Driver Allocation Optimization in Ride-Hailing Platforms: A Study Based on Machine Learning Demand Prediction and Min-Cost Max-Flow Modeling

Yihan Shang
International College of Digital Innovation, Chiang Mai University, Chiang Mai Province 50200, Thailand
SYHsyhzs@outlook.com

ABSTRACT

Ride-hailing platforms have become an important supplement to urban mobility in cities where public transport coverage is incomplete, transfer convenience is limited, or travel demand fluctuates sharply across time and space. Yet the operational advantages of platform-based mobility do not automatically guarantee efficient matching. In practice, passengers may still face long waiting times, distant drivers, and service shortages in hotspot areas, while other areas simultaneously retain idle supply. Focusing on this mismatch, the present study proposes an integrated forecasting-and-optimization framework for driver allocation in the Bangkok taxi context. Because complete platform order data and real-time driver status records are not publicly available, the study adopts a literature-constrained simulation design. The urban study area is divided into 60 spatial zones and organized into 10-minute time slots. On this basis, an XGBoost model is trained to predict zone-level demand for the next time slot, and the prediction output is then transferred to a min-cost max-flow dispatch model that reallocates limited supply across zones under capacity and distance constraints. All modeling and experimental procedures were implemented in Python and executed on Google Colab. The empirical results show that XGBoost outperforms a lag-based baseline on the test set, with a Test MAE of 2.0920 and a Test RMSE of 2.9884. In the case of sufficient supply, the best solution is to handle the distribution within each region without cross-regional scheduling. However, in the case of peak demand and tight supply, the total demand is predicted to reach 720, while the available supply is only 590. At this time, the min-cost max-flow model can achieve 81.94% demand satisfaction rate, keep 83.73% distribution in the original area, control the proportion of cross-regional distribution at 16.27%, and keep the average distribution distance at 0.2051. These results show that the combination of demand forecasting based on machine learning and interpretable network flow optimization can improve the quality of driver allocation when resources are limited, and provide a practical prototype scheme for the operation improvement of Bangkok and similar cities.

Keywords: Ride-Hailing Dispatch; Xgboost; Min-Cost Max-Flow; Bangkok Taxi; Google Colab.

1. INTRODUCTION

Online car platforms such as Grab, Uber, Bolt and Didi have become an increasingly important part of short-distance travel in cities. Compared with the traditional roadside taxi service, these platforms coordinate taxi calling, matching, navigation and payment through digital interfaces, which reduces the transaction trouble between passengers and drivers. These advantages are especially obvious in those cities where the public transportation network can't fully meet the daily travel needs, the last mile connection is still weak, or the demand changes greatly at different times of the day and in different functional areas of the city. Bangkok is such an example. It is a metropolis with obvious spatial differences, complex traffic conditions, mixed travel and commuting needs, and the demand for taxi is obviously concentrated near some commercial and activity centers. In this case, the operational problem is not only whether there are enough cars in the city. The more relevant question is whether the vehicle is located where the demand is about to appear.

Existing studies provide useful but incomplete answers to this problem. One research line concentrates on short-term demand prediction and uses machine learning or deep learning models to infer future requests from time, weather, land-use, and historical-order features [1] [4-7] [13]. Another line focuses on dispatching, rebalancing, empty-vehicle routing, and dynamic ride-sharing under various optimization structures [8-10]. Research on Bangkok also shows that mobility-related GPS data and taxi trajectories can support urban transport analysis [2-3]. However, many studies either only focus on "predicting what will happen" or only study "optimizing under assumed demand". In other words, forecasting and scheduling are often regarded as two independent problems, rather than a continuous step in a decision-making process. This separate treatment weakens the actual value of prediction for platform operation.

This research fills a gap, and it connects the short-term demand forecast at the regional level with an easy-to-understand scheduling optimization model. The research did not pursue the most complicated prediction method, but wanted to establish a clear, explainable and suitable framework for course learning to answer two related questions. First, can you use the variables of time, space and historical demand to accurately predict the demand of a certain area in the next 10 minutes? Secondly, if the future demand has been estimated, but the supply is limited, how can the existing drivers be assigned to different areas in order to improve the service and avoid unnecessary dispatch? In order to answer these questions, a literature-based simulation Bangkok scene is designed, and the demand for the next time period is predicted by XGBoost, and then the prediction results are input into a min-cost max-flow model. The study uses two scenarios to explain the behavior of the model: one is the case of sufficient supply, and the other is the case of peak demand but tight supply. Together, they show when local scheduling is enough and when low-cost cross-regional scheduling is needed.

2. RESEARCH DESIGN AND LITERATURE CASES

We didn’t write a single chapter on literature review, but integrated the previous research into the logic of the whole research design. The research on Bangkok has proved that people’s travel rules and operation structure can be seen from the taxi’s driving trajectory and the probe data of public transportation [2][3]. For example, Phiboonbanakit and Horanont [2] analyzed the data of Bangkok taxis, studied the pricing and profit issues, and showed that these data can not only be used to count the traffic situation, but also help to evaluate the operation situation. Siangsuebchart et al. [3] used the GPS probe data of public transportation to study the travel law of Bangkok Metropolitan Region, and proved that Bangkok’s travel data set can well reflect the changes of time and space. Although these studies have not directly solved the problem of driver assignment here, they show that Bangkok is a very suitable place to study travel problems with data, and it has practical significance.

At the level of forecasting, many studies have found that the demand for online car rental will be affected by short-term time continuity, patterns at different times of the day and spatial differences. Chen, Thakuriah, and Ampountolas [4] used deep learning to predict the short-term demand for car rental on the Internet. They proved that we can predict the short-term demand according to regular historical signals. Jin et al. [5] combined a variety of spatio-temporal information and found that the prediction effect would be better if historical changes and regional conditions were considered together. Qi, Cheng, and Ge [6] believe that the demand for online car rental should be modeled as a multi-factor process, rather than a single sequence problem; Hou, Yang, and Barth [7] show the value of multi-step forecasting structure in grasping the change of demand with time. In addition, Yin et al. [13] pointed out that the built environment variables may have nonlinear and threshold effects on the demand for network car, which further supports the view that “different areas should not be regarded as interchangeable units”.

On the optimization level, the literature emphasizes that once the demand changes in space, matching and rebalancing are the key to system efficiency. Guo, Caros and Zhao [8] studied the rebalancing under uncertain demand, and found that taking the demand-side uncertainty into account can significantly affect the system performance. Braverman et al. [9] analyzed the problem of empty car route in carpooling system, and pointed out the operating cost caused by spatial imbalance. Agatz et al. [10] reviewed the optimization of dynamic carpooling, and proved that route planning and matching decision-making have wider significance in digital coordinated travel service. The main enlightenment of these documents is that we can’t evaluate the travel system only by forecasting quality or static optimization under known demand. When the future demand information is transmitted to the dispatching stage, the operational value will appear.

We learned that method in this study, but applied it to more practical curriculum research. We didn’t try to rebuild the whole database of a proprietary platform, and we didn’t engage in that kind of particularly complicated reinforcement learning system (the kind of system that needs complete online supply status, real travel schedule and a lot of debugging). On the contrary, we use a simpler and clearer method: XGBoost is used to predict the demand of each area in the next period of time, and then min-cost max-flow algorithm is used to assign drivers to different areas. The advantage of this choice is that it is easy to understand and calculate, and it is helpful for practical operation. Therefore, the contribution of this study is not to make the algorithm more complicated, but to show how to combine prediction and optimization, and this method is still easy to understand, repeatable and meaningful in an environment like Bangkok.

3. METHODS AND EXPERIMENTAL DESIGN

3.1 Data Construction and Variable Setting

Because Bangkok's complete real-time network car platform data can't be obtained publicly, this study adopts the simulation design with literature constraints. The simulated city is divided into 60 spatial regions, and each region represents an operation unit of about 1 km ×1 km. In the time dimension, every day is divided into 144 time periods, each of which represents 10 minutes. Then a 30-day "region-time period" observation panel is generated, which creates a structured environment for demand forecasting and scheduling experiments. This design allows the research to focus on the core analysis logic of prediction and redistribution, rather than reverse cracking a proprietary platform database.

This variable system combines time information, historical demand signals and spatial attributes. Time variables include hour, slot identifier, day of week and weekend flag. Historical dependence is represented by lag variables (lag_1, lag_2, lag_6) and rolling averages (rolling_mean_3 and rolling_mean_6). Spatial differences are represented by zone_type and base_demand. The forecast target is target_demand_t_plus_1, which is the demand level in the next period. These variables are deliberately limited to a concise but explainable range, because the main goal is not to build the most complex black-box model, but to provide both predictive and practical input for the subsequent optimization stage. The key variables used in this study are summarized in Table 1.

Table 1. Variable Definitions.

Variable	Definition	Type
zone_id	Identifier of the spatial zone	Categorical
slot_id	Identifier of the 10-minute time slot	Numeric
current_demand	Observed demand in the current time slot	Numeric
predicted_demand	Predicted demand for the next time slot	Numeric
available_supply	Available taxi supply in a zone	Numeric
lag_1	Demand in the previous time slot	Numeric
lag_2	Demand two time slots earlier	Numeric
lag_6	Demand six time slots earlier	Numeric
rolling_mean_3	Mean demand over the most recent 3 time slots	Numeric
rolling_mean_6	Mean demand over the most recent 6 time slots	Numeric
zone_type	Zone category: hotspot, medium, or normal	Categorical
flow_amount	Allocated dispatch flow from one zone to another	Numeric
distance	Dispatch cost proxy between two zones	Numeric

3.2 XGBoost Demand Prediction Experiment

The prediction module uses XGBoost regression [1]. XGBoost is selected for three reasons. First, it captures nonlinear relationships and feature interactions effectively, which is useful when demand depends simultaneously on time, historical demand, and spatial context. Second, it trains efficiently on medium-scale data, which suits the computational scale of this study. Third, it produces interpretable feature-importance outputs, making it better aligned with the explanatory needs of a course paper than more opaque high-complexity architectures.

The dataset is split into training, validation, and test sets in chronological order to avoid information leakage. The input variables include temporal position, historical demand, rolling demand patterns, and zone characteristics. The output variable is next-slot predicted demand at the zone level. Model performance is evaluated using MAE and RMSE, and a lag-1 baseline is included for comparison. All modeling and experimental procedures were implemented in Python and executed on Google Colab [11-12]. The predictive module is not treated as the final outcome of the study. Instead, predicted demand becomes an operational input to the dispatch model. The entire analytical logic is summarized in Figure 1, which shows the progression from simulated data construction to feature engineering, prediction, scenario selection, optimization, and dispatch evaluation.

Before the model is trained, the features are organized to reflect three operational ideas. The first is temporal regularity: demand at 8:00 in the morning should not be interpreted in the same way as demand late at night, and weekday travel should not be interpreted in the same way as weekend travel. The second is short-term continuity: when a zone has been busy in the most recent slots, the next slot is likely to remain relatively busy unless a clear turning point is reached. The third is spatial differentiation: hotspot zones and normal zones do not share the same baseline demand level even under similar time conditions. These ideas are modest compared with large deep learning architectures, but they are conceptually aligned with how ride-hailing demand actually behaves.

Model training and evaluation are carried out in a deliberately conservative manner. The data are not randomly shuffled because the task is predictive in time and the dispatch stage depends on temporal direction. Instead, the sequence is preserved so that the model is always evaluated on later observations than those used for training. In practice, this produces a more realistic test of whether the model could be used operationally. The lag-1 baseline is included not as a strawman but as a useful benchmark, because repeating the most recent observation is often surprisingly competitive in short-term forecasting. If XGBoost can’t perform better than this simple baseline, its practical value is questionable. By directly comparing the proposed model with this simple benchmark, this study can ensure that the improvement of the prediction effect we see is truly meaningful, not just a technical trick.

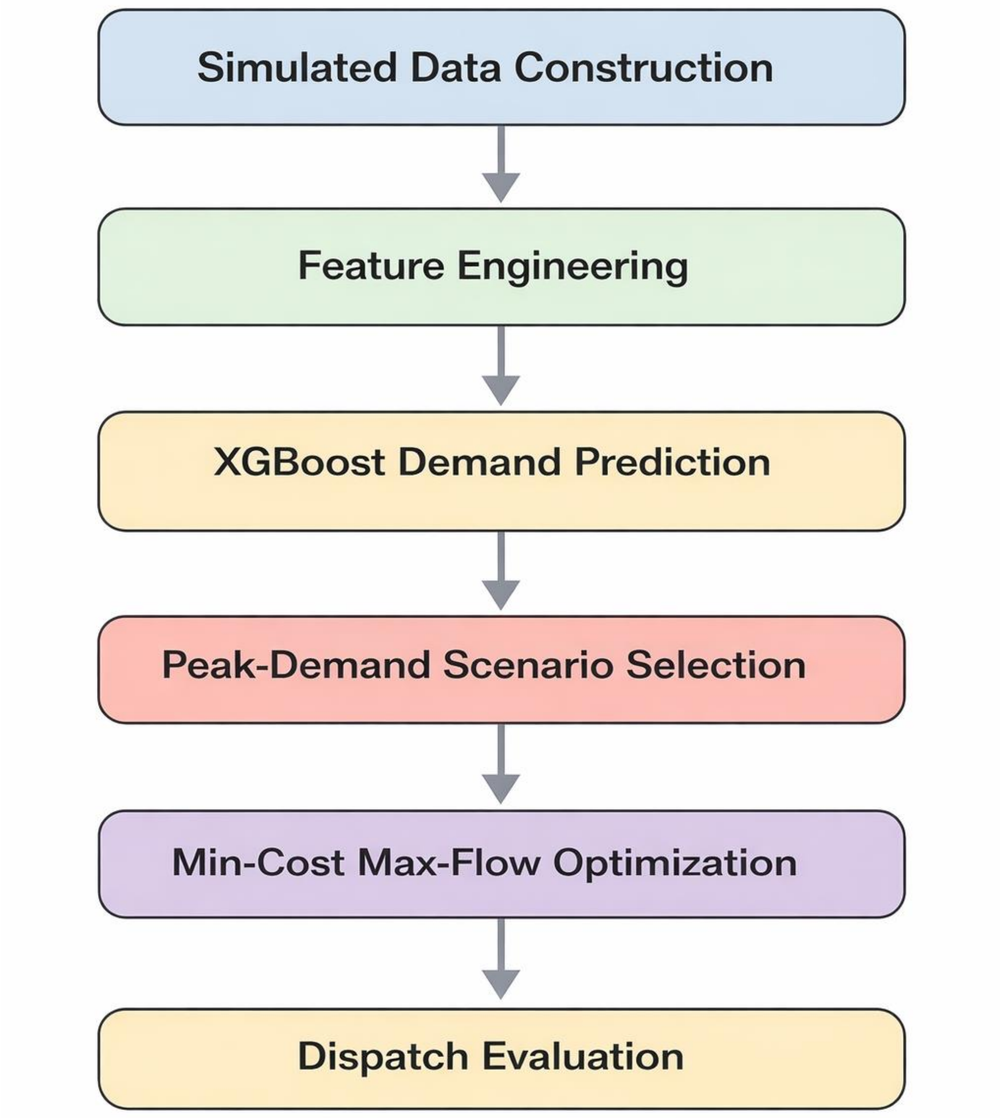

Figure 1. Research Framework.

3.3 Min-Cost Max-Flow Dispatch Experiment

The optimization module is designed as a minimum cost and maximum flow problem. Each time period, each area has a supply status (represented by available drivers) and a demand status (represented by predicted demand). There is a source node, a regional supply node, a regional demand node and a sink node in the network. The capacity from the source node to each supply node is equal to the available supply in this area. The capacity from each demand node to the sink node is equal to the predicted demand. The directed edge from the supply node to the demand node represents the feasible scheduling connection, and the cost of the edge is equal to the distance proxy value between regions. This design can make the model meet the demand as much as possible and minimize the scheduling cost.

We use two scenarios to explain the behavior of the model. Scenario A is the case of sufficient supply, which is used to test whether the optimization algorithm will avoid unnecessary cross-regional movement when the supply is much higher than the demand. Scenario B is a situation of peak demand and tight supply. Its approach is to first find the time period with the highest forecast demand, and then deliberately make the supply in hot areas tight, keep the middle areas in balance, and leave a little surplus in ordinary areas. The second scenario is very important, because it simulates the situation that the total supply is lower than the total demand. At this time, the platform must not only decide where the vehicles are going, but also consider which needs can be really met under the limited resources.

4. RESULTS AND ANALYSIS

4.1 XGBoost Forecasting Results

XGBoost's performance is obviously better than that simple benchmark method based on lag value. On the verification set, the MAE and RMSE of this model are 2.1042 and 3.0451, respectively. On the test set, the Test MAE of the model is 2.0920 and the Test RMSE is 2.9884. The Test MAE and Test RMSE of the lag-1 benchmark method are 2.9141 and 4.2185, respectively. Table 2 summarizes these results. This difference is important, because it shows that our model does not simply copy the previous observation. Instead, it uses rolling historical signals, time structure and spatial variables to improve short-term forecasting.

Figure 2 further shows that the predicted curve can keep up with the overall trend of actual demand in the test sample. Although the model will smooth out some short-term spikes, it can follow the main rise and fall of demand quite stably. This mode is acceptable for the network car operation, because in the dispatching stage, it is more important to identify the direction of high demand period and low demand period reliably, instead of perfectly reproducing every tiny fluctuation.

The results of feature importance also support that our model is reliable. Figure 3 shows that rolling_mean_3 and rolling_mean_6 are the most influential forecasting factors at present, which shows that short-term demand continuity is the most important signal for forecasting the next period. Region-related variables, such as zone_type_normal, zone_type_hotspot and base_demand, also contribute a lot, which shows that the spatial difference is very obvious in a demand pattern like Bangkok. In contrast, hour, current demand and weekend status are auxiliary variables. Together, these results tell us that the demand at the regional level is determined by the recent historical inertia and the stable spatial differences between different regions in the city.

Table 2. XGBoost Prediction Performance.

Model	MAE	RMSE
XGBoost (Validation)	2.1042	3.0451
XGBoost (Test)	2.0920	2.9884
Lag-1 Baseline (Test)	2.9141	4.2185

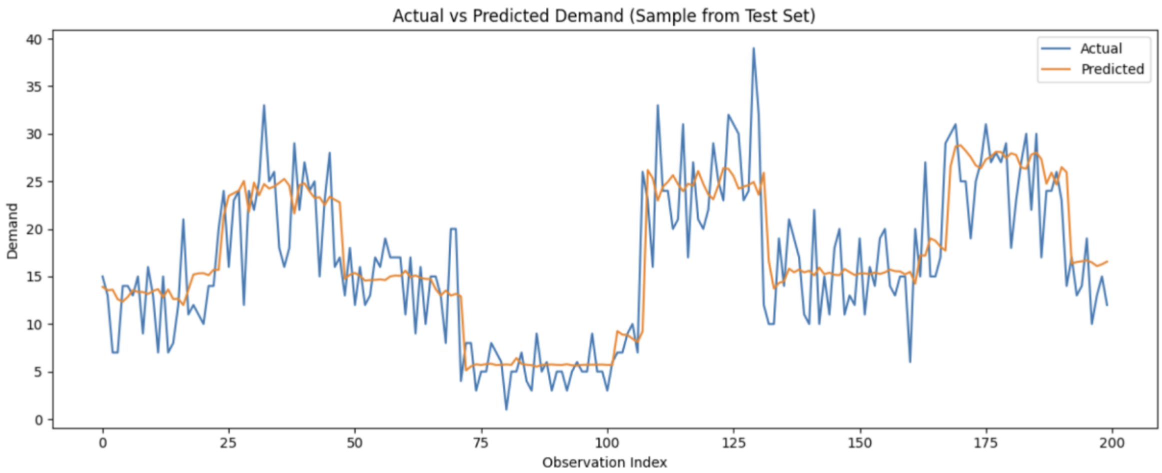

Figure 2. Actual vs Predicted Demand (Test Sample).

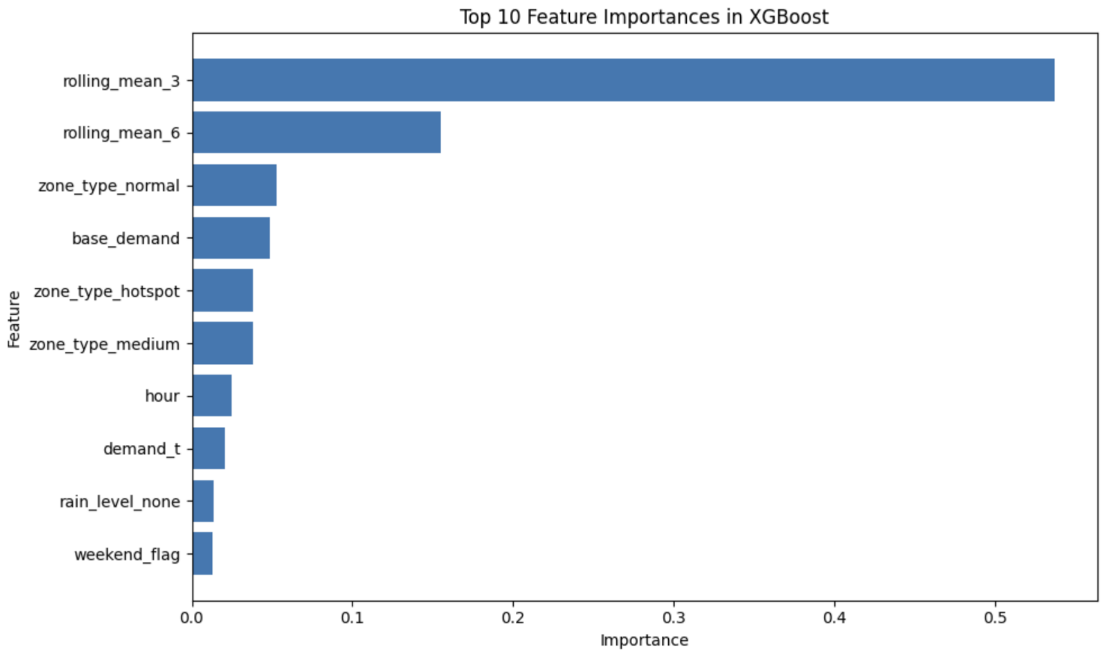

Figure 3. Top 10 Feature Importances in XGBoost.

4.2 Results under the Sufficient-Supply Scenario

Scenario A shows that our supply is sufficient to meet our predicted demand. Under this condition, the optimization result is simple: the total flow is 335, the same regional flow is 335, the cross-regional flow is 0, the same regional ratio is 1.0000, and the average scheduling distance is 0. Table 3 gives a statistical summary of this situation. This result is important, not because it looks great, but because it proves that our optimization model will not move for the sake of moving. When each area can meet the predicted demand with its own local supply, the most economical solution is to leave the vehicles in the original area.

This result can also be used to understand more stringent situations. It shows that the model will act conservatively under loose conditions and prefer zero-cost local satisfaction. In practice, this means that this framework will not suggest unnecessary rescheduling when the platform supply is sufficient. The stability of this behavior enhances the credibility of the optimal design.

From a management point of view, the difference between adequate supply and tight supply is crucial. If a platform is over-scheduled when the supply is sufficient, it will waste the driver's time and increase idle driving; If it is not scheduled when the supply is tight, it will make the areas that need support unable to meet the actual demand. The current framework can clearly and operationally distinguish these two situations.

Table 3. Scenario A Summary.

Metric	Value
Total Flow	335
Same-Zone Flow	335
Cross-Zone Flow	0
Same-Zone Ratio	1.0000
Cross-Zone Ratio	0.0000
Average Dispatch Distance	0.0000

4.3 Results under the Peak-Demand Tight-Supply Scenario

Scenario B is more valuable in analysis because it reflects a situation of high demand and low supply. In this scenario, we choose the time period with the highest forecast demand during the test period as the experimental case. The total forecast demand has reached 720, while the total available supply is only 590 under the low supply conditions we set. Therefore, the model called min-cost max-flow can't meet all the requirements. Its optimal scheme gives a maximum scheduling flow of 590, so that the demand satisfaction rate is 0.8194 and the total scheduling cost is 121. Table 4 summarizes these results.

The structure in the scheduling result is also very important. Of the 590 distributed traffic, 494 were completed in the original zone, and only 96 need to be dispatched across zones. That is to say, 83.73% of all our service needs are still handled locally, and only 16.27% need to be redistributed across zone. The average scheduling distance is only 0.2051, which shows that even if there is a cross-zone movement, the range is very small and basically local. Table 5 reports the most important scheduling flows in this case. The maximum traffic is basically the distribution in the same zone, which shows that our model will give priority to local completion whenever possible. Cross-zone dispatch is used as a supplementary response to local shortages rather than as a default strategy.

This distinction matters for operational interpretation. The model does not suggest that more relocation is always better. Rather, it shows that when supply is insufficient in aggregate, the platform should first preserve as much local fulfillment as possible and then use limited, low-cost neighboring dispatch to alleviate pressure in selected zones. Compared with a pure max-flow formulation, which may satisfy demand with excessive cross-zone movement, the min-cost max-flow structure produces a more plausible operational pattern: high utilization of scarce supply, restrained relocation, and low average dispatch distance. In the context of Bangkok, this means that a forecasting-informed dispatch system can help the platform respond to demand concentration without imposing widespread and costly vehicle movement across the city.

The high-demand scenario therefore serves as the real test of the framework. Because total available supply is below total predicted demand, the system faces a resource allocation problem rather than a simple matching problem. Under these conditions, the optimization result becomes interpretable on three levels. First, the satisfaction rate of 81.94% reflects the hard resource ceiling imposed by limited supply. Second, the positive but limited dispatch cost indicates that the model does use spatial reallocation, but only to the extent needed to improve service. Third, the dominance of same-zone flow shows that the model does not sacrifice local stability for aggressive rebalancing. This combination is precisely what one would hope to see in a realistic ride-hailing dispatch strategy under scarcity.

The dispatch structure reported in Table 5 deserves particular attention. Even among the top dispatch flows, many of the largest allocations remain same-zone flows. This means that the model still values immediate local matching whenever the local zone possesses feasible supply. Cross-zone flows emerge not because the model prefers movement, but because some zones cannot internally cover their predicted demand while nearby zones retain mobilizable supply. In practical terms, this suggests that a forecasting-informed platform should not treat the city as a homogeneous pool of interchangeable vehicles. Instead, it should preserve local matching as the default and activate nearby repositioning only when zone-level forecasts indicate an upcoming imbalance.

The second influence is related to the judgment of managers. If the platform only adjusts according to the orders currently seen, it may underestimate the areas where demand is about to rise and overestimate the role of other idle resources. In contrast, our current framework is to use the predicted demand for the next period to drive resource allocation. This means that optimization is not only to solve the shortage, but to respond to the places that may aggravate the shortage in advance.

This difference is subtle but important. The value of this framework is not only to mobilize drivers, but also to mobilize according to the prediction of the coming pressure, rather than relying solely on the current static situation.

Finally, the comparison between Scenario A and Scenario B shows what is the use of combining forecasting and optimization. The same model structure, because of the different resource conditions of the system, will produce two very different but easy-to-understand results. When resources are sufficient, the best way is to do it locally at zero cost. When resources are in short supply, the best way is to do it locally, and at the same time, with a small amount, transfer some from the side. This shows that this framework is flexible, not rigid. It will not forcibly use the same allocation method for every situation of the system, but will adjust the intensity of transferring resources according to the predicted imbalance.

Table 4. Scenario B Summary.

Metric	Value
Total Predicted Demand	720
Total Available Supply	590
Min-Cost Max-Flow Value	590
Demand Satisfaction Rate	0.8194
Total Dispatch Cost	121
Total Dispatched Flow	590
Same-Zone Flow	494
Cross-Zone Flow	96
Same-Zone Ratio	0.8373
Cross-Zone Ratio	0.1627
Average Dispatch Distance	0.2051

Table 5. Top Dispatch Flows under the Peak-Demand Tight-Supply Scenario.

From Zone	To Zone	Flow Amount	Distance
Z03	Z03	20	0
Z09	Z09	19	0
Z06	Z06	19	0
Z01	Z01	18	0
Z04	Z04	18	0
Z07	Z07	18	0
Z08	Z08	18	0
Z15	Z15	15	0
Z25	Z25	15	0
Z20	Z20	14	0
Z10	Z10	14	0
Z14	Z14	14	0
Z17	Z17	14	0
Z02	Z02	14	0
Z11	Z11	12	0
Z24	Z24	12	0
Z26	Z26	12	0
Z19	Z19	11	0
Z05	Z05	11	0
Z21	Z21	10	0

5. CONCLUSION

This study develops and tests an integrated forecasting-and-dispatch framework for driver allocation in a Bangkok-oriented ride-hailing setting. Using a literature-constrained simulation design, it first constructs zone-level demand data, then applies XGBoost to next-slot demand prediction, and finally feeds the predicted demand into a min-cost max-flow dispatch model. The empirical results show that the forecasting module performs better than a lag-based benchmark and that its most influential predictors are short-term rolling demand signals and zone-level spatial attributes. The optimization results further show that dispatch behavior depends strongly on the resource condition of the system.

In the sufficient-supply scenario, the optimal solution is entirely local: each zone fulfills its own demand and no cross-zone dispatch is necessary. In the peak-demand tight-supply scenario, however, total supply falls below total predicted demand, and the platform can satisfy only 81.94% of predicted requests. Even under that constraint, the min-cost max-flow model still keeps 83.73% of fulfilled demand within the original zone and limits cross-zone dispatch to a relatively small share with a low average dispatch distance. These findings indicate that a forecasting-informed optimization framework can improve allocation rationality under scarcity without encouraging excessive relocation.

The broader methodological implication is that model choice should match data conditions. In the absence of proprietary platform databases, complete online driver states, and real travel-time matrices, it is more useful to combine an interpretable predictive model with a transparent optimization structure than to adopt an overly complex black-box framework that cannot be properly supported by available data. The combination of XGBoost and min-cost max-flow therefore provides not only a workable course-research prototype, but also a practical and explainable template for future operational analysis in Bangkok and similar cities.

REFERENCES

[1] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785-794).
[2] Phiboonbanakit, T., & Horanont, T. (2021). Analyzing Bangkok city taxi ride: Reforming fares for profit sustainability using big data driven model. Journal of Big Data, 8, Article 7.
[3] Siangsuebchart, P., Khamkong, M., Satiennam, T., Satiennam, W., Jaensirisak, S., & Wiriyasermkul, P. (2021). Public transport GPS probe and rail gate data for assessing the pattern of human mobility in the Bangkok Metropolitan Region, Thailand. Sustainability, 13(4), Article 2178.
[4] Chen, L., Thakuriah, P. V., & Ampountolas, K. (2021). Short-term prediction of demand for ride-hailing services: A deep learning approach. Journal of Big Data Analytics in Transportation, 3(2), 175-195.
[5] Jin, G., Cui, Y., Zeng, L., Tang, H., Feng, Y., & Huang, J. (2020). Urban ride-hailing demand prediction with multiple spatio-temporal information fusion network. Transportation Research Part C: Emerging Technologies, 117, Article 102665.
[6] Qi, Q., Cheng, R., & Ge, H. (2022). Short-term travel demand prediction of online ride-hailing based on multi-factor GRU model. Sustainability, 14(7), Article 4083.
[7] Hou, Y., Yang, C., & Barth, M. (2019). Data-driven multi-step demand prediction for ride-hailing services using convolutional neural network. In Computer Vision Conference (pp. 11-22).
[8] Guo, X., Caros, N. S., & Zhao, J. (2021). Robust matching-integrated vehicle rebalancing in ride-hailing system with uncertain demand. Transportation Research Part B: Methodological, 150, 161-189.
[9] Braverman, A., Dai, J. G., Liu, X., & Ying, L. (2019). Empty-car routing in ridesharing systems. Operations Research, 67(5), 1437-1452.
[10] Agatz, N., Erera, A., Savelsbergh, M., & Wang, X. (2012). Optimization for dynamic ride-sharing: A review. European Journal of Operational Research, 223(2), 295-303.
[11] Google Research. (n.d.). About Colaboratory. Retrieved from https://research.google.com/colaboratory/faq.html
[12] Google for Developers. (n.d.). Meet the new Colab. Retrieved from https://developers.google.com/colab
[13] Yin, J., Zhao, F., Tang, W., & Ma, J. (2024). The nonlinear and threshold effect of built environment on ride-hailing travel demand. Applied Sciences, 14(10), Article 4072.