Explainable Demand Prediction for Ride-Hailing Services

Elif Arslan[a,*], Emir Demirović[b], Marco Rinaldi[a], Oded Cats[a] and Serge Hoogendoorn[a]

[a]Department of Transportation and Planning, Delft University of Technology, Netherlands
[b]Department of Computer Science, Delft University of Technology, Netherlands

Abstract. The emergence of ride-hailing services has introduced computational challenges in optimizing vehicle routes. One of these challenges is solving the routing problem for long-term periods, as it requires accurate and explainable demand predictions. This is because explainability influences solution trustworthiness while accuracy determines final optimization outcomes. To address this, we employed Multi-Objective Optimal Regression Tree (MOO-RT) with accuracy and explainability objectives. Comparative analysis against other models, utilizing accuracy (MSE) and explainability (global Shapley Value) metrics, demonstrated MOO-RT's impact on enhancing explainability whereas Random Forest emerged as the top performer in accuracy. Moreover, using Shapley values our study identified model-independent significant features for passenger prediction.

1 Introduction

In the last decade, transportation services have become more accessible as the use of the GPS technology in mobile applications has facilitated supply-demand matching. This has led to the emergence of ride-hailing services, bringing novel challenges such as handling dynamic ride requests and real-time solution generation. Stochasticity of requests plays a role in complexity, as the future requests' time and pick-up/drop-off locations are not known a priori. This issue is addressed by anticipatory vehicle routing methods (e.g. [1] and [12]) to find solutions that are optimum for longer time horizons. Sub-optimal solutions have ramifications for all parties involved in the services: having negative implications for passengers' level-of-service and drivers' income as well as platform's profitability.

When anticipatory vehicle routing methods use predictions to determine future ride requests, the selection of the prediction model becomes crucial. We argue that accuracy and explainability play a key role in final solution quality, since: 1) the accuracy level of the predictions directly affects the routing solutions, as inaccurate predictions can guide the vehicles to locations with no ride requests in the near future, causing supply-demand imbalance 2) the prediction method's explainability can determine the overall vehicle routing framework's explainability. We define explainability as the *ability to convey to system users why a particular decision is made*. If a prediction method is not explainable, it may not be trusted by its users, potentially causing it to be overlooked and not utilized while making routing decisions. Additionally, explainable models can serve a dual purpose: not only as predictive tools but also as valuable aids for users when the data used for decision-making contains errors. This dual role arises from clarifying unexpected model outcomes and detecting inconsistencies within the data. A clear understanding of how the predictions are made is essential for these tasks.

Despite the growing acknowledgment of the importance of explainability ([10]), the main focus of the literature concerned with passenger predictions has been on evaluating the accuracy of the methods while explainability is rarely or at best implicitly mentioned. In fact, in passenger prediction literature, a significant number of recent studies use deep learning, and more specifically Convolutional Neural Networks and Long Short-Term Memory Networks (such as [6], [16] and [18]). The reason behind these model choices is to model complex spatial relations and sequential interactions ([16]). However, these models' structures are too complex to be understood by the users. Although the importance of model explainability is sometimes acknowledged ([16]), in these studies model comparison is done only for accuracy evaluation.

Traditional prediction models such as Linear Regression (e.g. [7], [9]), Random Forest, (e.g. [5], [7], [13]), ARIMA (e.g. [4], [11]) and other time series models (e.g. [2], [17]) are widely used in passenger prediction literature as well. These prediction methods are generally easier to understand compared to the deep learning methods and hence can be considered more explainable. Nevertheless, similar to deep learning models, traditional prediction methods are not evaluated for their explainability although their explainability may be determined by model parameters or characteristics. For instance, the average feature significance level of a prediction model could serve as a measure of its explainability. However, in the literature (e.g. [7], [13] and [14]) the determination of feature significance typically focuses on the accuracy or on presenting the resulting model whereas how informative an average feature is can be a proxy for the quality of prediction explanation.

Regression trees are one of the few models in the field of Explainable AI (XAI) literature, thanks to their graphical structure. This structure facilitates explanation as it is easy to convey how a prediction is made or to detect data errors. Figure 1 and Figure 2 exemplify the data error detection case. The regression tree in Figure 1 predicts the hourly passenger demand in the Manhattan district of New York City. Using the regression tree, passenger demand at *Hamilton Heights* is predicted to be 1500 or 2000 passengers per hour (depending on the weather conditions), which is significantly higher than the area's historical demand. When we look at the historical passenger demand heat map in Figure 2, it is clear that *Hamilton Heights* (the criss-crossed area on the map) is one of the

* Corresponding Author. Email: e.arslan-1@tudelft.nl.

areas in North Manhattan with a relatively low passenger demand. However, *Hamilton Heights'* predictions lie in the sub-tree with [*The area is in North Manhattan* = False]. Hence, it can be concluded that *Hamilton Heights' "The area is in North Manhattan"* feature is incorrectly valued as *False* and data error caused notable training error for the area.

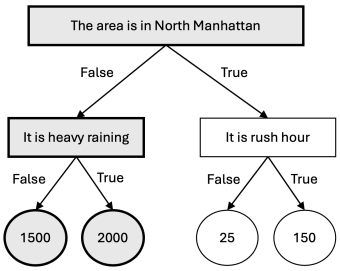

Figure 1. *Regression tree for Manhattan passenger prediction.* Values on the leaf nodes represent number of predicted passengers per hour. Highlighted (gray shaded) nodes are used for *Hamilton Heights* area passenger prediction.

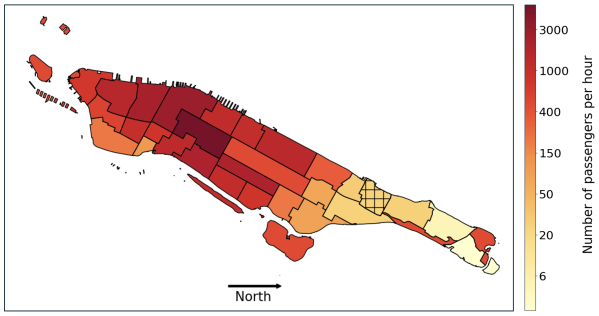

Figure 2. *Heat map of average Manhattan passenger demand (in number of passengers per hour) for the period January 2010 - October 2010. Hamilton Heights* area is criss-crossed with lines in the map.

Even though this case can be easily detected using regression trees, achieving similar reasoning using other explainable models such as linear regression, particularly when multiple features are involved, can be quite challenging. Despite the strong indications of regression trees' explainability, these have not been featured in the passenger prediction literature. To bridge this research gap, this study evaluates the accuracy and explainability performance of regression trees.

Thanks to the immediate relationship between regression trees' topology and their level of explainability, the regression tree explainability can be directly quantified easily, and thereby explicitly considered when training the prediction model. To capitalise on this capability, we employ *multi-objective optimal regression trees (MOO-RT)* in our prediction task, using both explainability and accuracy in the objective functions. This way, we do not only train regression trees with quantified accuracy and explainability performances but can also evaluate the trade-off between accuracy in explainability in passenger prediction, and determine whether the model can be employed for passenger demand estimation purposes.

For the sake of comparing MOO-RT with other explainable models, we train other white-box prediction models. Accuracy is measured through the Mean Squared Error metric, while explainability is evaluated via Shapley values as proposed in [8]. Shapley values are widely used model-agnostic explainability metrics, and they determine the marginal contribution of features in the prediction value. After calculating the model explainability and accuracy metrics, we investigate the trade-off between these two objectives, and through Shapley values, we determine model-independent significant features.

2 Methodology

In this section, we explain how MOO-RT is trained for passenger prediction and how it is compared with other white-box models. We define the predictor of y as $p : X \rightarrow \hat{y}$ where $X = \{x_i \mid x_i \in \{0,1\}^{|F|}\}$, F is the set of binary features and $\hat{y} \in R$. Since the choice of F determines both the accuracy and explainability of p, we perform feature engineering to add more features to the data.

2.1 Data and Feature Engineering

The prediction methods are trained and evaluated using the 2010 NYC TLC dataset (NYC Open Data). Each row in the dataset corresponds to a taxi request with entries pick-up date and time, pick-up latitude and longitude, passenger count, drop-off date and time, drop-off latitude and longitude. Due to data spatio-temporal granularity, the operational area in New York City is divided into 195 zones using *NYC 2010 Neighborhood Tabulation Areas* dataset (NYC Open Data) and ride requests are aggregated into hourly zone demands. Furthermore, pick-up related information is used to extract basic spatio-temporal features such as [*It is rush hour*] and [*The zone is in west Manhattan*]. However, additional data are also used to capture ride request dynamics. The features that are frequently used in passenger prediction literature (e.g. [13] and [14]) and that are extracted from these data sets are summarized in Table 1.

The data contains both binary and non-binary features. The MOO-RT predictor is a binary regression tree, thus requiring the binarization of numerical and categorical features. To this end, we first convert numerical features into categorical ones by means of creating quartile intervals. Thereafter, we apply binary encoding for all categorical features in the dataset by creating a new column for each category.

2.2 Explainable Multi-Objective Regression Tree Framework

Our approach towards training MOO-RT is adapted from the *STreeD* framework of [15], where an optimal decision tree is found with dynamic programming, leveraging and conditioning *separability* in the training problem. The respective optimization problem is separable if solutions of sub-trees can be calculated independently and can be combined without losing the optimality conditions.

In the following we adopt the formulation presented in [15]. Let $T = \{B, L, b, l\}$ be the regression tree where B is the set of branching nodes, L is the set of leaf nodes, b is the branching function which assigns a binary feature to the node and l is the leaf label assignment function. Given the separable optimization task, the cost (loss) of a tree can be calculated as follows:

Table 1. Features used for training. *Event, population and transportation infrastructure features are extracted via NYC Open Data platform datasets. Weather features are obtained via Visual Crossing Weather API. Demographics features are extracted from NYC Census Bureau datasets.*

Feature Type	Feature
Event	Number of events
Population	Population of the area
Weather	Weather conditions Temperature Wind speed Snow rate Wind gust Wind chill
Demographics: residence and workplace area characteristics (rac and wac)	Total # of jobs Total # of high pay jobs Total # of low pay jobs Total # of jobs with education level 1 (less than high school) Total # of jobs with education level 2 (high school or equivalent) Total # of jobs with education level 3 (college or associate degree) Total # of jobs with education level 4 (bachelor's degree or higher)
Transportation infrastructure	Number of subway stations

$$C(d,u) = \begin{cases} g(d, l(u), u), & \text{if } u \in L \\ C(div(d, \overline{b(u)}), u_L) & \\ \quad \bigoplus \quad C(div(d, b(u)), u_R), & \text{otherwise} \end{cases} \tag{1}$$

In this equation, the cost of the tree is a function of the tree node u and input dataset $d \in X \cup y$ where X is the input feature matrix and y is the passenger demand vector. If the node is not a leaf node, a split and hence two subtrees are created by using a branching function and dividing the dataset based on binary feature satisfaction. The costs of the subtrees are combined with the operator $\bigoplus$. When leaf nodes are reached, the cost of the node is calculated using the assigned label and the leaf node data.

In our multi-objective setting, we need to determine two objectives associated with a leaf node - maximizing accuracy (minimizing prediction loss) and minimizing explainability loss - without violating the separability assumptions of the optimization problem. The prediction loss is calculated using Equation 2.

$$g_{prediction\_loss}(d, l(u)) = \sum_{\{x_i, y_i\} \in d} (y_i - l(u))^2 \tag{2}$$

During training we measure explainability loss through decision depth, i.e. the number of features the leaf dataset goes through for prediction (see Equation 3). If most of the instances are in higher leaf nodes, often predictions are made with a small number of features. Fewer features are likely to be more explainable, since they are less likely to lead to confusion for the model user.

$$g_{decision\_depth}(d, u) = |d|\, depth(u) \tag{3}$$

Figure 3 exemplifies this situation. The figure shows two trees with same topology but different number of instances in the leaf nodes. While for the tree in Figure 3a, total explainability loss is $(10 \times 2) + (20 \times 2) + (10 \times 1) = 70$, the explainability loss of the tree in Figure 3b is $(15 \times 2) + (20 \times 2) + (5 \times 1) = 75$.

Using accuracy and explainability objective functions, a Pareto front can be found by determining the non-dominated set of feasible solutions.

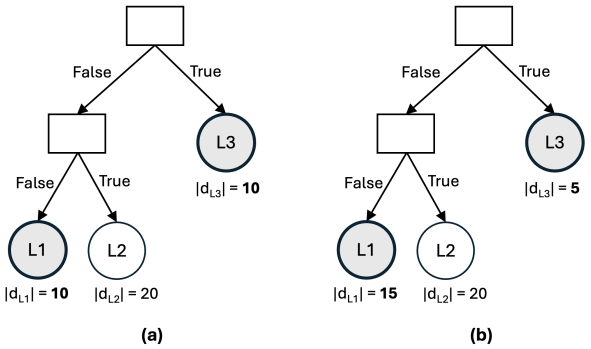

Figure 3. Trees with varying leaf-instance distributions. Highlighted (gray colored) nodes are the modified leaf nodes.

2.3 Comparison of Prediction Models

Given a prediction function p, the Shapley value ([8]) of feature f and instance x is calculated as follows:

$$SV(x,f) = \sum_{S \subseteq F \setminus f} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \left(p(S \cup f, x) - p(S, x)\right) \tag{4}$$

where F is the complete feature set, $S \subseteq F$ is the all subsets of F of any length and $|.|$ is the size operator. In this equation, prediction values directly impact the Shapley values. However, while comparing different models, the value of the prediction should be normalized. Hence, we normalize the Shapley values as follows:

$$NSV(x,f) = \frac{SV(x,f)}{p(F,x)} \tag{5}$$

where $p(F,x)$ is the prediction value of the whole model for a given datapoint x. Given Equation 5, a model's global Shapley value can then be calculated as follows:

$$GSV = \frac{1}{|F||X|} \sum_{x \in X, f \in F} NSV(x,f) \tag{6}$$

3 Results and Discussion

3.1 Data Preprocessing

2010 NYC TLC dataset is used for training and testing. The dataset contains 169,001,162 taxi requests, 3,601,614 of which were removed due to missing information. Discarding areas with sparse requests, we kept 117 zones, which constitute 99.9% of the demand. Thereafter, the cleaned data was aggregated into 117 zones x 365 days x 24 hours buckets, resulting in 1,024,920 instances. After applying binary encoding to the data, we obtained 300 features, of which 191 could be removed due to weak correlations with passenger demand, resulting in 109 binary features. Finally, the resulting data was split into training, validation, and test sets: 10 months of the instances (January 2010 - October 2010) for training, whereas both the validation and test datasets contain 1 month of the instances (November 2010 and December 2010, respectively).

3.2 Multi-Objective Optimal Regression Tree

The first set of experiments considers MOO-RT training. Before solving the dynamic programming formulation of the multi-objective problem, hyperparameters (maximum depth (d) and maximum node number (n)) of the regression tree are tuned while considering prediction loss leading to us choosing values of $d = 4, n = 16$ for the remainder of the experiments. Thereafter, MOO-RT is trained and the optimal Pareto front with 66 trees was found.

The prediction and explainability loss values of each tree in Pareto Front are shown in Figure 4 for both test and training sets. Both loss values are normalised by the respective data set size. From the figure we can conclude that the tree performance for these data sets is comparable. We consider this a confirmation that the training set is representative of the test set. Furthermore, we can observe that the highest prediction loss aligns with the lowest explainability loss. A tree with zero explainability loss, featuring only a root node, leads to the lowest prediction accuracy. Conversely, the tree with the highest explainability loss (full tree with depth 4) yields the lowest prediction loss for both datasets. This suggests that the training set does not overfit, even with the largest tree structure.

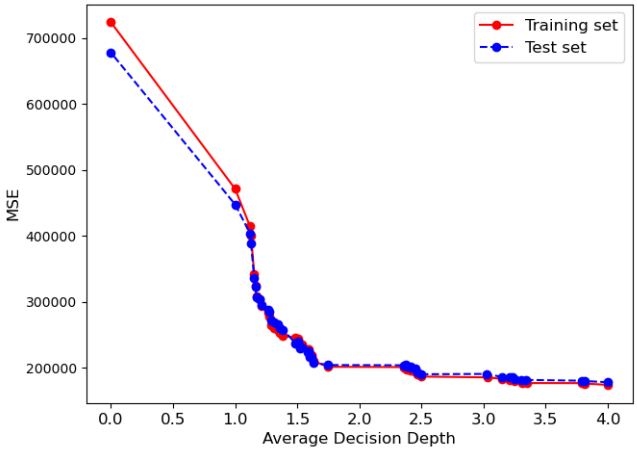

Figure 4. MOO-RT Pareto Front

Examining the Pareto Front, the rapid decrease in prediction loss until an explainability loss of ~1.5 is observed, indicating a critical threshold. After this point, the marginal change in the prediction loss notably decreases. Given that decision depth is the number of features an instance goes through for prediction, the results indicate that most areas in NYC require at least 1 or 2 features for more accurate prediction. Furthermore, the use of additional features does not lead to significant added value for prediction accuracy.

Even though reaching an explainability loss of ~1.5 seems to be critical for obtaining trees with substantially lower prediction loss, trees in the Pareto Front form a non-dominated set, hence no tree is better than another in the joint accuracy / explainability metric. In the following, we select a set of points from the Pareto Front and evaluate them against other methods.

3.3 Comparison of Prediction Models

To evaluate MOO-RT's performance in accuracy and explainability with respect to other explainable prediction models, we select the following benchmark models: Vector Autoregression (VAR), ARIMA, Lasso Regression, and Random Forest. Lasso Regression and Random Forest models are trained through the same binary data used for MOO-RT. However, ARIMA and VAR models require time series. While ARIMA utilizes detrending historical demand, VAR uses the historical demand of neighboring areas, to comply with (non)independence assumptions.

The selected comparison methods' hyperparameters (see Table 2) are tuned using the validation set, and their prediction loss values are determined by calculating the mean squared error (MSE) of the test set predictions. Furthermore, models' explainability is determined by finding the global Shapley values.

Prediction models' global Shapley and MSE values are presented in Table 3. We omitted VAR and ARIMA from the global Shapley value calculation since these models have significantly different feature sets, hampering a direct comparison. Furthermore, from the MOO-RT Pareto Front we select the trees with minimum (MOO-RT-min) or with the quartile prediction loss values. The results show that MOO-RT-min has the highest global Shapley value, indicating the best model explainability, whereas Random Forest obtains the lowest MSE value, at a substantially lower global Shapley Value.

In Table 3, the global Shapley values of MOO-RTs indicate that higher accuracy leads to higher explainability, which contradicts the trees' explainability loss values presented in Figure 4. However, the Shapley value measures the changes in the prediction when a feature is included in the model, if the prediction performance is too poor (such as in MOO-RT-Q3), adding a feature in the model may not significantly affect the prediction value.

In the last part of our experiments, we determine the top-10 features with the highest feature importance values for the MOO-RT-min, Lasso Regression, and Random Forest models. We additionally calculate the average of the feature importance values among these models. Figure 5 summarizes the findings where the average feature importance is indicated as *Mean* and the features in the x-axis are ordered from the highest to the lowest mean feature importance value for the features with top-10 *mean* feature importance values. We find that "*The zone is in West Manhattan*" (or its complement "*The zone*

Table 2. Tuned parameters of comparison methods

Method	Tunned Parameters
Vector Autoregression	Order of the time series
ARIMA	p, d, q
Lasso Regression	Coefficient of feature vector size (alpha)
Random Forest	Tree number, depth, feature number, allow boosting

Table 3. Model Shapley and MSE value results

Model	SHAP	MSE
VAR	-	255257.60
ARIMA	-	310323.81
Lasso Regression	0.97	190681.64
Random Forest	0.57	**145751.53**
MOO-RT-min	**1.94**	202055.57
MOO-RT-Q1	0.99	226668.47
MOO-RT-Q2	0.22	305165.91
MOO-RT-Q3	0.03	548660.94

is in East Manhattan") and "*The zone is in South Manhattan*" (or its complement "*The zone is in North Manhattan*") features are significant in all models. Moreover, "*The zone is Midtown-Midtown South*" and "*It is late night*" features are in all models' top-10 features list. Thus, we can conclude that it is important to include these features to predict passenger demand regardless of the chosen prediction model.

Figure 5. Importance values of top-10 features based on *mean* feature importance

4 Conclusions

In this paper, we propose an explainable passenger demand prediction model for ride-hailing services. For this purpose, we quantify the explainability of a regression tree and adopt the Multi-Objective Optimal Regression Tree (MOO-RT) model, which uses explainability and accuracy objective functions concurrently. The obtained Pareto Front shows that incurring an explainability loss of 1.5 features per instance is critical to obtain significant prediction accuracy.

Furthermore, we compare explainability and accuracy of MOO-RT to various white-box models using mean squared error and Shapley values. The results indicate MOO-RT's ability to enhance explainability compared to other models. However, Random Forest is the top performer in terms of accuracy. Finally, we employ the computed Shapley values to additionally investigate whether model-independent significant features arise from the dataset. Four spatial/temporal features with high Shapley values for all models suggest that these features are key in predicting passenger demand.

For future work, we will focus on assessing explainability using other explainability metrics to consider additional prediction models and to better evaluate and compare models for the passenger prediction task. Further, we aim to evaluate prediction models with the optimization results of the dial-a-ride problem as more accurate predictions may not guarantee better optimization results. This holistic evaluation will provide significant insights into prediction model choice for the dial-a-ride solution mechanism.

Acknowledgements

This research was funded by TU Delft AI Labs initiative through XAIT Lab. In this research, we used Delft Blue supercomputer ([3]) for data cleaning, feature engineering and comparison methods' hyperparameter tuning. Furthermore, we used ChatGPT for text improvement.

References

[1] J. Alonso-Mora, A. Wallar, and D. Rus. Predictive routing for autonomous mobility-on-demand systems with ride-sharing. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3583–3590. IEEE, 2017.

[2] N. Davis, G. Raina, and K. Jagannathan. A multi-level clustering approach for forecasting taxi travel demand. In *2016 IEEE 19th international conference on intelligent transportation systems (ITSC)*, pages 223–228. IEEE, 2016.

[3] Delft High Performance Computing Centre (DHPC). DelftBlue Supercomputer (Phase 1). https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase1, 2022.

[4] X. Li, G. Pan, Z. Wu, G. Qi, S. Li, D. Zhang, W. Zhang, and Z. Wang. Prediction of urban human mobility using large-scale taxi traces and its applications. *Frontiers of Computer Science*, 6:111–121, 2012.

[5] J. Liu, E. Cui, H. Hu, X. Chen, X. Chen, and F. Chen. Short-term forecasting of emerging on-demand ride services. In *2017 4th International Conference on Transportation Information and Safety (ICTIS)*, pages 489–495. IEEE, 2017.

[6] L. Liu, Z. Qiu, G. Li, Q. Wang, W. Ouyang, and L. Lin. Contextualized spatial-temporal network for taxi origin-destination demand prediction. *IEEE Transactions on Intelligent Transportation Systems*, 20(10):3875–3887, 2019.

[7] Z. Liu, H. Chen, Y. Li, and Q. Zhang. Taxi demand prediction based on a combination forecasting model in hotspots. *Journal of Advanced Transportation*, 2020:1–13, 2020.

[8] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[9] I. Markou, F. Rodrigues, and F. C. Pereira. Real-time taxi demand prediction using data from the web. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 1664–1671. IEEE, 2018.

[10] J. Marques-Silva. Disproving xai myths with formal methods – initial results. In *2023 27th International Conference on Engineering of Complex Computer Systems (ICECCS)*, pages 12–21, 2023. doi: 10.1109/ICECCS59891.2023.00012.

[11] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, and L. Damas. Predicting taxi–passenger demand using streaming data. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1393–1402, 2013.

[12] C. Riley, P. Van Hentenryck, and E. Yuan. Real-time dispatching of large-scale ride-sharing systems: integrating optimization, machine learning, and model predictive control. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 4417–4423, 2021.

[13] A. W. Smith, A. L. Kun, and J. Krumm. Predicting taxi pickups in cities: Which data sources should we use? In *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*, pages 380–387, 2017.

[14] Y. Tong, Y. Chen, Z. Zhou, L. Chen, J. Wang, Q. Yang, J. Ye, and W. Lv. The simpler the better: a unified approach to predicting original taxi demands based on large-scale online platforms. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1653–1662, 2017.

[15] J. G. van der Linden, M. de Weerdt, and E. Demirović. Necessary and sufficient conditions for optimal decision trees using dynamic programming. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[16] H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, J. Ye, and Z. Li. Deep multi-view spatial-temporal network for taxi demand prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[17] K. Zhang, Z. Feng, S. Chen, K. Huang, and G. Wang. A framework for passengers demand prediction and recommendation. In *2016 IEEE International Conference on Services Computing (SCC)*, pages 340–347. IEEE, 2016.

[18] X. Zhou, Y. Shen, Y. Zhu, and L. Huang. Predicting multi-step citywide passenger demands using attention-based neural networks. In *Proceedings of the Eleventh ACM international conference on web search and data mining*, pages 736–744, 2018.