MULTISTEP SHORT-TERM PARKING FORECASTING

WITH ENSEMBLE AND DEEP-LEARNING MODELS

SUNSE KWON

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
AT THE SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
OF TILBURG UNIVERSITY

STUDENT NUMBER

2076460

COMMITTEE

dr. Sebastian Olier Jauregui
MSc. Federico Zamberlan

LOCATION

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science &
Artificial Intelligence
Tilburg, The Netherlands

DATE

January 13, 2023

WORD COUNT

8359/8800

ACKNOWLEDGMENTS

I genuinely appreciate Dr. Sebastian Olier Jauregui for his guidance. His challenging idea for executing experiments and selecting methodology helped me grow my skills and mindset as a researcher. Also, thanks to my family in South Korea for all the unconditional support.

MULTISTEP SHORT-TERM PARKING FORECASTING

WITH ENSEMBLE AND DEEP-LEARNING MODELS

SUNSE KWON

Abstract

A search for parking lots has been problematic in many urban areas. Many studies focused on comparing models and have yet to consider the effect of features, time windows, time horizons, and regional differences. This study conducted the parking lot's availability prediction with candidate models on univariate and multivariate datasets. It determined suitable time window size and performance on different time horizons with two multistep forecasting techniques. The result showed that three types of features are adequate for parking availability: spatial, temporal, and facility related. Using multivariate data, models showed performance increase. To determine window size, models were tested in three different sizes of sample, reporting that window 3 (using previous 16 timesteps(lag=16) to predict the single timestep ahead, 1 timestep=15minutes) was relevant. Multistep forecasting in the 40-time horizon (10 hours) was conducted in three sizes of sample and five different regions in Singapore (East, West, South, North, and Central). The study reported that Seq2seq models with a hybrid recursive-multioutput strategy were superior across three regions (Central, North, and West). In contrast, Bagging regressor with a recursive strategy showed the highest performance in the East and South area. In that sense, the study proposes multiple models in different regions.

1 INTRODUCTION

1.1 *Problem Statement and Research Goal*

The centralization of the metropolitan area has been accelerated since the industrial revolution. The increasing population caused an increasing number of vehicles in big cities. Consequently, this phenomenon affects urban and transportation-related problems such as traffic jams, accidents, air pollution, and insufficient parking lots. A search for parking lots has been problematic in many urban areas. Even considering factors such as

the high availability of nearby parking lots and the occupancy rate from suburbanites, 90 percent of parking lots are taken during business hours (Litman, 2016). Also, Vlahogianni, Kepaptsoglou, Tsetsos, and Karlaftis (2016) argued that one factor that caused heavy traffic was cars searching for available parking spaces.
extensive research has been conducted predicting parking availability (see Section 2), but most studies only focused on comparing models with limited configurations. From the analysis of literature, although three studies (Alajali, Wen, & Zhou, 2017; Yang, Ke, Cui, Wang, & Murthy, 2022; Zeng, Ma, Wang, & Cui, 2022) utilized multivariate datasets, none of the studies tested the effect of facility-related features(e.g., `car_park_type`, `free_parking`, `night_parking`). also many studies used specific lag values to determine window sizes(Kirby, Watson, & Dougherty, 1997; Van Der Voort, Dougherty, & Watson, 1996; Yang et al., 2022), only one study(Stathopoulos & Karlaftis, 2003) tested time window on different locations. lastly, only two studies(Ji, Tang, Guo, Blythe, & Wang, 2014; Mei, Zhang, Zhang, & Wang, 2020) conducted multistep forecasting while others predicted on the single-step horizon with different shift sizes(e.g., 15 minutes, 30 minutes, and 60 minutes ahead). Based on identified literature gaps, the study aims to simultaneously predict the parking lot's availability with candidate models on univariate and multivariate datasets, different time windows, and different time horizons in five regions.

1.2 *Scientific Relevance*

A problem to be solved in this research is scientifically relevant in terms of contribution to the novelty of predicting parking availability using algorithms that have strength in time series data analysis. In addition, Most earlier papers using ML/DL models did not consider different configurations, such as prediction on different time horizons, time window sizes, and feature combinations. Camero, Toutouh, Stolfi, and Alba (2019); Stolfi, Alba, and Yao (2017) only focused on comparing the model's performance, while Alajali et al. (2017) used additional features. However, it is not related to the parking lot itself. Also, no studies simultaneously compared bagging methods' performance against LSTM and XGBoost. Garg, Lohumi, and Agrawal (2020) compared XGBoost with other ML models, and Sadia, Reza, Alam, and Rahman (2021) compared LSTM and Random Forest Regression. also seq2seq models are adopted forecasting in another domains(Kao, Zhou, Chang, & Chang, 2020; Zhang, Li, & Zhang, 2020), but hardly used in parking prediction. therefore, it is worth comparing all promising models together.

1.3 *Societal Relevance*

The societal benefit of this research is related to predicting accurate parking lot availability using IoT (Internet of Things) sensors to contribute to building a smart city. Bilal, Persson, Ramparany, Picard, and Boissier (2012) argued that actual time data generated by sensors in the parking lot is crucial for smart cities as efficient utilization of mentioned data could be attractive to policymakers or urban planning officials. In addition, considering the estimated number of residents of HDB flats reported as 3.2 million, which is 80 percent of the estimated resident population in Singapore (Housing & (HDB), 2020), Providing more accurate parking availability would reduces the time spent searching for parking spaces for ordinary people.

1.4 *Research Strategy and Research Questions*

The following research questions could be formulated to fill the research gap mentioned in section 1.

> *MQ. To what extent does the combination of spatial, temporal, and facility-related information affect the performance of ensemble and deep learning models in parking prediction?*

The main question could be examined by answering the following sub-questions, as such:

RQ1 *To what extent do individual features of the parking lot affect the model's performance?*

The focus of RQ1 is to find relevant features that impact the models' performance. The spatial feature like x and y geographic coordinates, temporal features such as `day_of_week`, `hour_of_day`, as well as parking facility-related features such as `free_parking`, `night_parking` could be considered. Then, the algorithm performed evaluating the importance of features, which produces a ranking of features correlated with parking availability. Then univariate and multivariate datasets were tested on candidate models to see any difference in the performance.

RQ2 *To what extent does models' performance vary when using a different time window?*

For RQ2, the investigation focused on how different time windows affect the model's performance and determine which window size would be relevant by evaluating performance on different sizes of sample and five different regions.

RQ3 *How does the performance of models vary in different prediction horizons?*

For RQ3, the focus was to examine further performance measurement considering multistep time horizons. Two strategies tested forty timesteps on different sizes of the sample in different regions. Moreover, examined how long the performance of the model would be stable.

RQ4 *How does the performance of models vary in different geographical locations?*

For RQ4, the study created five different sample datasets based on x,y coordinates analysis. Then the performance of models in each region was measured to identify regional tendencies.

2 RELATED WORK

Since machine learning models get interested, many studies compared the performance of ML/DL models as well as statistical models. Camero et al. (2019) compared two RNN models, which used genetic-based strategy and evolutionary-based strategy, with previous work (Stolfi et al., 2017). The comparison showed that RNNs performed similarly to statistical models(Polynomial Fitting, Fourier Series, K-Means, KM-Polynomials, Shift and Phase, and Time Series). It was also revealed that the arbitrary configuration of RNN's architecture would not outperform the models invented earlier. Tekouabou, Cherif, Silkan, et al. (2020) compared the performance between bagging and boosting methods and revealed that the bagging regressor outperformed boosting methods as well as models from the previous study conducted by (Camero et al., 2019; Stolfi et al., 2017). Garg et al. (2020) mainly compared machine learning models' simultaneously, including XGBoost. The study identified that XGBoost had the best performance among other ML models as well as a neural network. Sadia et al. (2021) compared LSTM and random forest regression, but random forest outperformed LSTM given dataset. Moreover, since sequence-to-sequence models started to apply in many domains, but hardly studies used them in parking prediction, it is necessary to look at literature in other fields. Kao et al. (2020) compared two seq2seq models, which used feedforward neural network and LSTM for flood forecasting. The study reported LSTM-based seq2seq models superior to the Feedforward-based seq2seq model across the t+1 to t+6 consecutively. Zhang et al. (2020) studied wind power forecasting and compared attention-seq2seq models with other forecasting models, including Random forest and Adaboost. The study revealed that seq2seq models outperformed Random forest and Adaboost.

On the other hand, three studies (Alajali et al., 2017; Yang et al., 2022; Zeng et al., 2022) utilized multivariate datasets. Alajali et al. (2017) studied

the impact of temporal features (`day_of_week` and `time_of_day`) as well as external features (pedestrians volume, traffic volume) on the model's performance. Mainly performed the comparison of GBRT (Gradient Boosting Regression Tree) against RT (Regression Tree) and SVR (Support Vector Regression) on two datasets which with and without pedestrian data. The study revealed that data with pedestrian features would help increase the models' performance in general. Zeng et al. (2022) conducted a comparison of RNN models ranging from GRU, RNN, and LSTM, as well as a combination of different types of RNN cells. comparison was made between univariate and multivariate dataset which contains `day_of_week`, weather, vacation as a features. all candidates showed an increase in performance in multivariate datasets. This revealed that weather and holiday, as well as a temporal feature, would be correlated to the parking prediction. Yang et al. (2022) used `lot_id`, `lot_type`, `time_of_day`, `day_of_week`, weather condition, as a features. the study revealed that `lot_id`, `lot_type` showed the highest correlation on prediction with 2.32 percent contribution, also, `time_of_day`, `day_of_week` influenced 2.04 and 1.98 percent. Also, weather conditions contributed 1.01 percent on average.

Many studies used specific lag values to determine window sizes(Kirby et al., 1997; Van Der Voort et al., 1996; Yang et al., 2022), only one (Stathopoulos & Karlaftis, 2003) tested time lag values on different locations. the study used state-space models and compared them with ARIMA on five locations and six discrete periods in a day. The study tested relevant lag values of five locations and six periods of each location. Moreover, it revealed how the lag values of the other four locations correlated with one location.

Two studies(Ji et al., 2014; Mei et al., 2020) conducted multistep forecasting while others predicted on a single-step horizon with different shift sizes(e.g., 15 minutes, 30 minutes, and 60 minutes ahead). Ji et al. (2014) conducted single-step and multistep forecasting using Lyapunov exponents methods, wavelet neural network, and hybrid models, which combine two models (WNN-LE). for multistep forecasting, WNN with less than four timesteps performed accurately, while LE methods was superior in a situation when timestep value was over four. Therefore, the study proposed switching models based on threshold value 4. Mei et al. (2020) conducted multistep prediction mainly compared Fourier-Transform Least Square Support Vector Regressor (FT-LSSVR) against Least Square Support Vector Regressor (LSSVR) and neural network. they used two strategies(recursive and direct strategy) to conduct multistep forecasting, then tested six different parking lots on both short-term and long-term periods. Interestingly, the direct strategy was better than the recursive strategy in general. The study Proposed FT-LSSVR depending on the threshold if the step value is

less than the value, then includes irregular components; otherwise, ignore it to control non-linearity in long-term prediction.

State-of-the-art techniques have been introduced in the recent few years. Yu, Yin, and Zhu (2017) proposed a Spatio-temporal graph convolutional network that jointly captures spatial and temporal correlation by the Chebyshev polynomial. It overcomes the limitation of RNN-based models as their structure cannot handle spatial dependency. Therefore, the spatial correlation can be relatively well considered using the graph network layer in the middle. Xiao, Jin, Hui, Xu, and Shao (2021) proposed a hybrid spatial-temporal graph convolution network with an attention mechanism. It compared with other baselines such as Historical Average, ARIMA, LSTM, DCRNN, STGCN, and ASTGCN. The difference from existing methods is that it considered the distribution of parking occupancy duration as a spatial feature. They predicted 15 minutes, 30 minutes, and 60 minutes later time points, and the proposed model outperformed its competitors in long-term periods but was similar in the short-term. Yang et al. (2022) used an attention mechanism with LSTM to predict short-term and long-term time horizons. With that model, comparison was made with other baseline models such as the Karlman filter, Random Forest, RNN, LSTM, google parking difficulty estimation, and PewLSTM. The result showed that the performance of attention with LSTM was superior to other candidate models for the long-term horizon. It is interesting to mention that the study used two datasets: real-time data for short-term (lookback the previous 16 timesteps) and historical data for long-term (containing values that were measured on the same date and time in the past, containing 16 timesteps). Using attention, the model can pay more attention to which attribute to be weighted. From the study, when predicting 5 min and 10 min future time steps, the real-time dataset was more considered; however, for 30 min and 1 hour, 2 hour later steps, the historical dataset was more considered.

Since machine learning and deep learning hype, many studies have used machine learning models rather than statistical or state space models. Nevertheless, the limitation of ML/DL studies mainly focused on comparing models' performance on single-step forecasting with limited configurations. Moreover, most neglected different time horizons and different time windows and how the performance vary in different locations. Therefore, further research needs to be conducted.

3 METHOD

3.1 *Algorithms*

3.1.1 *Bagging Regressor*

Firstly, the Bagging regressor was selected as one of the ensemble learning methods. The final model's prediction is formed by aggregating results from several weak learners using different random subsets known as a bootstrap to achieve a more robust and better prediction than that from a single model. In the prediction task, high variance is problematic as it leads to overfitting the model on the dataset. The risk of high variance could be reduced by averaging with randomized subsets and weak learners, such as decision trees. It was a rationale to choose it as Tekouabou et al. (2020) revealed that the bagging regressor outperformed boosting methods and RNN.

3.1.2 *XGBoost Regressor*

Second, the XGBoost regressor was selected one of the candidates as it is boosting category of ensemble methods. Chen and Guestrin (2016) proposed an enhanced version of the gradient-boosting algorithm by adding regularization terms, using shrinkage and column subsampling, handling sparsity in data, and using weighted quantile sketch in the structure of the algorithm. XGBoost still learns by adding weak learners to strong learner sequentially, but within the weak learner level, processing can be done in parallel with column block. Garg et al. (2020) reported that XGBoost had the best performance among other ML models, but it did not compare with the bagging regressor, which is reported from Tekouabou et al. (2020). Therefore, there was a necessity to compare bagging regressor with XGBoost.

3.1.3 *Long Short-Term Memory (LSTM)*

LSTM was chosen as it is one of the variants of the Recurrent neural network. Hochreiter and Schmidhuber (1997) argued that RNN has limitations, such as vanishing or exploding weights during backpropagation. The study proposed an architecture with an input, output, and forget gate to overcome low performance in the long-term prediction task. Therefore, the memory cell will dismiss unimportant information while preserving important information over time. LSTM is suitable for our task as (i) the nature of the dataset is sequential, and the architecture of RNN receives and processes data sequentially. (ii) LSTM will be remarkably better than RNN as our task is time series forecasting, which need to handle temporal

dependency better. (iii) as Camero et al. (2019) reported that RNNs exhibited similar performance against the models invented earlier, the study included LSTM in one of the candidate models.

3.1.4 *Gated Recurrent Unit (GRU)*

Gated Recurrent Unit (GRU), introduced by Cho, Van Merriënboer, Bahdanau, and Bengio (2014), is one of the variants of a recurrent neural network. It is similar to LSTM but lacks an output gate, meaning less complexity, enabling the train and running faster. The architecture of GRU consists of two gates which are the update gate and the reset gate. Specifically, the update gate decides which information pass to output, and the reset gate determines whether information needs to be retained. The reason for the selection of GRU was that it shows similar performance with LSTM in sequential data processing tasks, including natural language or speech recognition. It is rational to assume that GRU could show similar performance with LSTM.

3.1.5 *Sequence to Sequence (Seq2seq)*

Sequence to sequence model is based on encoder-decoder architecture. It consists of two partitions. The first is the encoder, and the second is the decoder. As can be seen from the name, the encoder part receive the input sequence and process it by the RNN cell to produce a context vector. Decoder parts receive a context vector as an input, and then it produces outputs sequentially by feeding the previous output and hidden state to the next cell. seq2seq models were adopted forecasting in another domains(Kao et al., 2020; Zhang et al., 2020), but hardly used in parking prediction. the main reason for selecting seq2seq was to examine how performance vary between models which produce single output and seq2seq, which produce multi-output to test RQ3. in the initial stage, the study planned to compare only the above four models. However, comparing single-output models with multi-output models would provide more robust result.

3.2 *Experimental Setup*

3.2.1 *Dataset*

Two datasets in the experiment are real-time residential parking lot data and parking facility data to the corresponding parking lot. Regarding the former, real-time data could be retrieved from API every minute. in this study, data was collected from 22-09-2022 to 01-11-2022 in 15 minutes intervals and stored in CSV format. The main reason for setting the 15-

minute interval of data collection was based on earlier studies. Also, it could be rational to assume a minimal parking time will be around 15 minutes in case of delivery of goods or short-term visits nearby the area. The total size of the raw data was 10072795 rows and six columns. Total 1968 parking lots were collected from the sensors which installed in each location. It consists of six features from Table 1

Table 1: Details of the real-time dataset

No	Feature	Description	Type
1	total_lot	total num of parking spaces in each carpark	Num
2	lot_type	type of lot represented by an alphabet	Cat
3	lots_available	num of available parking spaces in each car park	Num
4	carpark_number	the unique identifier of each car park	Cat
5	update_date	date represented by date-month-year	Num
6	update_time	time represented by second-minute-hour	Num

Regarding the latter, Table 2 shows each parking lot's facility information which is available in CSV format.

Table 2: Details of each parking lot information

No	Feature	Description	Type
1	car_park_no	the unique identifier of each carpark	Cat
2	address	address of parking lot	Cat
3	x_coord	x-geographic coordinate	Num
4	y_coord	y-geographic coordinate	Num
5	car_park_type	type of parking lot	Cat
6	type_of_parking_system	electronic or coupon parking system	Cat
7	short_term_parking	available parking period specified	Cat
8	free_parking	free parking availability	Cat
9	night_parking	night parking availability	Cat
10	car_park_decks	num of decks in correspond parking lot	Num
11	gantry_height	value represents the gantry height	Num
12	car_park_basement	type of carpark is basement or not	Cat

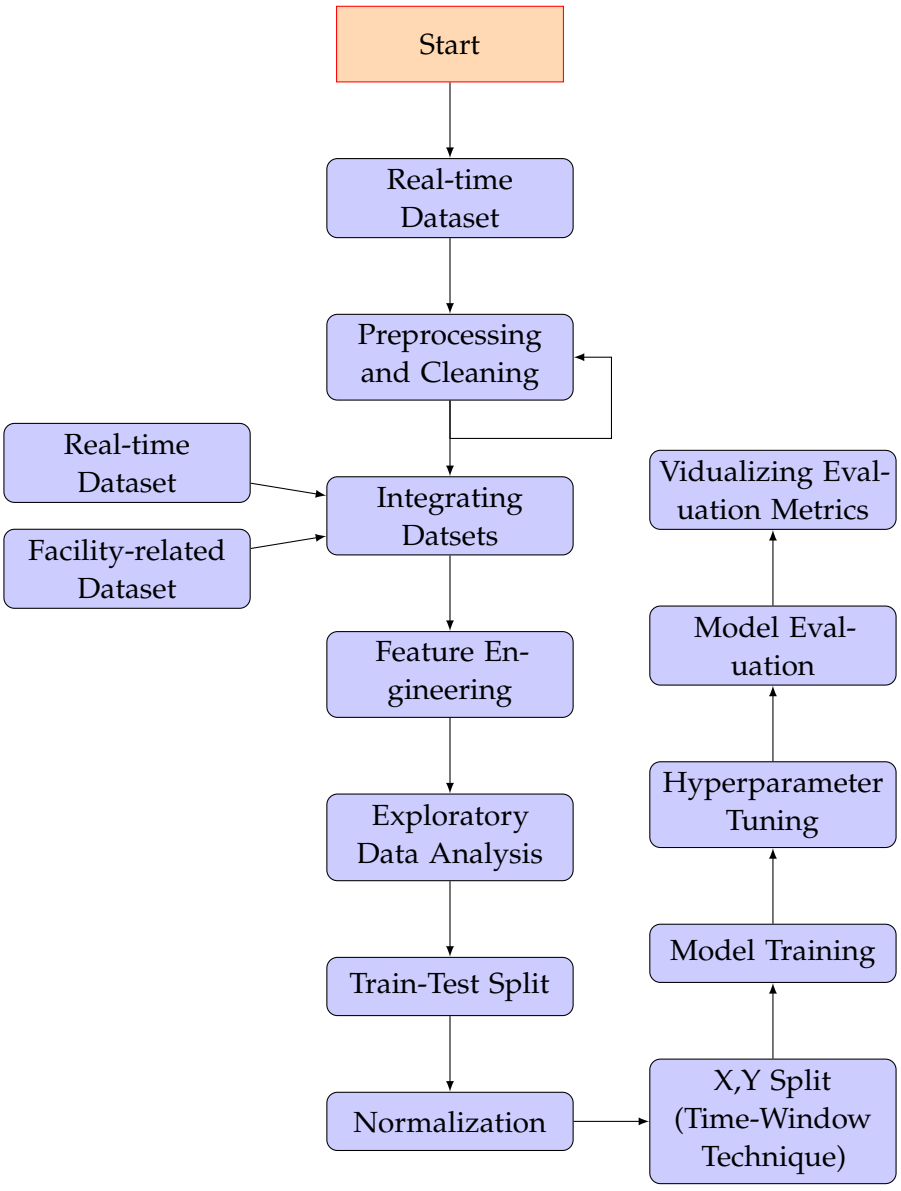
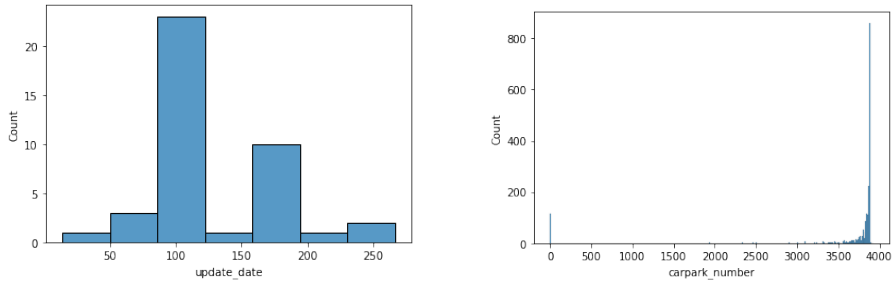


Figure 1: Flowchart of experiment setup

3.2.2 *Cleaning/Preprocessing*

Issue 1: Irregular data collection
Data was not collected every 15-minute interval, especially in October, due to the system being down during the data collection period. Figure 2(a) clearly shows the irregularity of timesteps per date. In the original setup, collected data should be 96 timesteps per date (4*24 = 96, where 4*15min = 60min), but it varies less or higher than 96 timesteps. It means that data was collected in a shorter time interval than the original setup in case of timesteps are less than 96. to maintain data in every 15-minute time interval, the DateTime index was rounded to 15 minutes nearest time points and dropped duplicated values.
Issue 2: Different length of timesteps per parking lot
As the entire time length per parking lot has differed (Figure2(b)), assumed that the API call during the retrieval was inconsistent. Therefore, some parking lot data was not collected. Justification was needed regarding which parking lot to retain or to drop. The study empirically examined parking lots where less than 1850 timesteps are less frequent, thus, dropped for further processing convenience.



(a) Counts timesteps per date of an individual parking lot

(b) Counts time periods per individual parking lot

Figure 2: Issues identified during individual parking lot analysis

Issue 3: Homogeneous availability values over the entire period
The parking lot in which the availability values were all the same during the entire data collection period was dropped. It could be assumed hypothetical reasons, such as the sensor installed malfunctioning. To detect that parking lot, for each parking lot, the most frequent `lots_available` value was compared with the entire timestep of the parking lot. If the values were identical, the parking lot was dropped.
Issue 4: `total_lot`, `lot_type` features
Table 1 shows that the characteristic of two features (`total_lot`, `lot_type`) is the facility attached. `total_lot` and `lot_type` should be determined upon the facility's construction and should not vary over time. Its value

should have a constant across entire timesteps. However, study identified more than one value. Therefore, the parking lot in which `total_lot`, and `lot_type` showed more than one value was identified. The incorrect values were replaced by the majority accordingly.

Issue 5: `lots_available` feature

It is rational to assume that the maximum value of `lots_available` should be less or identical to the `total_lot` value. Therefore, the availability higher than the `total_lot` value was considered invalid. An instant feature called difference was created to identify invalid values, which is the `total_lot` was subtracted by `lots_available`. Two parking lots were dropped where the negative values in the difference column were detected. At the same time, for parking lots where only a single negative value was detected in the difference, the availability value was imputed by the mean value between the previous and later timestep.

Issue 6: Data points that were out of collection periods

Data points that were out of collection periods were detected during the examination. after treating issue 3, there were still a few data points existed which date was earlier than 22-09-2022. Therefore, those parking lots were dropped.

After cleaning the stage, 3056876 rows remained, 1113 parking lots were dropped, and 855 parking lots were retained.

3.2.3 *Data Integration and Feature Engineering*

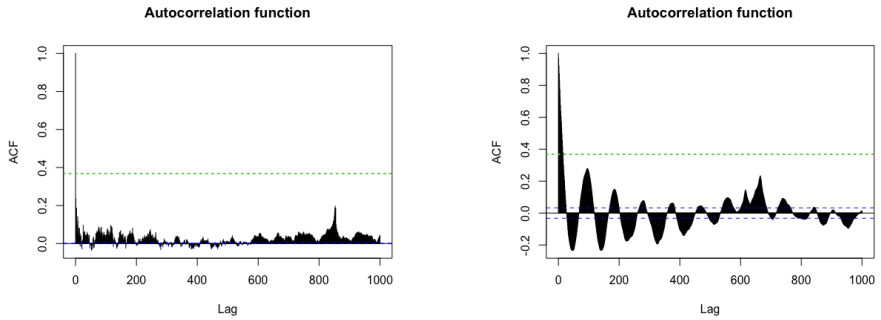
First, the facility dataset was merged with a real-time dataset by matching `carpark_number` on real-time data with the `car_park_no` in facility-related data. Next, two features (`lot_type`, `type_of_parking_system`) were dropped as their values were the same across the entire dataset. 5 categorical features (`car_park_type`, `short_term_parking`, `free_parking`, `night_parking` and `car_park_basement`) transformed to dummy variables. In addition, two DateTime features (`day_of_week`, `hour_of_day`) were added to the dataset. Lastly, for `carpark_number`, categorical values were replaced by numerical values, which were assigned 0 to 854 for feeding into the model. each value served as an identifier.

3.2.4 *Exploratory Data Analysis (EDA)*

(i) Autocorrelation Analysis

To find the relevant time window, the timeLag function in the nonlinearTseries package in R was used with the autocorrelation technique. Figure 3(a) revealed that lag=1 for 855 parking lots as the autocorrelation function reported that the first value that crosses the threshold around 0.38 is rel-

evant. On the other hand, when measured the time lag of an individual parking lot(Figure 3(b)), the function returned lag=16.

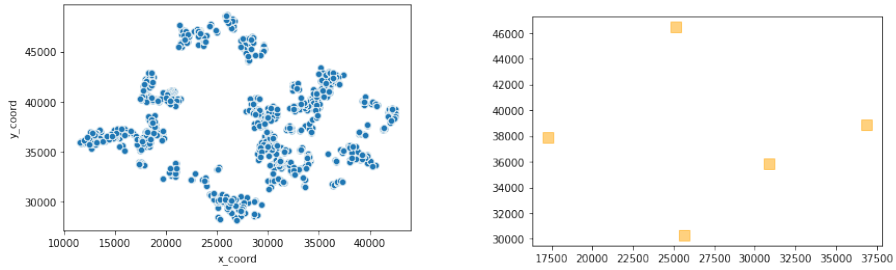


(a) 855 parking lots (lag=1)

(b) Individual parking lot (lag=16)

Figure 3: Autocorrelation of 1000 lags of `lots_available` feature

(ii) K-mean Clustering Analysis
Singapore geographically could be partitioned into five regions (central, east, west, north, and south). For the testing performance of the model on different regions, the study detected five centroids using the 5-fold k-mean clustering technique. Figure 4(b) is five centroids based on the x and y coordinates of 855 parking lots. Accordingly, a subset of the dataset (n=10 to 12) was created for each region based on coordinates close to each centroid.



(a) X,Y coordinates of 855 parking lots

(b) Five centroids based on 5-fold K-mean clustering analysis

Figure 4: Analysis of the geographical location of 855 parking lots

3.2.5 *Train-Test Split*

For machine learning models, the last 1-week period timesteps were assigned to Test-set, which was 672 data points, and the rest were assigned to the Train-set. train-test split for each parking lot was done by itera-

tion. Dataset stacked one by one row-wise using for loop. Aside from the test-set, the train-set was divided into train and validation sets for the neural network models. Validation-set was assigned to the last 4-day period, which was 380 timesteps. Rest was assigned to the train-set.

3.2.6 *Normalization*

Normalization was done by the MinMax scaler to transform the data in a specific range (range 0 to 1). The formula is following:

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{1}$$

Normalization was needed due to each feature had different scales. Scaling values to be specific ranges was needed for faster learning speed. After fitting training data with a scaler, then used trained scaler to transform the test and validation set.

3.2.7 *Time Window Technique*

Y,Y were split based on the time window technique to test RQ2. Three different window sizes were tested. window 1 uses a single previous time step (15 min) to predict a timestep ahead. window 2 uses four previous timesteps (1 hour) to predict a timestep ahead, and window 3 uses 16 previous timesteps (4 hours) to predict a timestep ahead. The reason for selecting three window sizes was based on autocorrelation analysis. Also, to answer RQ3, unlike a single timestep that was assigned to y, four timesteps were assigned to y for seq2seq models, which produce multioutput. For seq2seq models, modified window 3 was needed, which uses 16 previous timesteps to predict the next four consecutive timesteps (1 hour).

3.2.8 *Hyperparameter Tuning*

Hyperparameter tuning was conducted by fitting models using window 1(uses the previous timestep to predict the single timestep ahead). window 1 was selected based on autocorrelation analysis. (With 855 parking lots, time lag size = 1 was relevant).

Bagging Regressor
GridSearchCV from the sklearn package was used to find the best combination of hyperparameters. Due to the nature of data as time-series, TimeSeriesSplit function was used for cross-validation. For search space, `n_estimators` and `max_samples` were considered important hyperparameters to tune. Shahhosseini, Hu, and Pham (2022) tried `n_estimators`: 100,

200, 500 and max_samples: 0.7, 0.8, 0.9, 1.0 as search space. These values were used in GridSearchCV search space with 5-fold cross-validation. The optimal value of n_estimator and max_samples were 500 and 0.7, respectively.

XGBoost Regressor

Tekouabou et al. (2020) considered learning rate and n_estimators were vital hyperparameters for boosting methods. Shahhosseini et al. (2022) tried gamma: 5, 10, learning rate: 0.1, 0.3, 0.5, n_estimators: 50, 100, 150, max_depth: 3, 6, 9 for search space. Thus, those hyperparameters were chosen for search space. The study conducted GridSearchCV with 5-fold cross-validation using TimeSeriesSplit. The optimal gamma value was 5, the learning rate was 0.3, the max_depth was 3, and the n_estimators were 50.

LSTM and GRU

Sadia et al. (2021) tried two LSTM layers with 100 neurons per layer. Dropout layers were followed by each LSTM cell with a rate of 0.4. Ten epoch was selected, and the batch size was 40. with that architecture and values, the structure of the neural network was set as two LSTM/GRU layers and dropout layers followed by each of the recurrent cells. Then RandomSearch from keras tuner was used for following search space. number of neurons: 100, 150,200, dropout rate: 0, 0.1, 0.2, 0.3, 0.4, 0.5. Table 3 is the result of RandomSearch.

Table 3: Best hyperparameters of LSTM and GRU

Model	Hyperparameter	Value
LSTM	First layer unit	150
	First layer dropout	0.1
	Second layer unit	100
	Second layer dropout	0
GRU	First layer unit	100
	First layer dropout	0
	Second layer unit	150
	Second layer dropout	0.1

Sequence to Sequence (Seq2seq)

Zhang et al. (2020) used four recurrent cells with 100 neurons per layer for both the encoder and decoder. Due to computational resource limitations, this study naively adopted values from the literature. In addition,

epochs=10 were chosen for tuning with early stopping=3. Table 4 is the best hyperparameter configuration.

Table 4: Best hyperparameters for both LSTM and GRU seq2seq models

LSTM/GRU	Hyperparameter	Value
Encoder	First layer unit	100
	Second layer unit	100
	Third layer unit	100
	Fourth layer unit	100
Decoder	First layer unit	100
	Second layer unit	100
	Third layer unit	100
	Fourth layer unit	100

3.2.9 *Robustness Check of Models*

To test performance on the different horizon, the study needed to check the model's performance on train-test set. All models were trained with window 3 and the best hyperparameters configuration. for bagging regressor, Table 5 showed comparison of train and test errors as well as $\mathbf{R^2}$ score. RMSE revealed that test errors were slightly higher than train errors. Also, $\mathbf{R^2}$ values were close to 1, showing that models explain the variance well. bagging regressor creates a sampling distribution of sample `y_predict` values with aggregation. Therefore it could handle high variance risk to ensure robustness to the overfitting. XGBoost showed that both train and test errors are identical across epochs (Figure 5). Moreover, it seemed well-fitted, as test errors did not increase. For deep-learning models(Figure 6), models trained with early stopping with patience=3. Therefore, its epoch length differed per model. all models showed validation errors were lower than training errors, which implies a good fit.

Table 5: Comparison of train-test-error for ensemble models

Model	RMSE(train)	RMSE(test)	$\mathbf{R^2}$(train)	$\mathbf{R^2}$(test)
Bagging	0.00156	0.0026	0.99976	0.99934
XGBoost	0.00563	0.00533	0.99693	0.99725

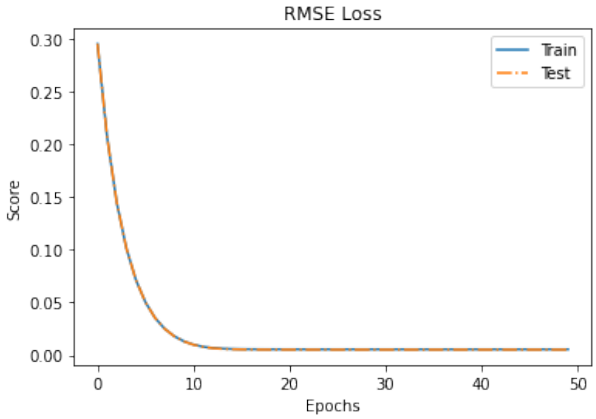
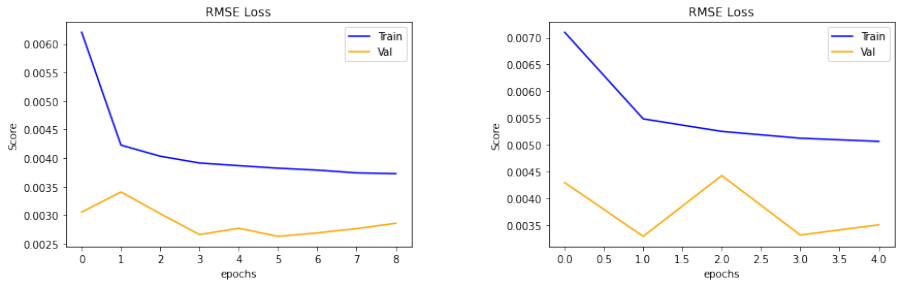
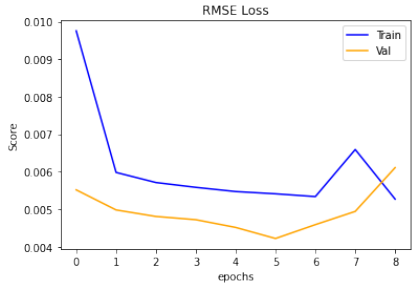


Figure 5: Comparison of train-test RMSE of XGBoost regressor

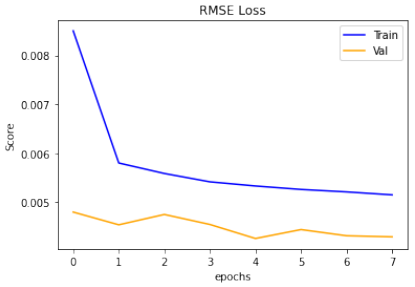


(a) LSTM

(b) GRU



(c) LSTM-seq2seq



(d) GRU-seq2seq

Figure 6: Comparison of train-test RMSE of deep-learning models, MSE plot also available in Appendix (page 34)

3.2.10 *Multistep Forecasting Strategy*

There are numerous existing methods to conduct multistep time-series forecastings, such as direct strategy, recursive strategy, a combination of direct and recursive strategy, and multioutput strategy (Taieb & Atiya, 2015). In this study, the recursive strategy and hybrid recursive-multioutput strategy

were compared. The recursive strategy was chosen due to lower computational cost than the direct strategy, which requires multiple models to be trained. Also, the hybrid recursive-multioutput strategy was chosen for comparison as a model with a recursive strategy only produce a single output.

(i) Recursive Strategy
Four models (Bagging regressor, XGB regressor, LSTM, GRU) were used for multistep forecasting with a recursive strategy. Ensemble models received the last batch of Train-set while the last batch of Validation-set was used for deep learning models. x,y were split using window 3. Models were trained to predict a single step ahead. The prediction output at time t was refed to the model to predict at time t+1 and so on. Forty steps ahead prediction was achieved by iterating 40 times.

(ii) Hybrid Recursive-Multioutput Strategy
A combination of multioutput and recursive strategies could be called a hybrid recursive-multioutput strategy in convenience. For LSTM-seq2seq and GRU-seq2seq, the encoder received the last batch of a validation set with modified window 3 (16 timesteps to predict the next four consecutive timesteps). The decoder sequentially generated four timesteps as an output. Then 40 steps ahead prediction was achieved by iterating 10 times. The reason for predicting only four multioutput from seq2seq models was to ensure high prediction accuracy. For the recursive strategy, if the output of the first prediction is inaccurate, the following output could quickly becomes worse.

3.3 *Evaluation Methods*

The following evaluation metrics were selected to evaluate the model's performance for the regression problem. Mean squared error (MSE), Mean absolute error (MAE), and Root mean squared error (RMSE). The reason for selecting those measures was that they are calculating errors between the predicted value and actual value for each observation. In general, MSE and RMSE penalize significant errors more than MAE due to using squared terms. Nevertheless, RMSE is more widely used than MSE as it has the same unit size as an actual outcome, so its values are more relevant for interpretation. The lower the MSE, MAE, and RMSE values, the higher accuracy of the models. In addition to error metrics, $\mathbf{R^2}$ was used. It represents the proportion of the variance in the dependent variable that is explained by the independent variables in a regression model. It is a measure of how well the model fits the data. In this study, MSE was

chosen for the loss function to minimize by ADAM optimizer as it can be differentiable compared to MAE. For validation loss for neural networks, RMSE was chosen. RMSE was primarily examined over other metrics for comparison of the performance of models. The reason was that even MAE is more robust to the outliers as it does not amplify values due to the lack of a squared term, outliers were well-treated at the cleaning stage. Hence, RMSE would be a better option.

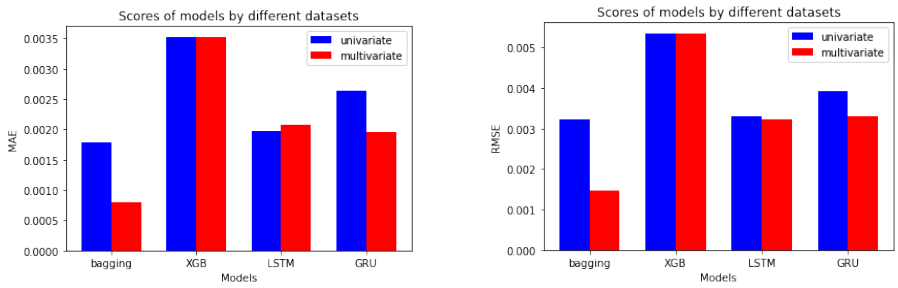
3.4 *Software*

Software usage: programming languages used were mainly R4.1.3 and python 10.3. IDE was visual studio code to process and build models, and R studio was used for the EDA process. Google Colab pro was used for neural network modeling, tuning, and analysis. Package usage would be NumPy, pandas, matplotlib, seaborn, sklearn, and kreas for python, nonlinearTseries package for R, and APSchedular package used for data collection.

4 RESULTS

4.1 *Effect of Features*

Two datasets were used to test the effect of features. D1 is a univariate dataset that only contains `lots_available`, while D2 is a multivariate dataset that contains 13 features. For selecting features for D2, RFE (recursive feature elimination) from sklearn package was used to select half of the entire feature set(24 features). It gave ranking to features with respect to `lots_available`. Selected features for D2 were `day_of_week`, `hour_of_day`, `lots_available`, `total_lot`, `carpark_number`, `car_park_type=MULTI-STOREY CARPARK`, `free_parking=SUN&PH7AM-10.30PM`, `car_park_decks`, `free_parking=NO`, `short_term_parking=WHOLE DAY`, `x_coord`, `y_coord`, `gantry_height`. Then, four models (Bagging regressor, XGBoost regressor, LSTM, GRU), which were trained with window 1 (see 3.2.7) used for comparison of D1 and D2.



(a) MAE (b) RMSE

Figure 7: Comparison of performance of four models using the 855 parking lots (whole sample) between D1 (univariate) and D2 (multivariate)

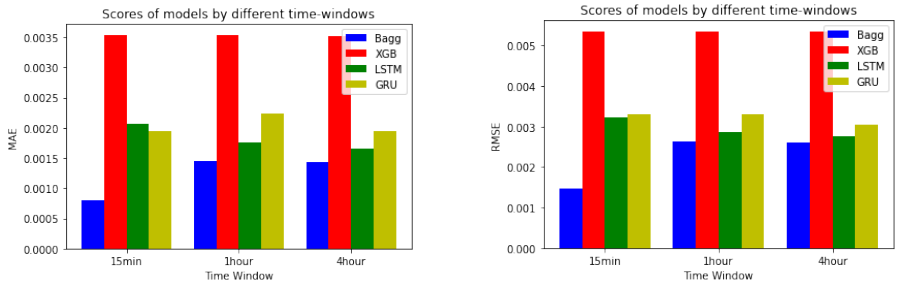
Two multivariate models indicated a performance increase (Bagging regressor, GRU), while XGB Regressor has not exhibited a difference in performance between the two datasets (Figure 7). Note that LSTM (Figure 7, a) showed a multivariate model with less performance than the univariate model. On the other hand, Figure 7 (b) exhibited the opposite result.

4.2 *Performance on Different Time Windows*

To asnwer RQ2, models were tested on window 1, window 2, and window 3 (see 3.2.7) to determine the appropriate time window size. Those three window sizes were selected based on EDA autocorrelation analysis. The expected suitable window size would be window 1 for 855 parking lots, while window 3 would be applicable for an individual parking lot. To answer RQ4, models were tested in five different regions with different subset levels(group level: 10 to 12 parking lots per region, individual level: 1 parking lot per region). Those regions were split based on 5-fold k-mean clustering using x,y coordinates. The result revealed that the five centroids and samples near each centroid were selected.

4.2.1 *855 Parking Lots (Whole Sample)*

From Figure 8, The Bagging regressor exhibited the highest performance at window 1 but dropped the performance when increasing window sizes. XGBoost did not show performance change across the different time windows. While deep learning models showed a slight change in performance across window sizes, the change was moderate. Thus, window 1 would be applicable for 855 parking lots as it accurately predicted the next timestep by using only one previous timestep.

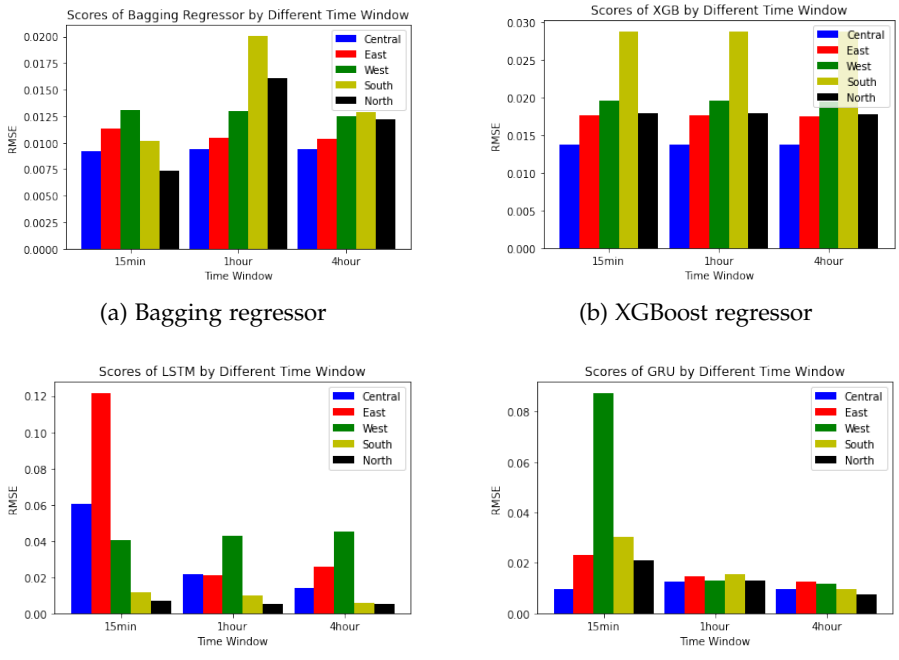


(a) MAE (b) RMSE

Figure 8: Comparison of model's performance on three-time windows at 855 parking lots (window 1=15min, window 2=1hour, window 3=4hour)

4.2.2 *Five Regions (Group Level)*

Overall, based on a comparison of RMSE, the Bagging regressor displayed the best performance at window 1. In contrast, it dropped the performance at window 2 and slowly recovered at window 3. XGBoost was not displayed performance change across window size. Deep learning models increased performance from window 1 to window 3. (Figure 9, (c),(d)).



(a) Bagging regressor (b) XGBoost regressor

(c) LSTM (d) GRU

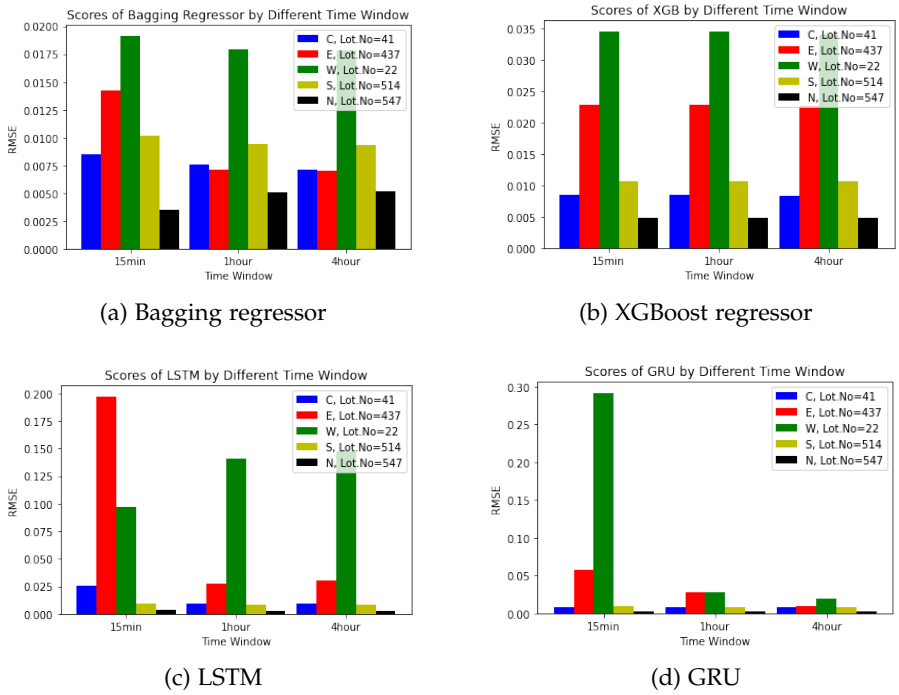
Figure 9: Comparison of the model's performance on three-time windows at a group level in five regions(window 1=15min, window 2=1hour, window 3=4hour)

(1) Central area, Ensemble models were not changed RMSE across time windows, while LSTM did increase it when increasing the time window size. also, GRU reported a performance drop in window 2 despite recovering at window 3. **(2) North area**, The performance of the bagging regressor dropped at window 2 and moderately recovered at window 3 though did not reached the value of window 1. Also, the XGBoost regressor exhibited no difference in performance across different windows. LSTM displayed no change in performance across time windows. GRU showed an increase in performance when increasing the window size. **(3) West area**, Ensemble and LSTM models did not demonstrate much difference in performance across the time windows, while GRU showed a performance increase in higher time windows. **(4) East area**, Ensemble models did not exhibited performance change across time windows. While deep learning models showed an increase in performance when increasing time windows. **(5) South area**, XGBoost did not display much difference in performance across time windows. The bagging regressor dropped performance at window 2 despite recovering at window 3. Deep learning models exhibited an increase in performance when increasing the window size.

4.2.3 *Five Regions (Individual Level)*

Overall, A comparison of the models' RMSE was conducted on individual parking lot levels in five regions. Except for the XGBoost regressor, all models displayed performance increases with higher time windows (Figure 10).

(1) Central area, for the Bagging regressor, a slight performance increase exhibited when the time window increases. The XGBoost not displayed any change across time windows. LSTM demonstrated an increase in performance when increasing time windows. GRU indicated no difference in performance across window sizes. **(2) North area**, The bagging regressor showed a performance drop when increasing time windows. XGBoost reported no difference in performance across different time windows. LSTM and GRU both exhibited slight changes across time windows. **(3) West area**, For the bagging regressor, it displayed a slight increase in performance when the time window increased. XGBoost showed no change in performance across windows. LSTM exhibited a performance drop when increasing time windows. GRU showed a performance increase when increasing the time window. While bagging regressor change was not significant, GRU showed a stiff increase in performance. **(4) East area**, The performance of the bagging regressor increased at window 2 and the same at window size 3. XGBoost indicated no change across time windows. Deep learning models increased performance when the window



(a) Bagging regressor

(b) XGBoost regressor

(c) LSTM

(d) GRU

Figure 10: Comparison of the model's performance on three-time windows at an individual level in five regions(window 1=15min, window 2=1hour, window 3=4hour)

was higher, especially LSTM had a drastic change of values. **(5) South area**, Except for the XGBoost regressor, three models displayed performance increases across window sizes. Nevertheless, the change was moderate.

Based on analysis, identified findings are the following: (i) When using the whole sample, the performance of models did not vary across window sizes. When dataset size decreases on a smaller scale, the performance of models varied based on different time window sizes and regions. (ii) Ensemble models are less affected by the window sizes. However, Deep learning models are sensitive to time windows. (iii) When comparing models' performance in five regions, East and West areas displayed a drastic change of performance across time windows while central, south and north areas exhibited not much affected by time windows.

4.3 *Performance on Different Time Horizon*

To answer RQ3, six models were tested on first 40 consecutive time horizon on test-set. as mentioned previous section, To answer RQ4, models were

tested in five different regions with different subset levels(group level: 10 to 12 parking lots per region, individual level: 1 parking lot per region).

4.3.1 *855 Parking Lots (Whole Sample)*

When using the whole sample, Bagging regressor and LSTM-seq2seq and GRU-seq2seq exhibited the highest performance, respectively, LSTM, GRU, and XGBoost. Unlike XGBoost, LSTM, and GRU, Bagging regressor and seq2seq models displayed stable prediction across 40 timesteps.

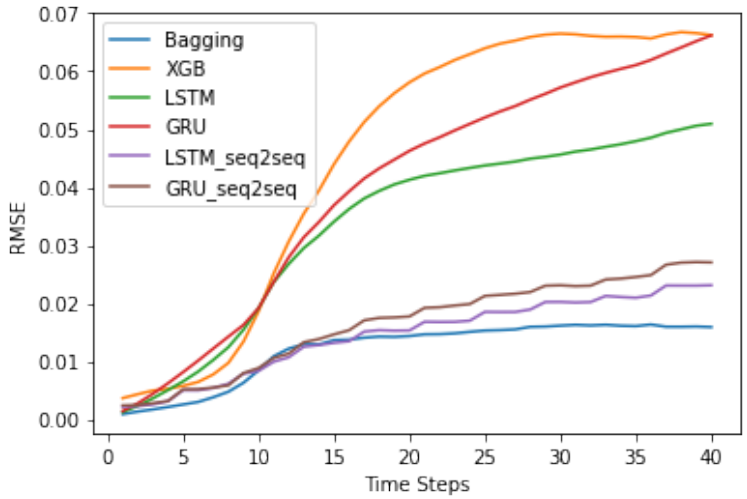
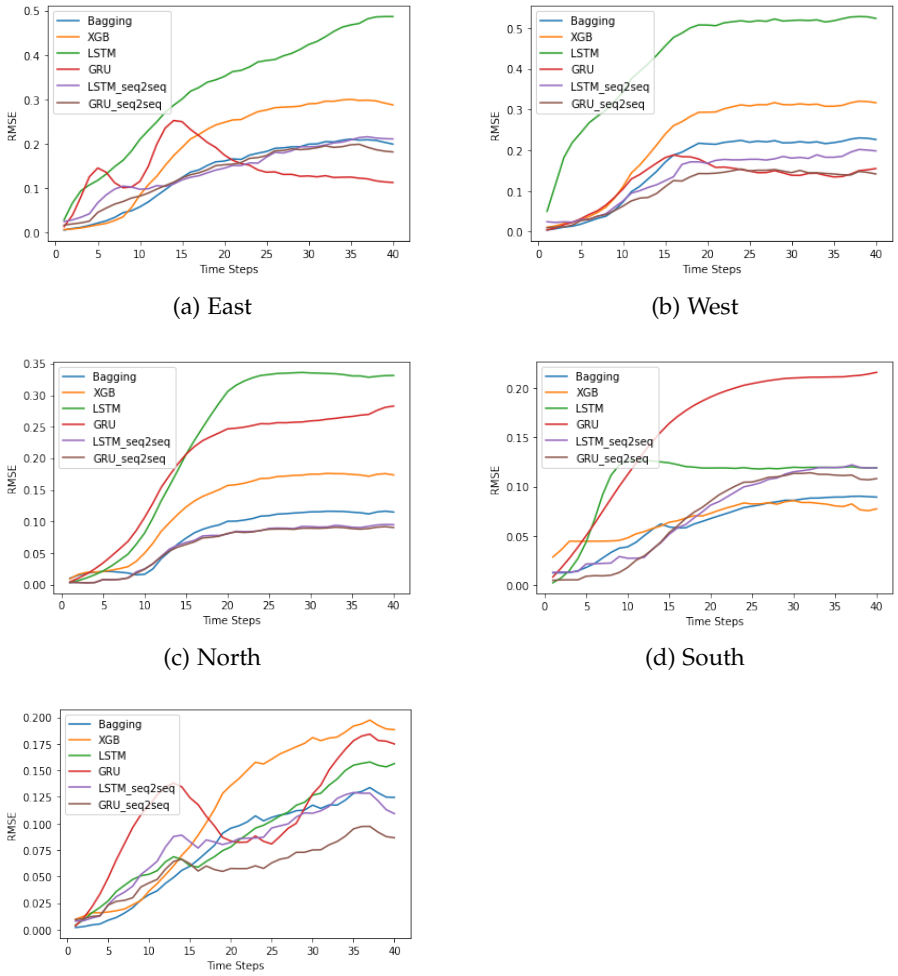


Figure 11: Comparison of 40-timesteps (10 hours, 1-timestep=15 minutes) forecasting horizon of models using 855 parking lots (whole sample)

4.3.2 *Five Regions (Group Level)*

(1) Central area, GRU-seq2seq displayed the highest performance, followed by the Bagging regressor and LSTM-seq2seq. The hybrid strategy generally performed better than the recursive strategy. XGBoost displayed a stiff increase of error after 8 step horizon, while GRU showed fluctuation of prediction values across the time horizon with two peak points. **(2) North area**, the hybrid strategy displayed superior performance across the whole horizon than the recursive strategy. Among models that used recursive strategy, the Bagging regressor also exhibited the best performance, while LSTM and GRU indicated a drastic increase of error after 5 step horizon. **(3) West area**, GRU-seq2seq models demonstrated the highest performance, followed by LSTM-seq2seq. for seq2seq models, Even though performance dropped slightly after 12 steps, they still had stable performance across the horizon. GRU displayed a similar performance compared to LSTM-seq2seq, which both exhibited stable performance with slight error rise



(a) East

(b) West

(c) North

(d) South

(e) Central

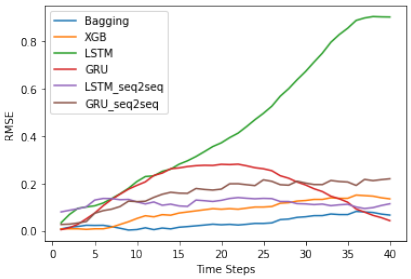
Figure 12: Comparison of 40-timesteps (10 hours, 1-timestep=15 minutes) forecasting horizon of models at group level in five regions (10 to 12 parking lots per region)

between 10 to 20-time steps. **(4) East area**, three models displayed the highest performance: Bagging regressor, LSTM-seq2seq, and GRU-seq2seq. Therefore, the hybrid strategy demonstrated superiority over the recursive strategy in general. LSTM and XGBoost showed higher errors than the three best-performing models. Also, GRU displayed fluctuation of errors with two peak points, 5 and 15, respectively. **(5) South area**, The ensemble models exhibited the highest performance, which displayed stable performance across the whole horizon. While seq2seq models underperformed ensemble models, they performed better than LSTM and GRU.

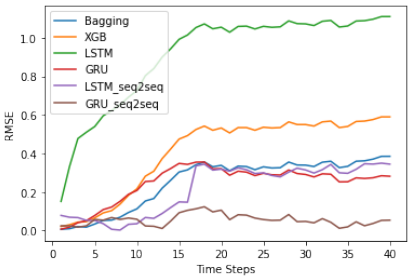
4.3.3 *Five Regions (Individual Level)*

From Figure 13 following result was obtained. **(1) Central area**, LSTM exhibited the best performance at an individual level showing stable performance until the entire horizon. GRU and GRU-seq2seq exhibited similar performance across the time horizon, followed by the Bagging regressor. Also, unlike group-level analysis, LSTM-seq2seq displayed the worst performance at the individual level across the time horizon. **(2) North area**, all models displayed high performance until 10 step horizon, then increased errors. However, LSTM-seq2seq and GRU showed the lowest position in the graph, which means better performance than others. The ensemble models outperformed GRU-seq2seq and LSTM, whose lines were lower than competitors. **(3) West area**, GRU-seq2seq models achieved the highest performance across the entire horizon. LSTM-seq2seq displayed stable performance until 15-time steps, then dropped performance. Bagging regressor and GRU exhibited similar performance across the entire horizon. In general, the hybrid approach showed better performance than the recursive strategy. **(4) East area**, The bagging regressor demonstrated the highest performance across the entire time horizon, while XGBoost outperformed LSTM-seq2seq until 28-time steps, then it started to underperform it. In addition, LSTM displayed a stiff increase of errors throughout the entire horizon, while GRU showed an upward convex shape. **(5) South area**, three models accurately predicted across the whole time horizon. Nevertheless, GRU showed the best performance, followed by Bagging regressor, LSTM, and LSTM-seq2seq, respectively. On the other hand, XGBoost and GRU-seq2seq demonstrated stiff performance drop after 10 step horizon.

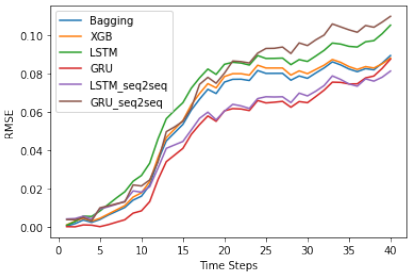
Comparison (Group Level vs. Individual Level)
(1) Central area, GRU-seq2seq exhibited the highest performance at group level and second at the individual level. **(2) North area**, the LSTM-seq2seq was the second-best model at the group level. However, it displayed the highest performance at the individual level. In general, all models' performances were stable until 10 step horizon. **(3) West area**, GRU-seq2seq displayed the highest performance, followed by LSTM-seq2seq at the group and individual levels. **(4) East area**, at the group level, bagging regressor and seq2seq models exhibited the best performance across the horizon. Nevertheless, while the bagging regressor showed the highest performance at the individual level, seq2seq models exhibited 3rd and fourth performance. **(5) South area**, Note that GRU at the group level exhibited the worst performance, but it was the best model at the individual level. The bagging regressor at the group level demonstrated the highest performance, but it was the third performing model at the individual level. also see Table 6 in Appendix (page 34)



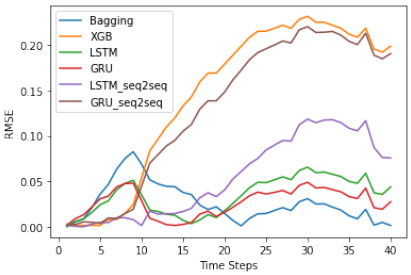
(a) East (lot number = 437)



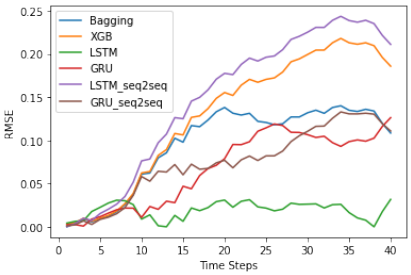
(b) West (lot number = 22)



(c) North (lot number = 547)



(d) South (lot number = 514)



(e) Central (lot number = 41)

Figure 13: Comparison of 40-timesteps (10 hours, 1-timestep=15 minutes) forecasting horizon of models at an individual level in five regions

Based on the analysis, identified findings are the following: (i) when the sample size became smaller scale, the performance of models dropped. (ii) As seen in section4.2, there was a performance drop in the east and west area against other regions. (iii) there was the superiority of models per sample size and region. for the whole sample level, hybrid strategy models exhibited superiority over recursive strategy models in general. At the group and individual levels analysis, while bagging regressor exhibited the highest performance in the east and south area, seq2seq models displayed superiority over other areas. (central, west, north).

5 DISCUSSION

In the previous section, the study tested the effect of features and the appropriate window size and performance on the different time horizons and regional differences. In general, multivariate models have demonstrated better performance. Moreover, window size 3 (using 16 previous time steps(4 hours) to predict the single time step ahead) would be suitable. Lastly, testing the performance of models for a 40-time horizon revealed that seq2seq models exhibited superior performance over other models using the whole sample. At the group and individual level analysis, specific models were superior in different regions.

5.1 *Effect of features*

Our result on feature revealed that, in general, multivariate models outperformed univariate models. This result strengthens the argument of Alajali et al. (2017) that data integrated with multiple features increases the model's performance. On the other hand, while Alajali et al. (2017) noted boosting method (Gradient boosting regression tree) demonstrated the increase in performance using multivariate data, XGBoost in our study was not influenced by multivariate features. In addition to temporal (`day_of_week`, `hour_of_day`) and facility features (`total_lot`, `gantry_height`, `car_park_type=MULTI-STOREY`, `car_park_decks`, `short_term_parking=WHOLE DAY`, `free_parking=NO`, `free_parking=SUN&PH7AM-10.30PM`, `carpark_number`), D2 also included spatial features (`x_coord`, `y_coord`). Unlike the study(Yu et al., 2017) indicated the limitation of ML/RNN-based models, which cannot take spatial correlation into account, the Spatial feature in this study explained the models somehow.

5.2 *Performance on Different Time Windows*

At the 855 parking lots analysis, the Bagging regressor displayed a performance drop when increasing window size, while others not exhibited a stiff change in performance across window size. It implies that the appropriate time window would be window 1, which result was in line with autocorrelation analysis (855 parking lots, lag=1).

At the group level analysis, while ensemble models less affected by window size, RNN models displayed a stiff performance increase when increasing the window size. It could be assumed that RNN models are more sensitive to capturing temporal dependency than ML models. Except for XGBoost, all models increased performance at the individual level when

increasing the window size. Furthermore, it strengthens our assumption during the EDA autocorrelation analysis (individual parking lots, lag=16).

From the experiment, the study identified several trends. (i) When using the whole sample, models' performance did not vary across window sizes. Nevertheless, when sample size decreases smaller scale, the performance of models varies based on different window sizes and regions. (ii) Regarding the comparison of performance of models at group and individual level in five regions, East and West areas showed a drastic change of performance across time windows, while central, south, and north areas were not much affected by time windows. It implies that the west and east areas have more non-stationary patterns than other regions, making prediction more problematic. (this topic will be discussed further in section 5.3)

In that sense, the study determined the appropriate window size would be window 3 (use 16 previous timesteps to predict the next timestep ahead). The reason is the robustness of the model's performance at a different sample size. When window 1, the performance of models varied when decreasing size of samples. In contrast, the performance of models was stable across the different sizes of samples in window 3. In reality, People might check the whole sample level, group level, or individual level in search of a parking lot. thus, models trained with window 3 would produce consistent and robust results. In addition, Geographic location is one characteristics influencing parking need (Litman, 2016). Based on Stathopoulos and Karlaftis (2003) among five parking lots nearby located, time lags of 4 other locations correlated to the one location. Also, Rajabioun and Ioannou (2015) demonstrated that correlation between parking location's utility which correlation dropped when increasing distance. Therefore, group-level analysis accounted for spatial dependency and would be more critical than other levels.

5.3 *Performance on Different Time Horizon*

Based on regional analysis at both group and individual levels, the study identified a performance drop in the east and west area against other regions. There was a regional difference between east and west against other areas, as east and west exhibited higher error scores. The hypothesis of higher prediction error in the east and west area would be that those regions have characteristics such as longer travel time to the central area and less complexity of public transport network in comparison to other areas. Buehler (2011) revealed that individuals exhibited less usage of the private car when individuals reside in an area where easy public transport accessibility, higher population density which close to commercial area. For that reason, the study could assume people in the east and west uses

more private cars than public transport.

Also, there was a superiority of models per size of sample and different regions. For the whole sample level, the bagging regressor exhibited stable performance across the whole time horizon, but in general, the hybrid strategy was superior to the recursive strategy. Even though the bagging regressor used a recursive strategy, bagging produced robust prediction result in multistep forecasting compared to other recursive strategy models, as they displayed a drastic drop in performance after 5 step horizon. One possible reason of superiority of seq2seq models could be that multioutput sequence generation preserves temporal dependency well in the short term. Another possible reason could be that the number of iterations to produce the entire horizon was much less(10 iterations) than the pure recursive strategy(40 iterations). When comparing group and individual level analysis, while bagging regressor demonstrated the highest performance in the east and south area, seq2seq models had superiority over other areas. (central, west, north). Based on the three best-performing models in each region(Table 6), the study proposes GRU-seq2seq in the central and west areas, LSTM-seq2seq in the north area, bagging regressor in east and south area as proposed models produced robust and consistent result against different sizes of sample.

5.4 *Scientific and Societal Impact*

This study would contributes novelty in the parking prediction fields from several points of view. First, it is one of the earlier studies used an unexplored dataset in parking prediction fields. Since dataset API has been open to the public since March 2022, studies have yet to be found using dataset in this study. Second, in comparison of models, the study identified how performance varies in different sizes of sample. Third, the study found the superiority of performance in different regions. Based on the finding, the study proposes optimal usage of models per region. with this findings, optimal models per region would be applicable to further developing parking prediction systems. Therefore, result of the study could contribute better quality of life for the ordinary people in Singapore.

5.5 *Limitation and Future Direction*

First, in the data collection stage, there was a lack of equal length of time points per parking lot. Even though the time per different parking lot is independent, there could be a limitation in terms of more data from 1 parking lot in the model in comparison to a parking lot that has shorter time points. It might affected the performance of models per individual parking

lot. In future studies, the researcher need to effectively consider and treat those missing time points. Second, as explored literature reported the effect of other features such as pedestrian volumne(Alajali et al., 2017) and weather condition(Yang et al., 2022; Zeng et al., 2022), but not considered in this experiments. Also, the nature of the dataset in this study is off-street. Therefore, on-street and traffic, as well as commercial parking data, were not considered. Third, none of the candidate models in this study were designed to capture spatial dependency. Even if some spatial features were encoded to models somehow, the model was not captured that feature systematically. Therefore, further research is needed by exploring a spatial-temporal graph convolutional network to encode spatial correlation near parking lots. Also, attention mechanisms need to be considered as the performance of a model in case of long-term prediction(Yang et al., 2022). Fourth, the study only conducted short-term prediction, which only 40 timesteps (10 hours). further study needs to check how the performance will differ long-term periods, which is out of our 40-time horizon. Lastly, Due to the trade-off computational cost and model accuracy, the study had limitation that hyperparameter tuning was not systematically done, especially for seq2seq models. Further research needs to consider a detailed hyperparameter tuning to find the optimal configuration.

6 CONCLUSION

The study aimed to find optimal forecasting performance, forming the main research question.

> *MQ. To what extent does the combination of spatial, temporal, and facility-related information affect the performance of ensemble and deep learning models in parking prediction?*

That was achieved by answering four sub-questions. The answer to **RQ1: To what extent do individual features of the parking lot affect the model's performance?** is that three types of features influence the performance of models by comparing univariate datasets. Especially bagging regressor and GRU exhibited a performance increase. The answer to **RQ2: To what extent does models' performance vary when using a different time window?**, the study revealed that the performance of models varies based on window size as well as the size of the sample. In general, using the whole sample, the appropriate window size was window 1. On the other hand, at the group and individual level of the sample size, window size 3 was suitable. The study propose window 3 due to the robustness of prediction performance on different sample sizes and a spatial dependency accounted for at group levels. The answer to **RQ3: How does the performance of models vary**

in different prediction horizons? is answered by conducting a multistep analysis using two strategies with six models. The study revealed the superiority of seq2seq models with a hybrid recursive-multioutput strategy over a pure recursive strategy using the whole sample. Lastly, to answer **RQ4: How does the performance of models vary in different geographical locations?**, The study revealed a regional discrepancy of the performance of models. The study propose the bagging regressor in the east and south area while seq2seq models in the other three areas (west, central, and north). The result opened the possibility of multiple model usage in parking prediction applications per region.

7 DATA SOURCE/CODE/ETHICS STATEMENT

Work on this thesis did not involve collecting data from human participants or animals. Real-time API was obtained from `https://data.gov.sg/dataset/carpark-availability`. Also, the facility dataset was obtained from `https://data.gov.sg/dataset/hdb-carpark-information`. The code used in this thesis is not publicly available. The author produced all images used in this thesis. Thus, copyright holds under the author.

REFERENCES

Alajali, W., Wen, S., & Zhou, W. (2017). On-street car parking prediction in smart city: a multi-source data analysis in sensor-cloud environment. In *International conference on security, privacy and anonymity in computation, communication and storage* (pp. 641–652).

Bilal, M., Persson, C., Ramparany, F., Picard, G., & Boissier, O. (2012). Multi-agent based governance model for machine-to-machine networks in a smart parking management system. In *2012 ieee international conference on communications (icc)* (pp. 6468–6472).

Buehler, R. (2011). Determinants of transport mode choice: a comparison of germany and the usa. *Journal of transport geography*, *19*(4), 644–657.

Camero, A., Toutouh, J., Stolfi, D. H., & Alba, E. (2019). Evolutionary deep learning for car park occupancy prediction in smart cities. In *International conference on learning and intelligent optimization* (pp. 386–401).

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).

Cho, K., Van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Garg, S., Lohumi, P., & Agrawal, S. (2020). Smart parking system to predict occupancy rates using machine learning. In *International conference on information, communication and computing technology* (pp. 163–171).

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Housing, & (HDB), D. B. (2020). *key statistics hdb annual report 2019/2020*.

Ji, Y.-j., Tang, D.-n., Guo, W.-h., Blythe, P. T., & Wang, W. (2014). Forecasting available parking space with largest lyapunov exponents method. *Journal of Central South University*, *21*(4), 1624–1632.

Kao, I.-F., Zhou, Y., Chang, L.-C., & Chang, F.-J. (2020). Exploring a long short-term memory based encoder-decoder framework for multi-step-ahead flood forecasting. *Journal of Hydrology*, *583*, 124631.

Kirby, H. R., Watson, S. M., & Dougherty, M. S. (1997). Should we use neural networks or statistical models for short-term motorway traffic forecasting? *International journal of forecasting*, *13*(1), 43–50.

Litman, T. (2016). *Parking management: strategies, evaluation and planning*. Victoria Transport Policy Institute Victoria, BC, Canada.

Mei, Z., Zhang, W., Zhang, L., & Wang, D. (2020). Real-time multistep prediction of public parking spaces based on fourier transform–least squares support vector regression. *Journal of Intelligent Transportation Systems*, *24*(1), 68–80.

Rajabioun, T., & Ioannou, P. A. (2015). On-street and off-street parking availability prediction using multivariate spatiotemporal models. *IEEE Transactions on Intelligent Transportation Systems*, *16*(5), 2913–2924.

Sadia, K., Reza, R., Alam, A., & Rahman, M. A. (2021). Car parking availability prediction: A comparative study of lstm and random forest regression approaches. *International Journal of Automation, Artificial Intelligence and Machine Learning*, *2*(1), 16–29.

Shahhosseini, M., Hu, G., & Pham, H. (2022). Optimizing ensemble weights and hyperparameters of machine learning models for regression problems. *Machine Learning with Applications*, *7*, 100251.

Stathopoulos, A., & Karlaftis, M. G. (2003). A multivariate state space approach for urban traffic flow modeling and prediction. *Transportation Research Part C: Emerging Technologies*, *11*(2), 121–135.

Stolfi, D. H., Alba, E., & Yao, X. (2017). Predicting car park occupancy rates in smart cities. In *International conference on smart cities* (pp. 107–117).

Taieb, S. B., & Atiya, A. F. (2015). A bias and variance analysis for multistep-ahead time series forecasting. *IEEE transactions on neural networks and learning systems*, *27*(1), 62–76.

Tekouabou, S. C. K., Cherif, W., Silkan, H., et al. (2020). Improving parking availability prediction in smart cities with iot and ensemble-based model. *Journal of King Saud University-Computer and Information*

Sciences.

Van Der Voort, M., Dougherty, M., & Watson, S. (1996). Combining kohonen maps with arima time series models to forecast traffic flow. *Transportation Research Part C: Emerging Technologies*, *4*(5), 307–318.

Vlahogianni, E. I., Kepaptsoglou, K., Tsetsos, V., & Karlaftis, M. G. (2016). A real-time parking prediction system for smart cities. *Journal of Intelligent Transportation Systems*, *20*(2), 192–204.

Xiao, X., Jin, Z., Hui, Y., Xu, Y., & Shao, W. (2021). Hybrid spatial–temporal graph convolutional networks for on-street parking availability prediction. *Remote Sensing*, *13*(16), 3338.

Yang, H., Ke, R., Cui, Z., Wang, Y., & Murthy, K. (2022). Toward a real-time smart parking data management and prediction (spdmp) system by attributes representation learning. *International Journal of Intelligent Systems*, *37*(8), 4437–4470.

Yu, B., Yin, H., & Zhu, Z. (2017). Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*.

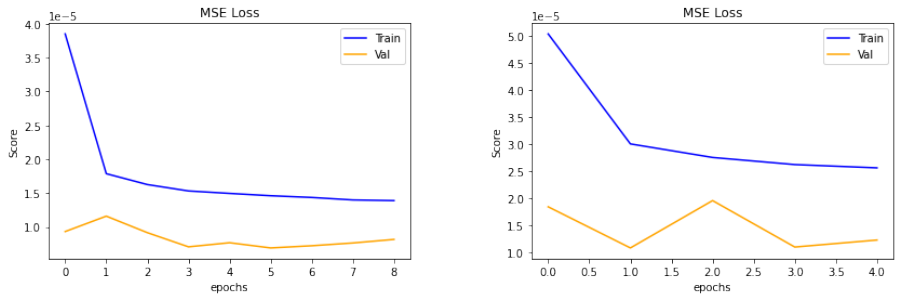
Zeng, C., Ma, C., Wang, K., & Cui, Z. (2022). Parking occupancy prediction method based on multi factors and stacked gru-lstm. *IEEE Access*, *10*, 47361–47370.

Zhang, Y., Li, Y., & Zhang, G. (2020). Short-term wind power forecasting approach based on seq2seq model using nwp data. *Energy*, *213*, 118371.

APPENDIX

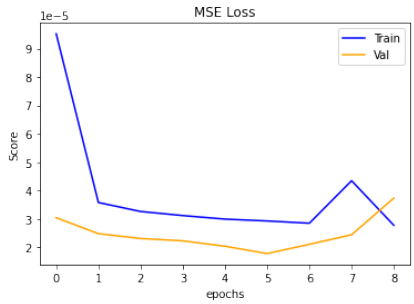
Table 6: The rank of three best models per region. Bold colors are proposed models per region (group level, individual level)

	Rank	Central	North	East	West	South
Group	1st	**GRU-seq**	GRU-seq	**Bagging**	**GRU-seq**	**Bagging**
	2nd	Bagging	**LSTM-seq**	LSTM-seq	LSTM-seq	XGBoost
	3rd	LSTM-seq	Bagging	GRU-seq	GRU	GRU-seq
Individual	1st	LSTM	**LSTM-seq**	**Bagging**	**GRU-seq**	GRU
	2nd	**GRU-seq**	GRU	XGBoost	LSTM-seq	LSTM
	3rd	GRU	Bagging	LSTM-seq	Bagging	**Bagging**

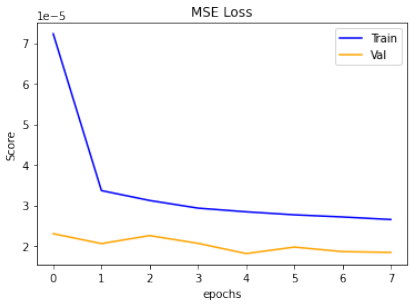


(a) LSTM

(b) GRU



(c) LSTM-seq2seq



(d) GRU-seq2seq

Figure 14: Comparison of train-test MSE of deep-learning models