Taxi Deployment System
—A Strategic Model from Machine Learning

Yuxin Zhang[1], Xiyan Cai[1], Yixiang Xiao[1],
1. New York University Shanghai
{yz3402, xc984, yx1215}@nyu.edu

***Abstract*—This paper explores a strategic taxi deployment decision model based on two prediction models on taxi demand and traffic congestion. In this paper, we used a stacking assembling approach to assemble a bunch of base prediction model for better prediction accuracy and reduced overfit performance. Using the dataset in Haikou, Hainan province, we eventually achieved test MSE of 0.1230 and 0.7927 in order demand and congestion prediction, respectively.**

I. Introduction

As cities prosper, transportation becomes an increasingly troublesome issue for the general population living in the city as well as the administrative officials. In metropolis like Haikou in Hainan Province, we can very often see cars, buses, even bikes stuck in the middle of the road, not being able to move a bit. These are currently generating problems for all participants of the society: the commuters, the taxi drivers, putting more and more pressure on public transportation. That's why an optimizing vehicle deployment system is beneficial to the flow of people in the city. It can not only save productive time for people, but also maximize the resource allocation inside the social system, thus increasing efficiency inside of it.

During rush hours, taxi companies face a dilemma of large number of demand and terrible traffic jams when considering dispatching taxis. Our project targets to provide a solution that dispatches taxis based on the congestion conditions and the taxi demand in a certain area.

As a problem of common concern, many parties have prompted certain approaches to tackle this problem.According to our research, local government usually handle this issue by building more public transportation infrastructures to reduce the amount of vehicles run on the road. This method works but has a diminishing effect and would eventually reach a plateau, as demand for faster, personal and precise transportation retains. On the business side, there are also many transportation companies and new entrepreneurs trying to create business opportunities by tackling this issue. And the leading tech-based transportation platform provider in China Didi, is one of them.Didi utilizes its large-scale datasets to dispatch taxis based on "Order Dispatch System" [1]. "Order Dispatch System" is aimed at improving the efficiency of the whole system instead of shortening the waiting time of a certain order. However, it ignores to consider the traffic conditions of that area, which may greatly influence the computation of predicted waiting time.

Prediction regardless of traffic conditions is at the risk of exceeding customers' tolerance to wait. Therefore, having a dispatch model that takes traffic conditions into consideration is of importance. Our taxi deployment optimization first divides a large area into small hexagons. Based on the traffic conditions at each small hexagons at a certain time, it predicts the number of taxis that is appropriate to be dispatched to each hexagon (See Fig. 1). It should have the results of both satisfying the order demand and also avoiding the traffic jams [2]. Also, the "Order Dispatch System" is a more tactical system which gives directions on order dispatch only when orders actually comes in. However, in the administrative perspective, it is definitely better to also consider the strategic deployment taxis first, as being strategic means to have knowledge on potential orders before it actually happens.

In all, comparing to previous works done by Didi and other research groups, our work focuses on a more strategic perspective on taxi deployment instead of the tactical dispatching of taxis upon order.

II. Method

A. Datasets and Feature Extraction

Upon the consideration of building up a strategic taxi deployment system, we figured out to derive two prediction models to tackle this problem, one for demand order prediction and the other for traffic situation prediction. And we are going to use the historical order data to predict the order demand, and use the historical traffic data to predict the congestion situation.

In addition to considering the time and location, which affect the taxi demand and traffic, the weather is an external factor that can never be ignored. The research by Fieremans et al. shows that the weather does affect drivers' income, which indicates its influence on the number of orders. And to our surprise, Snow conditions, in contrast, do not necessarily increase the hourly revenues compared to clear weather conditions [3]. Therefore, we decide to also include the weather data to assist with the model specifications.

There are three datasets
1) Didi taxi orders in Haikou from May to October in 2017 (See Table I)
2) Travel Time Index (TTI) in Haikou from May to October in 2017 (See Table II)

3) Hourly weather conditions in Haikou from May to October in 2017 (See Table III).
Dataset 1 and Dataset 2 are from Didi Gaia Open Database [4] and Dataset 3 is from 911 Weather Report by web crawling [5].

Table I
FEATURES IN DATASET 1

Feature	Description
order_id	the unique identifier of taxi orders
product_id	1: Didi special car 2: Didi company car 3. Didi company express
city_id	just one city Haikou in this dataset
district	index of districts in Haikou
county	index of level-2 districts in Haikou
type	1: appointed order 2: instant order
combo_type	1: not share with others 4: share with others
traffic_type	1: hourly renting by companies 2: package for company airport pickup 3: package for company airport drop-off 4: sharing cars 5: airport pickup 6: airport drop-off 302: sharing a car across cities
passenger_count	only for car sharing, the number of passengers set by each customer
driver_product_id	the product group that the driver belongs to
start_dest_distance	road distance between starting location and destination
arrive_time	the time when the driver presses arriving the destination
departure_time	the time when the driver presses starting counting the money
pre_total_fee	the estimated fee when the passenger orders
normal_time	the time spent
product_1level	1: special car 2: express car 3: luxurious car
dest_lng	destination longitude
dest_lat	destination latitude
starting_lng	starting location longitude
starting_lat	starting location latitude

Table II
FEATURES IN DATASET 2

Feature	Description
obj_id	TTI object indext
batch_time	the time when obtains the record
tti	Travel Time Index
speed	average speed
obj_name	the object location description
geom	the geographic orientation

In Dataset 2, the calculation of TTI and speed are as:

$$TTI = \frac{\sum_{i=1}^{N} \frac{L_i}{V_i} \cdot W_i}{\sum_{i=1}^{N} \frac{L_i \cdot W_i}{V_{free\_i}}} \quad (1)$$

$$speed = \frac{\sum_{i=1}^{N} L_i \cdot W_i}{\sum_{i=1}^{N} \frac{L_i \cdot W_i}{V_i}} \quad (2)$$

where L_i is the length of i-th road link, W_i is the weight of the road link, V_i is the real-time speed and V_{free_i} is the freeflow of the link.

Table III
FEATURES IN DATASET 3

Feature	Description
time	hourly based time
wind scale	average wind scale in one hour
precipitation	average precipitation in one hour
sensible temperature	average sensible temperature in one hour
weather	general weather condition in one hour
temperature	average temperature in one hour
humidity	average humidity in one hour
wind direction	general wind direction in one hour

There are three dimensions to decide the number of taxis to be dispatched to each hexagon: demand indicator, traffic indicator and external factors, which are in correspondence to three datasets. We extract order_id, departure_time, dest_lng, dest_lat, starting_lng, starting_lat from dataset 1) as demand indicators (See Table I); batch_time, geom, speed, tti from dataset 2) as traffic indicator (See Table II); time, wind scale, precipitation, sensible temperature as external factors (See Table III).

B. Section Setup And Approach Structure

Considering the geographic features of specific historical order data and that regarding the taxi deployment, we consider setting up sections as prediction targets to capture demand and congestion specifics. We eventually decided to use the Uber's Hexagon section breakdown approach(H3 package) because we think it has the following advantages [6]:

- Total coverage of the geographic area under study
- Standardized Object with equal distance to the corner from the center
- Size flexible in tuning using Uber's H3 library

Figure 1. Logic Model Approaches

After analyzing the purpose of the deployment system and studying the general feature of the congestion and order pattern, we decide to set different size of Hexagon for the

order prediction and the congestion prediction model. This is for the following reasons.

- Congestion variations are more sensitive to the external factors
- We would like to deploy the taxi at the place where demand is satisfied and also congestion is minimized

For these reasons, we decided to set the order sections bigger (taking i = 7 in H3 library) and set the congestion sections smaller (taking i =8 in H3 library). So the specific approach for our deployment system model is divided into three parts. Firstly, we predict the demand order in larger section. Secondly, we predict the congestion index in smaller sections. Finally, we combine two models geographically and deploy the amount of taxis at the place that achieve the minimal congestion index within each larger section (See Fig. 2).

Figure 2. Logic Model Approaches

C. Data Processing

Considering the essence of the problem we are going to solve, we consider using two prediction models, one predicting the demand within larger sections, one predicting the congestion situation within small sections. The hints for how we derive our methodology lie in both our study of the traffic jam and taxi orders in real life, and the preliminary analyses of the data we obtained from different resources. First, aiming at deploying taxis in an efficient way, it's intuitive for us to study and predict the demand. Secondly, using the time-series geographic distribution of all orders (indicating the historical demand), we could detect by eye two features of the data—one is that the distribution pattern and density changes with time; the other is that there are sectional geographic patterns. That's why we add two time-related attributes(hour_of_day and day_of_week), and a sectional dummy(Hexagon_id). However, we want our deployment system to work efficiently and flexibly, not just to satisfy the demand, but to also consider the detailed traffic situation within each demand sections. As demand is usually time and geographic dependent, traffic situations are more sensitive to other external variables such as the weather, traffic accidents, road restrictions, business times. That is why we decided to study and predict the different sectional traffic situation (by the variable congestion_index) within each demand larger section. In this way, we could optimize our deployment decision as for how many and where at given time, in a flexible and intuitive way.

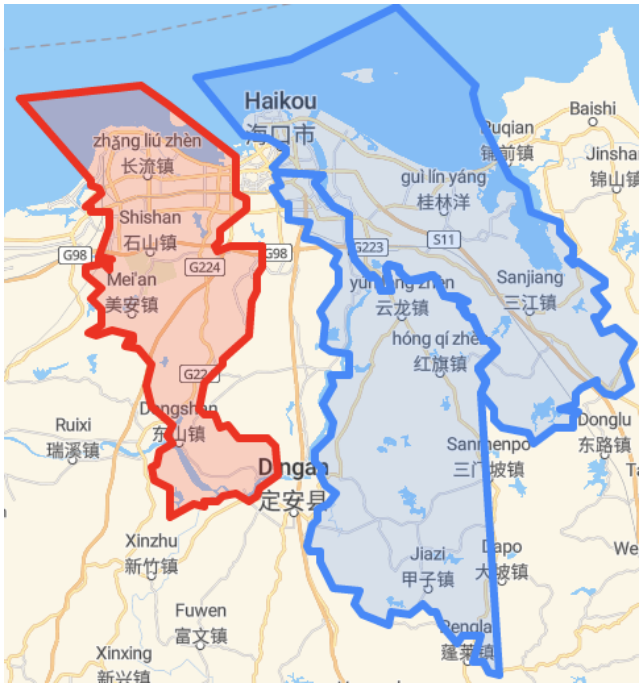

Figure 3. The 3 districts we choose to reduce dimension (number of hexagons)

For our demand prediction model, we have the historical hourly order number within each hexagon as the predicted value (demand). And we have six other attributes to predict it (3 weather indicators, time_of_day, day_of_week, Hexagon1_id).

For our traffic situation prediction model, we have the historical hourly congestion index within each smaller hexagon as the predicted value(Congestion_Index). And we have six other attributes to predict it (3 weather indicators, time_of_day, day_of_week, Hexagon2_id).

We have to transfer time_of_day, day_of_week and hexagon ids into dummy variables, which resulted in huge sizes of datesets and problems in training locally. Therefore, we only focus on the data in June and 3 districts in the city (See Fig. 3).

Besides, we also transfer the data types from int64 to int8 and from float64 to float16 to save memory. As a result, the size of the order dataset is 17555×128 and the size of congestion dataset is 372973×576.

D. Cross Validation And Mean Square Error

K-fold Cross-validation (CV) is widely applied method to estimate model performance [7]. Here, we use 5-fold cross validation with stratified CV split, which splits the entire dataset into 5 folds with the same distribution of labels [7]. We quantize this range to 5 bins and each fold would have the same percentage of the samples in each bin, as the whole dataset.

In addition, we use Mean Square Error as the loss function, which is a standard way to estimate the goodness of the

Table IV
RESULTS OF BASE MODELS

Prediction Model	Model	Best Training MSE	Best Validation MSE	Best Parameters
Order	Average	1.0063	0.9749	-
	SVM	0.4196	0.4209	$Degree = 1$ $C = 0.8860$ $\epsilon = 0.054$
	Decision Tree	0.0919	0.1716	$max\_depth = 9$
	NN	0.0872	0.1097	$hidden\_size = 50$ $weight\_decay = 0.01$
Congestion	Average	0.9385	1.2450	-
	SVM	3.1142	3.3422	$Degree = 1$ $C = 1.0$ $\epsilon = 1.5$
	Decision Tree	0.9271	0.9907	$max\_depth = 2$
	NN	0.5600	0.8808	$hidden\_size = 50$ $weight\_decay = 0.005$

Table V
FINAL RESULTS TWO PREDICTION MODELS

Prediction Model	Training MSE	Validation MSE	Test MSE	Baseline MSE
Order	0.0842	0.1280	0.1230	0.9664
Congestion	0.5600	0.8808	0.7927	0.8120

model.

E. Base Model and Stacking

Based on our dataset features, we handpicked Supporting Vector Machine (SVM), Decision Tree, and Neural Network (NN) to be our base models for the stacking purpose. We attain the best base models by training each of them on the training dataset and use the validation dataset to tune the parameters for each base model. Therefore, for each base model for each prediction model, the validation error is minimized (using MSE as the loss function).

After getting the base models, as for the order prediction, we stacked all the best base models for each prediction function and attain the meta-learner by regress the training and validation dataset on the base models. In this way, we obtain a model well-trained to capture different features in the dataset and ensures model stability by cross validation by the validation sets. As for the congestion predition, after rounds of practices, since the dataset is too large to be trained locally. Even on HPC, the program crashes before the administrator withdraws the task. Therefore, we decide to stick to the NN model which yields a better result in validation.

F. Baseline Model: Average

Besides, the model we use as a baseline to roughly evaluate our stacking model is the average. Basically, we use the average of the y's in our training set as the predicted value and calculate the loss.

III. RESULTS AND DISCUSSION

A. Base Model Results

Table. IV summarizes the best training scores, best validation scores and best parameters of four base models:

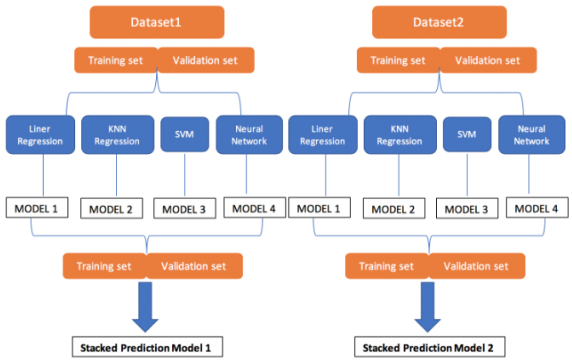

Figure 4. The idea of stack

average, SVM, Decision Tree and NN.

The MSE scores vary among different base models. However, each model has already obtained its best parameters according the the grid search.

B. Final Results

We take the stack in order prediction while use NN in congestion prediction. Table. V shows the final results.

The order prediction result shows little overfitting, which indicates the stability of the stacking model. NN will yield a really positive result if the parameters are tuned properly while the process of tuning is painful. Both prediction models beat the baseline.

IV. Conclusion

A. Program Output

Provided with the day of the week, time of the day and the weather condition, the program is able to generate the deployment of taxis to satisfy the demand without intensifying the traffic congestion. For instance, given Thursday, 18:00, $temperature = 40$, $wind\_speed = 7$ and $precipitation = 0.0$, the model suggests predicted demand of taxis should be sent to each of the four districts are (50, 83, 280, 310), with color encoding showing the demand density. And the congestion prediction model in the middle shows the predicted congestion situation in smaller sections given the same weather and time condition, with color encoding showing the TTI level. Combing two prediction results, we arrive at the model-predicted deployment strategy that we choose the least congested position (encoded the lightest by color) to deploy taxis within each larger section (cycled by red lines), as the picture on the right shows. (See Fig. 5).

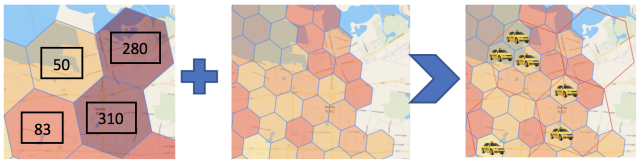

Figure 5. Test case Logistics and deployment decision

B. Future Work

The impact of the project is obviously improving the commuting efficiency. According to the opportunity cost of each person, the money they save is significant.

However, limited by the computation power, we are currently unable to achieve a higher accuracy by using larger dataset. In the future, more rows of data should be added to reduce variance. In addition, year-wide data can be added to use month dummies to capture the order and congestion seasonality. As for the data setup, the hexagon size can actually be treated as a hyper-parameter to be tuned to explore the best section size for models. Last but not the least, more base models or PCA methods can be applied to reach a better prediction performance.

Acknowledgments

This project is assigned by Professor Enric Junque de Fortuny and Instructor Ruowen Tan in CSCI-SHU 360 Machine Learning Fall 2019.

References

[1] L. Zhang, T. Hu, Y. Min, G. Wu, J. Zhang, P. Feng, P. Gong, and J. Ye, "A taxi order dispatch model based on combinatorial optimization." 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017, pp. 2151–2159.

[2] C. Kamga, M. A. Yazici, and A. Singhal, "Hailing in the rain: Temporal and weather-related variations in taxi ridership and taxi demand-supply equilibrium," 01 2013.

[3] E. Fieremans, J. H. Jensen, and J. A. Helpern, "Hailing in the rain: Temporal and weather-related variations in taxi ridership and taxi demand-supply equilibrium," *Neuroimage*, vol. 58, no. 1, pp. 177–188, 2011.

[4] "Didi chuxing gaia initiative," https://gaia.didichuxing.com.

[5] "911 weather report," https://tianqi.911cha.com.

[6] "H3: Uber's hexagon hierarchical spacial index," https://tianqi.911cha.com.

[7] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, ser. Springer Series in Statistics. Springer New York, 2013. [Online]. Available: https://books.google.com/books?id=yPfZBwAAQBAJ