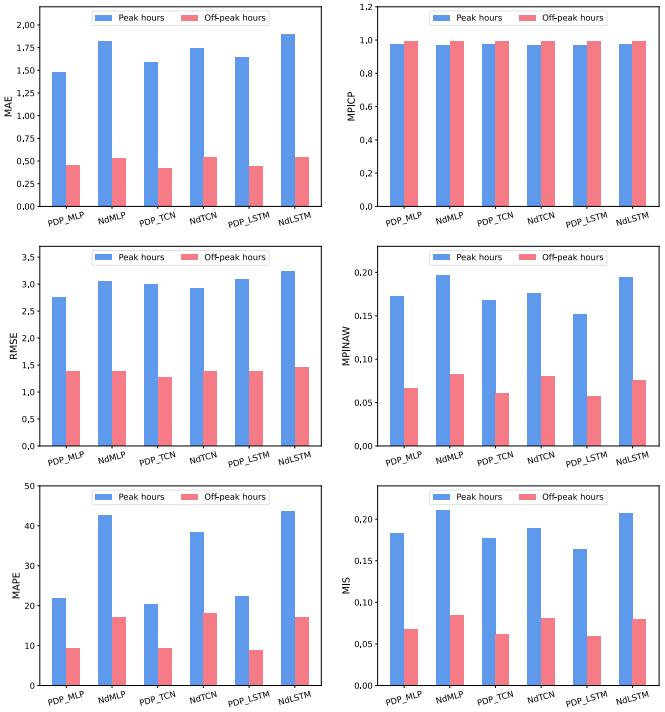


Fig. 6. Comparison of six metrics of PDP-based neural networks and their variants with double-hurdle mechanism missing during peak and off-peak hours.

predict accurately. And although PDP-MLP, PDP-TCN, and PDP-LSTM still exhibit higher prediction errors during peak hours than off-peak hours, we observe a significant reduction in error compared to their respective variants without the double-hurdle mechanism, from all metrics except MPICP. Notably, the improvement in performance during peak hours surpasses that during off-peak hours. This finding highlights the effectiveness of our proposed method in mitigating the impact of data imbalance.

3) Effect of Inference Strategy: In the proxy label distribution generation, we calibrate the parameter λ of the Poisson distribution with the real demand. For the Poisson distribution, both the expectation and the value corresponding to the maximum probability are close to λ. Therefore, both argmax and mean operations are reasonable when deriving a value from the predicted distribution as the predicted demand. The difference between the predicted demand and the real demand is a part of the final loss function, so the strategy of inferring $\hat{y}$ from $\hat{\boldsymbol{p}}$ has an impact on the performance of the proposed framework. To investigate the effectiveness of the argmax inference strategy, we compare the argmax operation

TABLE VI

PERFORMANCE COMPARISON UNDER DIFFERENT INFERENCE STRATEGIES

Method	MAE	RMSE	MAPE	MPICP	MPINAW	MIS
PDP_{argmax}	0.717	1.728	18.973	0.972	0.084	0.093
PDP_{mean}	0.754	1.645	27.445	0.993	0.098	0.101

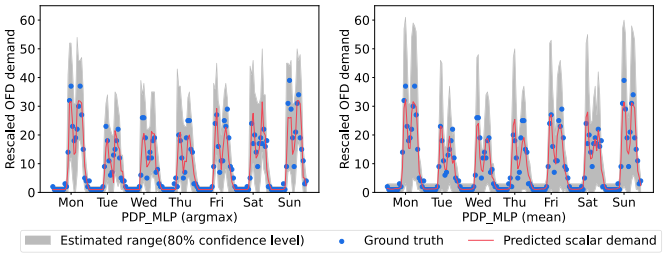


Fig. 7. Comparison of ground truth and estimated range over a test week under different inference strategies.

with the mean operation, which uses the mean values of the predicted distributions as predicted demand.

$$\hat{y}_{\text{mean}} = \sum_{k=1}^{|I|} k \cdot \hat{p}^k. \tag{14}$$

Table VI compares the results of PDP_MLP under different inference strategies. The RMSE of PDP_{argmax} is slightly higher than that of PDP_{mean}, but the MAE and MAPE are much lower, indicating that PDP_{argmax} achieves better scalar prediction performance. PDP_{argmax} also performs better on demand range estimation. The estimated ranges of PDP_{argmax} and PDP_{mean} have similar coverage (with similar MPICP), but PDP_{argmax} has a narrower range width (with smaller MPINAW), resulting in a lower MIS. Figure 7 compares the ground truth and estimated ranges with a 80% P_{thre} on the test set over a week. We can observe that the width of the estimated range of PDP_{argmax} is shorter than that of PDP_{mean}, and the estimated range of both methods can accurately cover all ground-truth demand observations during this week. These results demonstrate that the argmax operation is more effective than the mean operation in predicting the short-term demand range in our experiment.

4) Effect of Probability Distribution Type: In the proposed PDP framework, the proxy label distribution is assumed to be Poisson distribution. In real life, the actual distribution is unknown. In this subsection, we study the performance of PDP when the true distribution takes other forms. Three types of probability distributions—Gaussian distribution, Uniform distribution, and Negative binomial distribution—are tested and evaluated. Each of the three baseline distributions have two parameters. Since we calibrate one of the parameters by using the ground truth y, the other parameter must be treated as a hyperparameter, as shown in Table VII. It is worth noting that Poisson distribution does not bring a hyperparameter, because it only has one parameter λ, which can be calibrated by the ground truth y.

By changing the method of generating proxy label distribution, we can obtain a Gaussian-based distribution prediction (GDP) framework, a Uniform-based distribution prediction (UDP) framework, and a Negative binomial-based distribution prediction (NDP) framework. Table VIII presents different metrics achieved by PDP_MLP, GDP_MLP, UDP_MLP, and NDP_MLP, respectively. Although PDP_MLP has a slightly lower MPICP, it outperforms the other three distributions in both scalar prediction and range estimation. As σ increases, the scalar prediction of GDP_MLP is more accurate (which can be told from the first four metrics), but the width of the estimated range also increases rapidly, resulting in a significant increase in MIS. The same can be observed from UDP_MLP, whose performance on scalar prediction and range estimation conflict as c increases. The situation is different for the negative binomial distribution. When the probability of success p is equal to 0.4, the overall performance of NDP_MLP on the six evaluation metrics is better than the other two p settings (i.e., 0.3 and 0.5). This suggests that the probability of success under the Negative binomial assumption may be closer to 0.4. Compared with GDP_MLP, UDP_MLP, and NDP_MLP, which are greatly affected by hyperparameter settings in the probability distribution, the PDP_MLP does not require tunning of a hyperparameter (the only parameter is calibrated by the ground truth) and can avoid potential overfitting issues. In this sense, the models using Poission distribution should be more generalized than the models using the three baseline distributions.

Figure 8 shows the changes of the three range evaluation indicators under different P_{thre}. We can observe that while the PICP of PDP_MLP is slightly lower compared to other methods, all methods achieve high PICP values, exceeding 96%. The difference in PICP between PDP_MLP and other methods is less than 3%. However, the range width of PDP_MLP are consistently narrower than other methods regardless of the P_{thre} value, leading to a lower IS. We further visualize the estimated ranges of PDP_MLP and the other three distributions with optimal parameter settings that yield

TABLE VII

PARAMETER SETTINGS FOR DIFFERENT PROBABILITY DISTRIBUTIONS

Distribution	Parameter	
Gaussian	Mean: y	Variance: σ^2
Uniform	Low bound: $y-c$	Upper bound: $y+c$
Negative binomial	Number of successes: y	Probability of success: p

TABLE VIII

PERFORMANCE COMPARISON UNDER DIFFERENT PROBABILITY DISTRIBUTIONS

Method	MAE	RMSE	MAPE	MPICP	MPINAW	MIS
PDP	**0.717**	**1.728**	**18.973**	0.972	**0.084**	**0.093**
$\text{GDP}_{\sigma=2}$	0.938	2.277	28.897	0.987	0.100	0.110
$\text{GDP}_{\sigma=4}$	0.879	2.295	25.959	0.990	0.127	0.137
$\text{GDP}_{\sigma=6}$	0.812	1.988	24.668	**0.997**	0.154	0.156
$\text{UDP}_{c=2}$	0.792	1.984	22.201	0.992	0.116	0.120
$\text{UDP}_{c=4}$	0.814	2.005	24.508	0.993	0.131	0.134
$\text{UDP}_{c=6}$	0.853	2.023	26.162	0.989	0.107	0.112
$\text{NDP}_{p=0.3}$	0.949	2.563	25.608	0.987	0.128	0.144
$\text{NDP}_{p=0.4}$	0.731	1.770	19.901	0.969	0.103	0.111
$\text{NDP}_{p=0.5}$	0.917	2.221	27.772	0.987	0.098	0.107

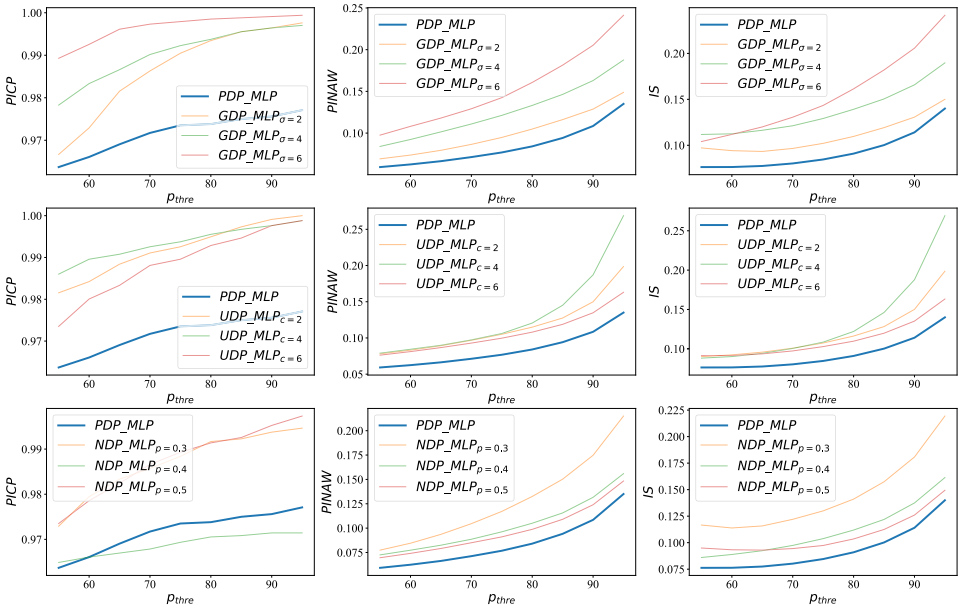


Fig. 8. Comparison of range estimation performance under different P_{thre} over a test week.

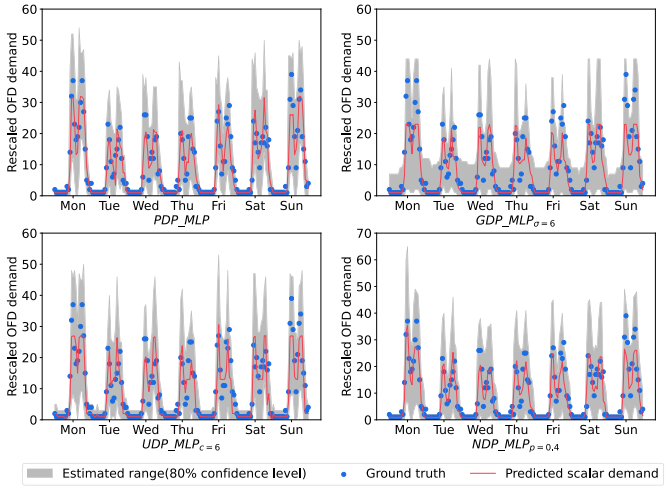


Fig. 9. Average reduction in MIE over different hours for PDP_MLP compared to GDP_MLP and UDP_MLP on the test set.

the minimum MIS. As shown in Figure 9, we can find that the reason for the wider demand ranges estimated by the other three distributions is different. GDP_MLP tends to estimate a higher demand upper bound during off-peak hours, which may be because the Gaussian distribution cannot well describe the sparse OFD demand. As the variance (i.e., σ) increases, the Gaussian distribution becomes more spread out, which results in a wider estimated range. On the contrary, UDP_MLP and NDP _MLP estimate demand ranges during peak hours with a higher upper bound and a lower bound closer to 0. The probability of demand occurring during peak periods (i.e., the probability of success p) is different from that during off-peak periods, which is difficult to capture by the Negative binomial distribution. The Uniform distribution assumes that the probability of OFD demand within a specific range of values is uniform. However, when the observed OFD demand falls outside this range, methods relying on the assumption of a uniform distribution may not yield satisfactory results. And thus UDP_MLP achieves poor performance during peak hours because demand peaks are often outside the interval in which most OFD demands fall. All in all, PDP_MLP can generally achieve comparable accuracy in range with narrower range width than the models based on other distributions.

V. Conclusion

In this paper, we study the problem of demand distribution forecasting for online food delivery platforms. We propose a novel Poisson-based distribution prediction (PDP) framework with a double-hurdle mechanism to tackle the issue of data imbalance. Our framework utilizes a neural network with multiple outputs to estimate the likelihood of zero demand and approximate label distribution. We use an uncertainty-based multi-task learning technique to strike a balance between BCE loss, KL divergence, and MSE loss. Extensive experiments are conducted based on a real dataset from a crowd-sourcing delivery platform in Asia. Experimental results show that PDP-based methods outperform several benchmarks by achieving more reliable demand range forecasting. Moreover, further ablation experiments highlight the effectiveness of the proposed PDP framework in improving demand forecasting performance, particularly during peak hours.

However, it is important to acknowledge the limitation regarding the lack of diverse OFD order datasets, which hinders a comprehensive assessment of our approach across different operating contexts. Further research should focus on obtaining and analyzing additional datasets to explore the generalizability of our method to various OFD scenarios. Another future research direction would be integrating advanced deep learning algorithms, such as graph convolutional neural networks, into the proposed PDP framework to further improve

the predictive performance. Moreover, expanding the proposed demand distribution learning framework to more scenarios, such as multi-step-ahead prediction, also merits investigation in future studies.

ACKNOWLEDGMENT

The data were supported by a crowd-sourcing delivery platform in Asia.

REFERENCES

[1] C. Gao et al., "A deep learning method for route and time prediction in food delivery service," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2021, pp. 2879–2889.

[2] S. Singh, "The soon to be $200B online food delivery is rapidly changing the global food industry," *Forbes*, vol. 9, Sep. 2019.

[3] H. Wang, "Transportation-enabled urban services: A brief discussion," *Multimodal Transp.*, vol. 1, no. 2, pp. 1–4, 2022.

[4] J. Ke, H. Zheng, H. Yang, and X. Chen, "Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach," *Transp. Res. C, Emerg. Technol.*, vol. 85, pp. 591–608, Dec. 2017.

[5] H. Wang and H. Yang, "Ridesourcing systems: A framework and review," *Transp. Res. B, Methodol.*, vol. 129, pp. 122–155, Nov. 2019.

[6] G. Xu et al., "Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 1. Worcester, MA, USA, 2019, pp. 3656–3663.

[7] G. Guo and T. Zhang, "A residual spatio-temporal architecture for travel demand forecasting," *Transp. Res. C, Emerg. Technol.*, vol. 115, Jun. 2020, Art. no. 102639.

[8] B.-B. Gao, C. Xing, C.-W. Xie, J. Wu, and X. Geng, "Deep label distribution learning with label ambiguity," *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 2825–2838, Jun. 2017.

[9] X. Geng, "Label distribution learning," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 7, pp. 1734–1748, Jul. 2016.

[10] H. Wang and A. Odoni, "Approximating the performance of a 'last mile' transportation system," *Transp. Sci.*, vol. 50, no. 2, pp. 659–675, May 2016.

[11] K. S. Shehadeh, H. Wang, and P. Zhang, "Fleet sizing and allocation for on-demand last-mile transportation systems," *Transp. Res. C, Emerg. Technol.*, vol. 132, Nov. 2021, Art. no. 103387.

[12] Y. Chen and H. Wang, "Pricing for a last-mile transportation system," *Transp. Res. B, Methodol.*, vol. 107, pp. 57–69, Jan. 2018.

[13] Z. Xu et al., "Large-scale order dispatch in on-demand ride-hailing platforms: A learning and planning approach," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 905–913.

[14] G. Lyu, W. C. Cheung, C.-P. Teo, and H. Wang, "Multi-objective stochastic optimization: A case of real-time matching in ride-sourcing markets," *Manuf. Service Oper. Manage.*, early access, Jul. 2023.

[15] H. Wang, "Routing and scheduling for a last-mile transportation system," *Transp. Sci.*, vol. 53, no. 1, pp. 131–147, Feb. 2019.

[16] H. Yang, C. Shao, H. Wang, and J. Ye, "Integrated reward scheme and surge pricing in a ridesourcing market," *Transp. Res. B, Methodol.*, vol. 134, pp. 126–142, Apr. 2020.

[17] J. Bai, K. C. So, C. S. Tang, X. Chen, and H. Wang, "Coordinating supply and demand on an on-demand service platform with impatient customers," *Manuf. Service Oper. Manage.*, vol. 21, no. 3, pp. 556–570, Jul. 2019.

[18] T. A. Taylor, "On-demand service platforms," *Manuf. Service Oper. Manage.*, vol. 20, no. 4, pp. 704–720, Sep. 2018.

[19] J. Ke, H. Yang, X. Li, H. Wang, and J. Ye, "Pricing and equilibrium in on-demand ride-pooling markets," *Transp. Res. B, Methodol.*, vol. 139, pp. 411–431, Sep. 2020.

[20] K. Zhang and Y. Nie, "To pool or not to pool: Equilibrium, pricing and regulation," *Transp. Res. B, Methodol.*, vol. 151, pp. 59–90, Sep. 2021.

[21] H. Sun, H. Wang, and Z. Wan, "Model and analysis of labor supply for ride-sharing platforms in the presence of sample self-selection and endogeneity," *Transp. Res. B, Methodol.*, vol. 125, pp. 76–93, Jul. 2019.

[22] X. Li, X. Li, H. Wang, J. Shi, and Y. P. Aneja, "Supply regulation under the exclusion policy in a ride-sourcing market," *Transp. Res. B, Methodol.*, vol. 166, pp. 69–94, Dec. 2022.

[23] J. D. Angrist, S. Caldwell, and J. V. Hall, "Uber versus taxi: A driver's eye view," *Amer. Econ. J., Appl. Econ.*, vol. 13, no. 3, pp. 272–308, 2021.

[24] Z. Zhu, J. Ke, and H. Wang, "A mean-field Markov decision process model for spatial–temporal subsidies in ride-sourcing markets," *Transp. Res. B, Methodol.*, vol. 150, pp. 540–565, Aug. 2021.

[25] A. Braverman, J. G. Dai, X. Liu, and L. Ying, "Empty-car routing in ridesharing systems," *Oper. Res.*, vol. 67, no. 5, pp. 1437–1452, Sep. 2019.

[26] E. Imani and M. White, "Improving regression performance with distributional losses," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 2157–2166.

[27] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, and L. Damas, "Predicting taxi–passenger demand using streaming data," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1393–1402, Sep. 2013.

[28] S. Carrese, E. Cipriani, L. Mannini, and M. Nigro, "Dynamic demand estimation and prediction for traffic urban networks adopting new data sources," *Transp. Res. C, Emerg. Technol.*, vol. 81, pp. 83–98, Aug. 2017.

[29] J. Tang, F. Liu, Y. Zou, W. Zhang, and Y. Wang, "An improved fuzzy neural network for traffic speed prediction considering periodic characteristic," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 9, pp. 2340–2350, Sep. 2017.

[30] R. Xue, D. Sun, and S. Chen, "Short-term bus passenger demand prediction based on time series model and interactive multiple model approach," *Discrete Dyn. Nature Soc.*, vol. 2015, pp. 1–11, Mar. 2015.

[31] T. Zhang and G. Guo, "Graph attention LSTM: A spatiotemporal approach for traffic flow forecasting," *IEEE Intell. Transp. Syst. Mag.*, vol. 14, no. 2, pp. 190–196, Mar. 2022.

[32] C. Li, L. Bai, W. Liu, L. Yao, and S. T. Waller, "Knowledge adaption for demand prediction based on multi-task memory neural network," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 715–724.

[33] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1–7.

[34] H. Yao et al., "Deep multi-view spatial–temporal network for taxi demand prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 2588–2595.

[35] X. Zhou, Y. Shen, L. Huang, T. Zang, and Y. Zhu, "Multi-level attention networks for multi-step citywide passenger demands prediction," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 5, pp. 2096–2108, May 2021.

[36] Y. Liu, C. Lyu, Y. Zhang, Z. Liu, W. Yu, and X. Qu, "DeepTSP: Deep traffic state prediction model based on large-scale empirical data," *Commun. Transp. Res.*, vol. 1, Dec. 2021, Art. no. 100012.

[37] C. Zhang, F. Zhu, X. Wang, L. Sun, H. Tang, and Y. Lv, "Taxi demand prediction using parallel multi-task learning model," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 794–803, Feb. 2022.

[38] K. Zhang, Z. Liu, and L. Zheng, "Short-term prediction of passenger demand in multi-zone level: Temporal convolutional neural network with multi-task learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1480–1490, Apr. 2020.

[39] G. Guo, W. Yuan, J. Liu, Y. Lv, and W. Liu, "Traffic forecasting via dilated temporal convolution with peak-sensitive loss," *IEEE Intell. Transp. Syst. Mag.*, vol. 15, no. 1, pp. 48–57, Jan. 2023.

[40] J. Ke et al., "Hexagon-based convolutional neural network for supply-demand forecasting of ride-sourcing services," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 11, pp. 4160–4173, Nov. 2019.

[41] Y. Wang, H. Yin, H. Chen, T. Wo, J. Xu, and K. Zheng, "Origin-destination matrix prediction via graph convolution: A new perspective of passenger demand modeling," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 1227–1235.

[42] C. Zheng, X. Fan, C. Wang, and J. Qi, "Gman: A graph multi-attention network for traffic prediction," in *Proc. AAAI*, vol. 34, no. 1, 2020, pp. 1234–1241.

[43] J. Ke, X. Qin, H. Yang, Z. Zheng, Z. Zhu, and J. Ye, "Predicting origin-destination ride-sourcing demand with a spatio-temporal encoder–decoder residual multi-graph convolutional network," *Transp. Res. C, Emerg. Technol.*, vol. 122, Jan. 2021, Art. no. 102858.

[44] J. Tang, J. Liang, F. Liu, J. Hao, and Y. Wang, "Multi-community passenger demand prediction at region level based on spatio-temporal graph convolutional network," *Transp. Res. C, Emerg. Technol.*, vol. 124, Mar. 2021, Art. no. 102951.

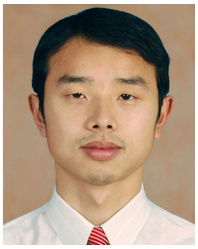
[45] J. Ke, S. Feng, Z. Zhu, H. Yang, and J. Ye, "Joint predictions of multi-modal ride-hailing demands: A deep multi-task multi-graph learning-based approach," *Transp. Res. C, Emerg. Technol.*, vol. 127, Jun. 2021, Art. no. 103063.
[46] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph WaveNet for deep spatial–temporal graph modeling," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 1907–1913.
[47] D. Zhang, F. Xiao, M. Shen, and S. Zhong, "DNEAT: A novel dynamic node-edge attention network for origin-destination demand prediction," *Transp. Res. C, Emerg. Technol.*, vol. 122, Jan. 2021, Art. no. 102851.
[48] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 17804–17815.
[49] A. Hess, S. Spinler, and M. Winkenbach, "Real-time demand forecasting for an urban delivery platform," *Transp. Res. E, Logistics Transp. Rev.*, vol. 145, Jan. 2021, Art. no. 102147.
[50] A. Pujara, V. Pattabiraman, and R. Parvathi, "Food demand forecast for online food delivery service using catboost model," in *Proc. 3rd EAI Int. Conf. Big Data Innov. Sustain. Cognit. Comput.* Cham, Switzerland: Springer, 2022, pp. 129–142.
[51] X. Geng, Q. Wang, and Y. Xia, "Facial age estimation by adaptive label distribution learning," in *Proc. 22nd Int. Conf. Pattern Recognit.*, Aug. 2014, pp. 4465–4470.
[52] X. Geng and Y. Xia, "Head pose estimation based on multivariate label distribution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1837–1842.
[53] X. Geng and P. Hou, "Pre-release prediction of crowd opinion on movies by label distribution learning," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, Jul. 2015, pp. 3511–3517.
[54] X. Geng, C. Yin, and Z.-H. Zhou, "Facial age estimation by learning from label distributions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 10, pp. 2401–2412, Oct. 2013.
[55] B.-B. Gao, H.-Y. Zhou, J. Wu, and X. Geng, "Age estimation using expectation of label distribution learning," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 712–718.
[56] C. Singh, G. N. Singh, and J.-M. Kim, "A randomized response model for sensitive attribute with privacy measure using Poisson distribution," *Ain Shams Eng. J.*, vol. 12, no. 4, pp. 4051–4061, Dec. 2021.
[57] J. G. Cragg, "Some statistical models for limited dependent variables with application to the demand for durable goods," *Econometrica, J. Econ. Soc.*, vol. 38, no. 5, pp. 829–844, 1971.
[58] D. C. Luvizon, H. Tabia, and D. Picard, "Human pose regression by combining indirect part detection and contextual information," *Comput. Graph.*, vol. 85, pp. 15–22, Dec. 2019.
[59] R. Cipolla, Y. Gal, and A. Kendall, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7482–7491.
[60] P. Mpfumali, C. Sigauke, A. Bere, and S. Mulaudzi, "Day ahead hourly global horizontal irradiance forecasting—Application to South African data," *Energies*, vol. 12, no. 18, p. 3569, Sep. 2019.
[61] A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya, "Lower upper bound estimation method for construction of neural network-based prediction intervals," *IEEE Trans. Neural Netw.*, vol. 22, no. 3, pp. 337–346, Mar. 2011.
[62] T. Gneiting and A. E. Raftery, "Strictly proper scoring rules, prediction, and estimation," *J. Amer. Stat. Assoc.*, vol. 102, no. 477, pp. 359–378, Mar. 2007.
[63] J. Gasthaus et al., "Probabilistic forecasting with spline quantile function RNNs," in *Proc. 22nd Int. Conf. Artif. Intell. Statist.*, 2019, pp. 1901–1910.

Jian Liang received the B.S. and M.S. degrees in traffic and transportation engineering from Central South University. He is currently pursuing the Ph.D. degree with the Department of Civil Engineering, The University of Hong Kong. His research interests include spatio-temporal forecasting and intelligent transportation systems.

Jintao Ke received the B.S. degree in civil engineering from Zhejiang University and the Ph.D. degree in civil and environmental engineering from The Hong Kong University of Science and Technology. He is currently an Assistant Professor with the Department of Civil Engineering, The University of Hong Kong. His research interests include spatial–temporal traffic forecasting, shared mobility (ridesourcing and ride-sharing services), transportation economy, machine learning in transportation, and urban computing.

Hai Wang received the B.S. degree from Tsinghua University and the dual master's degree in operations research and transportation and the Ph.D. degree in operations research from MIT. He is currently an Associate Professor with the School of Computing and Information Systems, Singapore Management University, and a Visiting Professor with the Heinz College of Information Systems and Public Policy, Carnegie Mellon University. His research interests include methodologies on operations research, data-driven modeling, computational algorithms, and machine learning, and relevant applications in smart city, innovative transportation, advanced logistics, modern e-commerce, and intelligent healthcare systems.

Hongbo Ye received the B.Eng. degree in automation from the University of Science and Technology of China and the Ph.D. degree in civil engineering from The Hong Kong University of Science and Technology. He is currently a Research Assistant Professor with the Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University. His research interests include transportation optimization and management, day-to-day dynamics, and eco-driving of trains and cars.



Jinjun Tang received the Ph.D. degree in transportation engineering from the Harbin Institute of Technology, Harbin, China, in 2016. From 2014 to 2016, he was a Visiting Scholar with the Smart Transportation Applications and Research Laboratory (STAR Laboratory), University of Washington, Seattle, WA, USA. He is currently an Associate Professor with the School of Traffic and Transportation Engineering, Central South University, Changsha, China. He published more than 50 technical articles in journal as the first author and corresponding coauthor. His research interests include traffic flow prediction, data mining in the transportation systems, intelligent transportation systems, and transportation modeling.