# Dual-Horizon Parking Availability Prediction with Multi-Factor Recommendation: An Empirical Study on Real-World Data

**Yanjie Huang**

**Abstract** To address the two key problems in urban parking guidance—missing real-time information and inaccurate short-term prediction—this paper presents an empirical study of dual-horizon availability prediction and multi-factor recommendation on the public Birmingham car park dataset. For 30-minute and 60-minute prediction horizons, we construct a 39-dimensional feature set comprising 38 time-series features in three groups—temporal, lag, and sliding-window statistics—plus one static capacity feature, and systematically compare LightGBM, XGBoost, LSTM, and GRU models. Results show that, with strong feature engineering, gradient-boosted trees significantly outperform recurrent neural networks (MAPE 5.53% versus 9.38% at the 30-minute horizon) and degrade the slowest as the horizon lengthens. Ablation experiments further reveal a horizon effect: short-horizon prediction relies most on sliding-window statistics, whereas at the longer horizon periodic temporal encodings become the dominant signal while lag features contribute almost nothing. Building on these predictions, we propose a prediction-driven multi-factor scoring recommendation strategy and evaluate it in a stylized simulation with data latency. The strategy achieves one-shot success rates of 98.8%, 98.4%, and 97.3% under normal, high-load, and "high-load + latency + strict threshold" scenarios respectively—on par with ideal real-time recommendation, significantly better than distance-only ranking, and without requiring a real-time data stream. Finally, we present a complete engineering implementation covering prediction models, recommendation decisions, and a user feedback loop.

**Keywords** parking guidance; availability prediction; gradient boosting decision tree; dual-horizon prediction; multi-factor recommendation; feedback loop

## Introduction

With the continuous growth of urban motor vehicle ownership, "parking difficulties" have become a prominent problem restricting urban traffic efficiency. Besides the insufficient total supply of parking spaces, the asymmetry of supply and demand information is a significant source of the problem: drivers repeatedly cruise around their destinations looking for empty spaces, a phenomenon known as "cruising for parking," which not only prolongs individual travel time but also exacerbates local road network congestion and generates additional emissions.

Effective parking guidance relies on two technological supports: first, short-term availability prediction. There is often a time lag of tens of minutes between a user's parking decision and their actual arrival at the parking lot. The availability of spaces at that moment does not represent the availability upon arrival; therefore, it is necessary to predict the future availability of parking spaces. Second, recommendation decision-making. Among multiple candidate parking lots, a ranking needs to be determined by comprehensively considering multiple factors such as future availability, walking distance, and time cost. This paper conducts an empirical study based on the Birmingham parking lot public dataset, with the following main contributions:

(1) On the same real public dataset, the performance of gradient boosting trees and recurrent neural networks in predicting parking spaces in a 30/60-minute dual-time window is compared. The contribution of three sets of feature engineering and the time window effect are quantified through ablation experiments, providing empirical evidence for model selection.

(2) A prediction-driven multi-factor rating recommendation strategy is proposed. Under realistic simulation conditions with data latency, it is verified that it can achieve a success rate comparable to ideal real-time recommendation without relying on real-time data streams, and its stability is superior to the pure distance strategy.

(3) A complete engineering system implementation from the prediction model to the recommendation decision and then to the user feedback closed loop is provided, offering

ISSN 3106-7018 (Print) ISSN 3106-5791 (Online)

a reference for the implementation of similar systems.

# 2 Systems and methods

## 2.1 Overall system architecture

This article implements a complete parking guidance system, using a bottom-up four-layer architecture:

- Data layer: Responsible for the collection, cleaning and storage of parking lot static information (location, capacity) and occupancy observation data. This layer completes missing value processing, abnormal cropping and feature calculation, provides a unified feature view for the upper layer, and maintains derivative data such as user feedback;
- Model layer: Train offline and deploy two prediction models online for 30 minutes and 60 minutes, respectively outputting vacancy predictions for the two future time slots. The design of independent deployment of dual models allows the two time windows to adopt differentiated feature configurations and update cycles, which also facilitates separate iterations;
- Decision-making layer: Based on the prediction results, observed vacancy rate and spatial distance, run a multi-factor scoring algorithm to generate candidate parking lot rankings, and encapsulate strategic logic such as retry and downgrade, such as falling back to the observed vacancy rate ranking when the predicted service is abnormal;
- Application layer: Provides destination query, parking lot recommendation, route navigation and arrival feedback entrance for end users, and flows feedback data back to the data layer to form a closed loop.

### 2.2 Dual-Window Parking Space Prediction

Problem Formal Definition: Given the observation sequence $\{x_{(i,1)},x_{(i,2)},\ldots,x_{(i,t)}\}$ of parking lot i, where $x_{(i,t)}$ represents the number of available parking spaces at time t (sampling granularity is 30 minutes), the prediction task is: based on all observable information up to time t (including static attributes, time attributes, and historical observations), predict the number of parking spaces $\hat{x}_{(i,t+1)}$ and $\hat{x}_{(i,t+2)}$ at time t+1 (30 minutes later) and time t+2 (60 minutes later). Formally, the learning mapping $f_h:X_t\to\hat{x}_{(i,t+h)}$, $h\in\{1,2\}$, and the training objective is to minimize the regression error between the predicted and actual values. The two time windows correspond to two typical use cases: "near arrival" and "mid-journey," respectively. This paper trains independent models for each, forming a dual-time-window prediction architecture.

Feature engineering. For each parking lot-time slot sample, 39-dimensional features are constructed, including 38 time-series features across three groups: time, lag, and sliding window statistics, plus a 1-dimensional static capacity feature (parking lot capacity). The three groups of time-series features are semantically divided as follows:

Time features (9 dimensions): Time slot index, hour, minute, sin/cos periodic encoding of slot position, day of the week, whether it is a weekend, and sin/cos periodic encoding of weekday. Parking occupancy exhibits significant intraday cycles (morning rise, midday peak, evening fall) and intraweekly cycles (significant differences between weekday and weekend patterns);

Llag features (9 dimensions): Number of vacant spaces in the previous 1–6 steps, number of vacant spaces in the same slot the previous day, number of vacant spaces in the same slot the previous week, and current occupancy rate. This set of features captures recent status and periodic continuations across days/weeks: the first 1–6 steps cover the immediate dynamics within the last 3 hours, while the previous day and week's troughs provide direct references for daily and weekly cycles;

Sliding window statistical features (20 dimensions): mean, maximum, minimum, and standard deviation for window lengths 3/6/12 (12 dimensions in total), mean of the first-order difference trend for windows 3/6 (2 dimensions), and sliding window mean and standard deviation of occupancy rate (6 dimensions). This set of features characterizes the statistical pattern and trend of recent occupancy levels, enabling the model to distinguish between "high-level stability" and "rapid rise," states that cannot be differentiated by single-point observation.

Dual-model architecture. In terms of model selection, this paper compares two paradigms: gradient boosting trees represented by LightGBM[13] and XGBoost[16], which directly use 39-dimensional feature vectors as input and capture the nonlinear interaction between table features through an additive model that fits the residuals tree by tree. They have the advantages of efficient training, robustness to hyperparameters, and output of feature importance; and recurrent neural networks represented by LSTM[14] and GRU[17], which use a 7-dimensional time window with a length of 18 steps (exactly one full business day) as input. The 7 dimensions include occupancy rate, slot sin/cos encoding, weekday sin/cos encoding, and whether it is the weekend and the same slot occupancy rate as the previous day. The dynamic dependence of the occupancy sequence is learned through a gating mechanism, and the training objective is normalized by capacity to eliminate the influence of differences in parking lot size. This design makes the comparison of the two models reflect the applicability of the two technical routes of "feature engineering driven" and "end-to-end sequence modeling" on this task. The conclusions can provide a reference for the model selection of similar city-level parking prediction systems.

### 2.3 Multi-factor scoring recommendation

The prediction results need to be transformed into actionable suggestions for users through recommendation decision-making. For user queries, the system first determines a set of candidate parking lots within walking distance of the user's destination, and then calculates a comprehensive score for each candidate parking lot j:

Where: The predicted vacancy rate, given by the prediction model in Section 3.2, reflects the expected availability at the user's arrival time and is the core basis for decision-making—compared to "whether there are still spaces available upon arrival," "whether there are still spaces available upon arrival" is the user's real concern; the observed vacancy rate is the current observation value, used as supplementary information when data is available to correct possible prediction biases; the distance item is the walking distance (in meters) from the user's destination to the parking lot, with a greater penalty for longer distances; the time cost item characterizes the combined time cost of travel and parking search. The four weights w_1, w_2, w_3, and w_4 reflect decision-making preferences and can be adjusted based on city data conditions and user preferences. The actual configuration used in the simulation experiment of this paper is w_1=0.55, w_2=0.15, w_3=0.0003; w_4=0 is used in the simulation of this paper. The time cost is reflected by the arrival judgment and retry penalty mechanism (see Section 4.4), that is:

This configuration reflects the strategy orientation of "prediction as the main factor, observation as the auxiliary factor, and moderate distance penalty": the prediction item has the highest weight, ensuring that the recommendation is based primarily on the availability at the arrival time; the observation item has a lower weight, only playing a corrective role when data is available; the distance item coefficient is small, so that distance only plays a role in breaking the tie when availability is similar, avoiding the distance factor overpowering the availability factor and degenerating into a "nearest distance" strategy. Candidate parking lots are ranked in descending order of their scores, with the top-ranked one being the preferred recommendation. If the preferred recommendation fails, users retry in order of ranking. Another advantage of this design is its robustness: when real-time observations are missing or delayed, simply setting w_2 to zero (or replacing it with a predicted value) allows the strategy to smoothly degrade to pure prediction-driven without altering the system flow.

### 2.4 Feedback Closed-Loop Mechanism

To enable continuous optimization of the system, this paper designs a user feedback closed loop: after completing parking, users submit arrival feedback ("parked/did not parked") through the application layer. The backend associates this feedback with recommendation records, decision context (predicted value, rating, candidate list, scene parameters) and writes it into the feedback table. Each feedback effectively provides a real-world result annotation for a recommendation decision, an evaluation signal that cannot be obtained in offline experiments. The accumulated feedback data has three uses: first, as incremental training samples for the prediction model, used for periodic model retraining, enabling the model to adapt to seasonal and trend-based shifts in occupancy patterns; second, as an evaluation basis for recommendation strategies, such as statistically analyzing the actual success rate of preferred recommendations in different scenarios and verifying the rationality of the "parkable" judgment threshold; third, providing data support for online tuning of the rating weights w_1–w_4. The closed-loop mechanism enables the system to evolve from "one-time modeling" to "data-driven continuous iteration," and also provides a data foundation for subsequent verification of prediction and recommendation strategies on real operational data.

## 3 Experiment and Results Analysis

### 3.1 Experimental Setup

Dataset. The experiment used the Birmingham parking lot public dataset (UCI Machine Learning Repository, Dataset #482), compiled and released by Stolfi et al. [5]. This dataset consists of parking lot occupancy records operated by the Birmingham City Council in the UK, and is one of the most widely used public benchmarks in the field of parking prediction. It has been used in many studies for model comparison [7,15]. The data time range is from October 4, 2016 to December 19, 2016, with a sampling granularity of 30 minutes and operating hours of 08:00–16:30 daily, totaling 18 time slots. The original data contains 30 parking lots, and after data quality screening (capacity not less than 200 and complete days not less than 40), 28 parking lots are retained, with capacities ranging from 220 to 4675 parking spaces, covering a variety of sizes from small and medium-sized parking lots to large commercial complex parking lots.

Preprocessing. Missing time slots were filled using intraday linear interpolation, and only days with at least 12 valid observations were retained to avoid over-interpolation of long periods of missing data. The number of gaps was obtained by subtracting the number of occupancy from the capacity and was cropped to the [0, "capacity"] interval to eliminate out-of-bounds values caused by observation noise. Parking lot occupancy exhibits a significant diurnal cyclical pattern, as shown in Figure 1. The weekly occupancy rate curve of the Bull Ring parking lot clearly demonstrates the intraday pattern of "early rise - midday high - evening decline," which is the data basis for the effective application of time characteristics and periodic lag characteristics.

ISSN 3106-7018 (Print) ISSN 3106-5791 (Online)

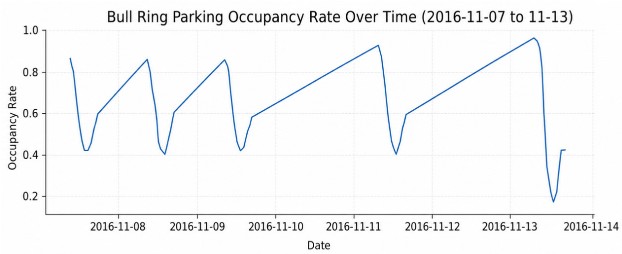


**Figure 1. Weekly Occupancy Time Series for Bull Ring Car Park**

Data Splitting. A strict chronological split was employed to prevent data leakage: samples prior to December 5, 2016, formed the training set (20,111 samples), while subsequent samples formed the test set (5,814 samples).

Models and Hyperparameters. Gradient Boosting Trees: LightGBM (`num_leaves=63`, `n_estimators=500`, learning rate 0.05, `subsample` and `colsample` both 0.9) and XGBoost (`n_estimators=500`, `max_depth=6`, learning rate 0.05). Recurrent Neural Networks: Both LSTM and GRU models featured 2 layers and a hidden dimension of 64, with an input window of 18 steps by 7 dimensions; the Adam optimizer was used (learning rate 5e-4) with a gradient clipping threshold of 1.0; training lasted 20 epochs with a batch size of 512, and training targets were normalized by capacity.

Baselines and Evaluation Metrics. Two naive baselines were established: a persistence baseline (extrapolating the current value directly—i.e., assuming future vacancy counts equal current observations) and a historical mean baseline (using the value from the same slot on the previous day), serving to assess the performance gains of the learning models over simple heuristics. Evaluation metrics included RMSE, MAE (both in units of parking spaces), and MAPE. RMSE is sensitive to large errors, reflecting prediction reliability in extreme scenarios; MAE reflects the mean absolute deviation; and MAPE provides normalized comparability across parking lots. To avoid distortion caused by small denominators in near-full or near-empty states, samples where the actual vacancy count was within 5% of capacity were excluded from the MAPE calculation. All metrics were computed on the test set, following the strict chronological split between training and testing ata.

### 3.2 Main experimental results

**Table 1. 30-minute (t+1) prediction results**

| Model | RMSE | MAE | MAPE |
|---|---|---|---|
| LightGBM | 36.18 | 20.51 | 5.53% |
| XGBoost | 35.92 | 20.43 | 5.56% |
| GRU | 57.16 | 31.73 | 7.97% |
| LSTM | 65.02 | 39.00 | 9.38% |
| Persistent baseline (extrapolation of current value) | 165.23 | 90.91 | 23.33% |
| Historical average baseline (same slot, previous day) | 235.75 | 129.34 | 41.02% |

**Table 2. 60-minute (t+2) prediction results**

| Model | RMSE | MAE | MAPE |
|---|---|---|---|
| LightGBM | 54.11 | 31.57 | 8.71% |
| XGBoost | 54.79 | 31.91 | 8.86% |
| GRU | 95.03 | 52.67 | 13.24% |
| LSTM | 125.50 | 68.33 | 17.10% |
| Sustained baseline | 310.42 | 174.95 | 47.15% |
| Historical average baseline | 273.17 | 155.71 | 49.31% |

As shown in Tables 1 and 2, the main experiment yields three key findings. First, with robust feature engineering, gradient boosting trees significantly outperform recurrent neural networks: for a 30-minute time window, the MAPE values for LightGBM and XGBoost are 5.53% and 5.56% respectively—approximately 2–4 percentage points lower than those of GRU (7.97%) and LSTM (9.38%)—with a consistent ranking of superiority observed across RMSE and MAE metrics as well. This conclusion aligns with the findings of Lucchese et al. [15] on the same dataset, demonstrating that for this type of parking prediction task—characterized by tabular data and limited daily operating hours—carefully constructed features combined with gradient boosting trees constitute a more effective approach than end-to-end sequence modeling. The underlying reason is that the dataset contains only 18 time slots per day, resulting in short sequences; consequently, recurrent neural networks struggle to learn patterns superior to manually engineered features from such limited data and instead face a higher risk of overfitting. In contrast, gradient boosting trees can directly leverage non-linear interactions among the 39 features, offering a distinct advantage for small-to-medium-sized tabular datasets. Second, extending the time window leads to a general decline in accuracy, with gradient boosting trees exhibiting the slowest rate of degradation: as the prediction window expands from 30 to 60 minutes, the MAPE for all models rises by approximately 3–8 percentage points; specifically, LightGBM's MAPE increases only from 5.53% to 8.71% (+3.18 pp), whereas LSTM's rises from 9.38% to 17.10% (+7.72 pp), representing a significantly larger drop in performance. This indicates that the utilization of cyclical temporal features and sliding-window statistics by tree-based models remains effective even with longer time windows, whereas the reliance of sequence models on recent observations causes their information sources to deplete more rapidly under such conditions. Third, simple baselines are entirely inadequate: the MAPE values for both the persistence baseline and the historical mean baseline exceed 23% (approaching or surpassing 47% for the

60-minute time window). This demonstrates that the task necessitates a learning-based model and confirms the essential role of the prediction module within the guidance system; if the system were to revert to simple heuristics, the availability estimates provided for recommendations would lose their practical value. Figure 2 compares the LightGBM dual-time-window prediction curves with actual values during the test period; the curves closely track intraday occupancy trends, with the 30-minute window showing a better fit than the 60-minute window—consistent with the performance rankings in Tables 1 and 2.

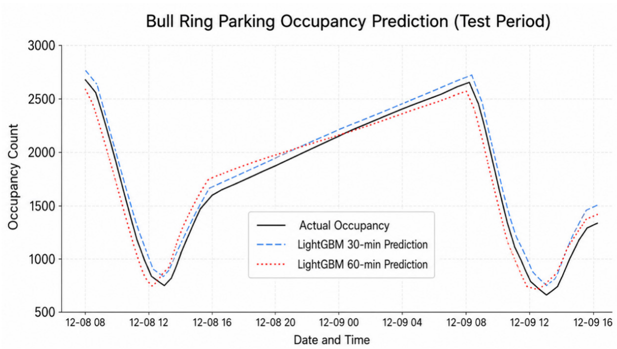


**Figure 2. Comparison of LightGBM dual-time-window predictions and actual values during the test period.**

### 2.3 Ablation Study

To quantify the contribution of each feature group and examine the effect of the time window, this study conducted ablation experiments using LightGBM as the base model; the models were retrained and evaluated after removing the three feature groups one by one, and the results are presented in Table 3.

**Table 3. LightGBM feature ablation experiment results (MAPE)**

| Configuration | 30min | 60min |
|---|---|---|
| Full set (39 dimensions) | 5.53% | 8.71% |
| Remove temporal features (9 dimensions) | 5.93% | 9.86% |
| Remove lag features (9 dimensions) | 5.59% | 8.72% |
| Remove sliding window features (20 dimensions) | 6.56% | 9.65% |

The ablation results reveal a significant time-window effect: for the short 30-minute window, sliding-window statistical features contribute the most—their removal causes the MAPE to rise by 1.03 percentage points (from 5.53% to 6.56%)—indicating that short-term prediction relies primarily on the statistical patterns of recent occupancy levels. Conversely, for the long 60-minute window, temporal features emerge as the most critical signal; removing them increases the MAPE by 1.15 percentage points (from 8.71% to 9.86%), whereas the contribution of lag features becomes negligible (an increase of only 0.01 percentage points). The underlying mechanism is that as the prediction window extends, the correlation between recent observations and the target time decays rapidly, significantly diminishing the predictive value of "current vacancy counts" for "vacancy counts one hour later." In contrast, the structural patterns embedded in cyclical temporal encodings—such as midday peaks and afternoon declines—remain valid regardless of the window length, serving as the primary basis for distinguishing occupancy levels. Sliding-window statistics occupy a middle ground: they incorporate recent information while partially mitigating information decay through statistical smoothing, thereby maintaining significant contributions across both time windows (with performance degrading by 1.03 and 0.94 percentage points, respectively, upon their removal). This finding offers direct guidance for feature engineering: prediction models targeting different time windows should employ differentiated feature configurations; specifically, long-window models should emphasize cyclical temporal encodings and higher-order statistical features rather than simply adopting configurations designed for short windows. Figure 3 illustrates the top 15 features by importance for the LightGBM model, showing the dominance of sliding-window statistical features, which corroborates the conclusions drawn from the ablation study.

## 4 Conclusions and Outlook

This paper addresses the issue of urban parking guidance by conducting an empirical study on the Birmingham public dataset, focusing on dual-time-window vacancy prediction and multi-factor scoring-based recommendation. Key conclusions include: (1) Supported by 39-dimensional feature engineering, gradient boosting trees (LightGBM/XGBoost) significantly outperformed LSTM/GRU across both 30-minute and 60-minute time windows and exhibited the slowest performance degradation as the time window lengthened, achieving a MAPE of 5.53% for the 30-minute window; (2) Ablation experiments revealed a "time-window effect": short-window predictions rely primarily on statistical features from sliding windows, whereas for long windows, periodic time encoding becomes the dominant signal while the contribution of lag features approaches zero; (3) Under simulated conditions involving data latency, the prediction-driven multi-factor scoring recommendation achieved a "first-attempt success rate" (97.3%–98.8%) comparable to ideal real-time recommendation without relying on real-time data streams, consistently outperforming distance-only strategies; (4) This paper presents a complete engineering implementation of the system, encompassing the prediction model, recommendation decision-making, and a user-feedback closed loop.

This study has several limitations: first, the experiments were based on a dataset from a single city, so the gener-

ISSN 3106-7018 (Print) ISSN 3106-5791 (Online)

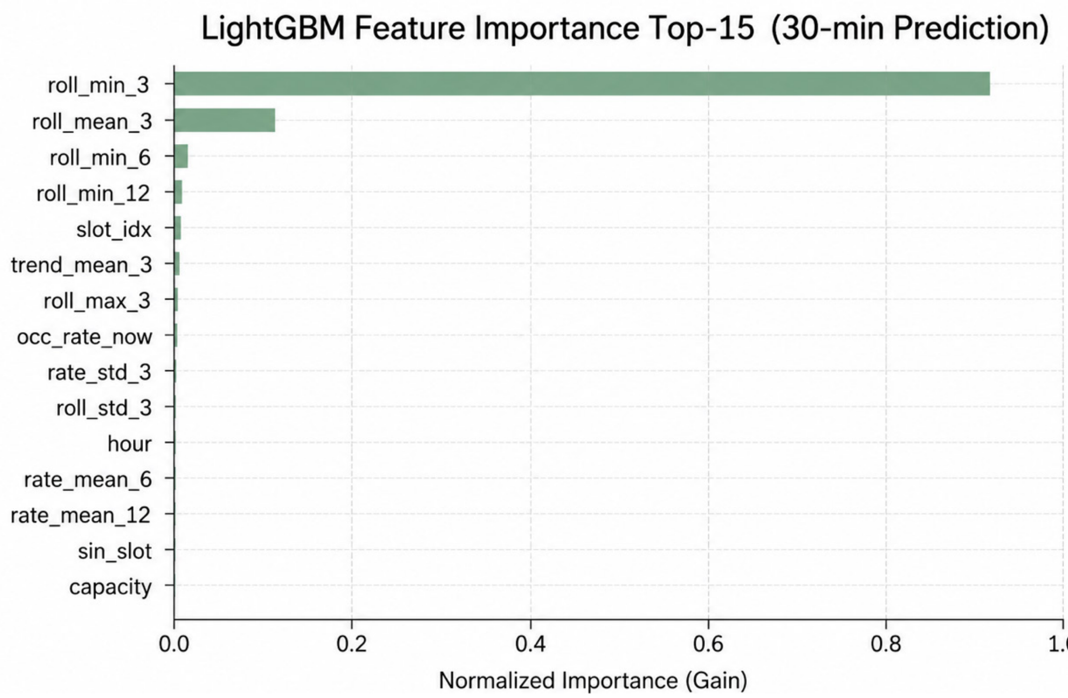


**Figure 3 Top 15 LightGBM feature importances**

alizability of the conclusions to other cities and data conditions remains to be verified; second, the data covered only daytime operating hours (08:00–16:30), excluding nighttime and 24-hour operational scenarios; third, recommendation evaluation relied on stylized simulations where parameters such as the urban layout, vehicle speeds, and retry penalties were based on assumptions, differing from real-world road environments; fourth, although the feedback closed loop has been implemented, the volume of accumulated feedback data is currently insufficient to support a full model retraining iteration. Future work will proceed in three directions: introducing spatiotemporal graph models to capture spatial correlations between parking lots and leveraging the status of neighboring lots to improve individual predictions; validating the method's generalization capabilities using multi-city data and examining the cross-city transferability of feature configurations and scoring weights; and conducting real-world road tests to evaluate prediction accuracy and recommendation effectiveness in operational environments, while using arrival data accumulated via the feedback loop to complete a full model retraining iteration, thereby establishing a comprehensive chain of evidence extending from offline experiments to online validation.

## Reference

[1] Zheng Y., Rajasegarar S., Leckie C. Parking availability prediction for sensor-enabled car parks in smart cities. Proc. 2015 IEEE 10th Int. Conf. on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), Singapore, 2015: 1-6. DOI: 10.1109/ISSNIP.2015.7106902

[2] Rajabioun T., Ioannou P.A. On-Street and Off-Street Parking Availability Prediction Using Multivariate Spatiotemporal Models. IEEE Trans. Intelligent Transportation Systems, 2015, 16(5): 2913-2924. DOI: 10.1109/TITS.2015.2428705

[3] Vlahogianni E.I., Kepaptsoglou K., Tsetsos V., Karlaftis M.G. A Real-Time Parking Prediction System for Smart Cities. Journal of Intelligent Transportation Systems, 2016, 20(2): 192-204. DOI: 10.1080/15472450.2015.1037955

[4] Awan F.M., Saleem Y., Minerva R., Crespi N. A Comparative Analysis of Machine/Deep Learning Models for Parking Space Availability Prediction. Sensors, 2020, 20(1): 322. DOI: 10.3390/s20010322

[5] Stolfi D.H., Alba E., Yao X. Predicting Car Park Occupancy Rates in Smart Cities. Smart-CT 2017, LNCS 10268, Springer, 2017: 107-117. DOI: 10.1007/978-3-319-59513-9_11

[6] Shao W., Zhang Y., Guo B., et al. Parking Availability Prediction with Long Short Term Memory Model. GPC 2018, LNCS 11204, Springer, 2018: 124-137. DOI: 10.1007/978-3-030-15093-8_9

[7] Camero A., Toutouh J., Stolfi D.H., et al. Evolutionary Deep Learning for Car Park Occupancy Prediction in Smart Cities. LION 12, Springer, 2019: 386-401. DOI: 10.1007/978-3-030-05348-2_32

[8] Yang S., Ma W., Pi X., Qian S. A deep learning approach to real-time parking occupancy prediction in transportation networks incorporating multiple spatio-temporal data sources. Transportation Research Part C, 2019, 107: 248-265. DOI: 10.1016/j.trc.2019.08.010

[9] Lin T., Rivano H., Le Mouël F. A Survey of Smart Parking Solutions. IEEE Trans. Intelligent Transportation Systems, 2017, 18(12): 3229-3253. DOI: 10.1109/TITS.2017.2685143

[10] Kotb A.O., Shen Y.-C., Zhu X., et al. iParker—A New Smart Car-Parking System Based on Dynamic Resource Allocation and Pricing. IEEE Trans. Intelligent Transportation Systems, 2016, 17(9): 2637-2647. DOI: 10.1109/TITS.2016.2531636

[11] Dong S., Chen M., Peng L., et al. Parking rank: A novel method of parking lots sorting and recommendation based on public information. Proc. 2018 IEEE Int. Conf. on Industrial Technology (ICIT), Lyon, 2018: 1381-1386. DOI: 10.1109/ICIT.2018.8352381

[12] Geng Y., Cassandras C.G. New "Smart Parking" System Based on Resource Allocation and Reservations. IEEE Trans. Intelligent Transportation Systems, 2013, 14(3): 1129-1139. DOI: 10.1109/TITS.2013.2252428

[13] Ke G., Meng Q., Finley T., et al. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. Advances in Neural Information Processing Systems 30 (NeurIPS 2017), 2017: 3149-3157.

[14] Hochreiter S., Schmidhuber J. Long Short-Term Memory. Neural Computation, 1997, 9(8): 1735-1780. DOI: 10.1162/neco.1997.9.8.1735

[15] Lucchese C., Callegher G., Modenese M., Dassiè S. A comparison of spatio-temporal prediction methods: a parking availability case study. Proc. 37th ACM/SIGAPP Symposium on Applied Computing (SAC 2022), 2022: 1013-1020. DOI: 10.1145/3477314.3507035

[16] Chen T., Guestrin C. XGBoost: A Scalable Tree Boosting System. Proc. 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD 2016), 2016: 785-794. DOI: 10.1145/2939672.2939785

[17] Cho K., van Merriënboer B., Gulcehre C., et al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. Proc. 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014: 1724-1734. DOI: 10.3115/v1/D14-1179

[18] Shoup D.C. Cruising for parking. Transport Policy, 2006, 13(6): 479-486. DOI: 10.1016/j.tranpol.2006.05.005