Predicting Vacant Parking Space Availability: A Long Short-Term Memory Approach

Junkai Fan, Qian Hu, Yingying Xu, and Zhenzhou Tang*

Are with the College of Computer Science & Artificial Intelligence and the Innovation Center for Intelligent Networking, Wenzhou University, China. Email: 435589127@qq.com; huqian@wzu.edu.cn; yyxu@wzu.edu.cn; and mr.tangzz@gmail.com

xxxxxxx

***Abstract*—**The accurate prediction of vacant parking space availability is becoming increasingly essential for assisting drivers to determine where to park in advance. It helps ease traffic pressure and reduce gas emission and pollution. This article proposes a novel multistep long short-term memory recurrent neural network (LSTM-NN) model to predict the number of the vacant parking spaces on the basis of historical parking availability information. The key parameters of the model are deeply optimized. The prediction model is fully benchmarked with five well-known machine learning models—i.e., a gated recurrent units neural network, a stacked autoencoder, a support vector regression, a back propagation neural network, and a k-nearest neighbor algorithm—whose key parameters are sufficiently optimized as well. Adequate experiments with practical data collected from two parking lots with various capacities and traffic flows were conducted to evaluate the models' performances on short- and long-term predictions. Experimental results show that the proposed multistep LSTM-NN model outperforms all the benchmark models, especially in a commercial parking lot with heavy traffic flow.

Digital Object Identifier 10.1109/MITS.2020.3014131
Date of current version: 1 October 2020

*Corresponding author

1939-1390/20©2020IEEE

With the amazing increase of vehicles, the parking problem has become increasingly serious, especially in crowded urban areas. Traffic congestion caused by cruising for parking spaces wastes resources and aggravates environmental issues. Giuffrè et al. [1] reported that cruising for parking spaces results in a peak increase of about 25–40% in the traffic flow. Research by Shoup et al. [2] has also shown that 8–74% of the traffic is due to cruising for parking spaces, and the average time spent finding a parking space is 3.5–14 min, which has seriously effects on the traffic network. In another study, Ayala et al. [3] found that cruising for parking spaces wastes 8.37 million gallons of gasoline and produces more than 129,000 tons of CO_2 emissions every year. Thus, finding parking spaces quickly and efficiently is of great significance for relieving traffic pressure, saving energy, and reducing emissions.

Owing to the rapid development of Internet of Things technologies [4], networked control systems [5] and [6], and the promotion of smart cities, intelligent parking assistance is becoming a hot topic in both academia and industry. There have been plenty of intelligent parking assistance applications on smartphones that provide information on the availability of vacant parking spaces. These applications basically collect the number of vacant parking spaces with the aid of sensor devices and camera equipment [7]–[9] and display the real-time number of vacant parking spaces of a parking lot.

However, real-time information sometimes causes confusion. Imagine this scenario: A driver learns that a parking lot a 10-min drive away has five vacant parking spaces. It is natural for him to wonder whether there will still be any vacant parking spaces left when he arrives there after 10 min. Given this confusion, an accurate forecast is much more significant than real-time information. Therefore, collecting real-time parking information by sensing equipment and then predicting the number of vacant parking spaces based on these data are believed to be a promising way to facilitate parking. In particular, long-term predictions, such as predictions over a period of tens of minutes, are more desirable since a 10-min drive is quite common in urban trips.

Collecting real-time parking information and then predicting the future vacant parking space availability is actually a typical problem of time-series prediction. Some studies have already attempted to address this problem by using classical probability and statistics models and machine learning models, such as Markov models, regression trees, neural networks, and support vector regression, and so on [10]–[16]. Artificial neural networks, especially deep learning (DL) models, have shown a potential advantage regarding effectiveness. With cascaded multiple-layer architectures, namely deep architectures, and a sufficient amount of data, DL, which is vaguely inspired by information processing and communication patterns in biological nervous systems, is capable of extracting inherent features and making data-driven predictions or decisions. In recent years, besides time-series prediction, DL has also been successfully applied to natural language processing, image recognition, system optimization, automatic control, and so on [17]–[22].

However, few of the existing studies have focused on long-term predictions on parking availability by leveraging DL approaches. Motivated by this, the purpose of this study was to propose a well-optimized DL model to make precise long-term predictions on vacant parking space availability. Specifically, considering the inherent advantage of making long-term predictions, the long short-term memory recurrent neural network (LSTM-NN) model [23] was leveraged among a number of different DL prototypes in this study. To further improve the performance, the essential parameters of the LSTM-NN model were fully optimized, including the number of inputs, the number of hidden layers, the gradient descent optimizer, and the multistep prediction methods.

To validate the superiority of the proposed LSTM-NN model, in this article, we develop five other mainstream machine learning models, including a gated recurrent units neural network (GRU-NN) [24] model, a stacked autoencoder (SAE) [25] model, a support vector regression (SVR) [26] model, a back propagation neural network (BPNN) model, and a k-nearest neighbor (KNN) [27] model. We provide a comprehensive cross-model performance comparison among these models and take into account both their accuracy and computational complexity. For the sake of fairness, these benchmark models are also completely optimized. Moreover, in our study, two different parking lots—a private medium-capacity parking lot and a public large-capacity one—were taken as the experimental subjects so that the effectiveness of the prediction models could be examined in various scenarios.

The main contributions of this study can be summarized as follows:

- *Long-term prediction*: Unlike previous studies that focused only on short-term predictions, this study fully considers long-term predictions by leveraging the LSTM-NN model. In addition, two different multistep prediction methods, namely the iterative multistep prediction and the direct multistep prediction, are proposed and integrated into the LSTM-NN model to further improve the accuracy of the long-term prediction.
- *Full optimization*: The key parameters of the LSTM-NN model—including the number of inputs, the number of hidden layers, the gradient descent optimizer, and the multistep prediction methods—are optimized by the grid search method. Moreover, all the benchmark models are also optimized in terms of key parameters and multistep prediction methods.

- *Comprehensive benchmarks*: The proposed prediction model is comprehensively benchmarked with five commonly used machine learning models. The data for the experiments were collected from two practical parking lots with different capacities and traffic volumes. The experimental results show that the iterative multistep LSTM-NN model outperforms all the benchmark models, especially in a commercial parking lot with heavy traffic flow. As far as we know, our comparative analysis is the most comprehensive of the existing work.

Related Work

Currently, the studies on vacant parking space availability prediction can be divided into two categories, i.e., prediction based on parking process analysis and prediction based on big data analysis and machine learning.

Prediction Based on Parking Process Analysis

In [10] and [28], the classical queuing theory was employed to model the vehicle arrival process and departure process, and the distribution of parking space occupancy was then predicted using an autoregressive Gaussian process. Peng and Li [12] modeled the discrete occupancy rates of the parking lots as nonstationary Poisson processes and proposed a cost-effective parking space search method by predictions with time-varying analysis of historical data. Xiao et al. [13] investigated a practical framework based on a continuous-time Markov M/M/C/C queueing model to predict the parking occupancy for various prediction intervals ranging from 2 to 12 min based on historical occupancy data. In [29], a real-time availability forecast algorithm was proposed to predict parking availability by allocating simulated parking requests, estimating future departures, and, finally, predicting parking availability. Bock and Sester [30] investigated the estimation of vacant parking space availability based on spatial methods using sensor data from San Francisco, using the interpolation method to estimate the parking availability of unobserved sections.

Because this type of existing work is highly dependent on assumptions of the arrival and departure processes, it can hardly be adapted to the dramatic fluctuations in traffic flow of parking lots. Further, most of the aforementioned studies mainly focused more on theoretical analysis and are hard to implement in practice.

Prediction Based on Big Data Analysis and Machine Learning

In [11], the parking occupancy rate was predicted by using three feature sets constructed of data collected by sensor networks. Three different machine learning methods—i.e., regression tree, BPNN, and SVR—have been leveraged to make predictions. Yu et al. [31] took advantage of the autoregressive integrated moving average model to predict the number of the vacant parking spaces. An SVR-based two-step approach has also been proposed to predict parking space availability [32]. The first step involves processing the raw data into a stationary state to tune the parameters of the SVR model. In the second step, the parameters obtained in the first step are used to train a multidimensional SVR model that is suitable for parking predictions. A prediction model based on the wavelets neural network and a time-space dependent multivariate autoregressive model have been proposed to predict the short-term number of vacant parking spaces in [33] and [34], respectively. After analyzing various data on parking, weather, and traffic flow in an integrative manner, Badii et al. [15] proposed a Bayesian-regularized neural network model to forecast vacant parking space availability. In [35], a hybrid LSTM-NN prediction model was proposed to estimate the real-time parking availability level of parking lots without sensing equipment using the historical parking availability information of other parking lots that were equipped with sensing equipment in addition to using various data sets on some relative factors.

In addition, Yang et al. [16] proposed a DL model for free parking space predictions by combining a graph-convolutional neural network and an LSTM-NN model to capture the spatial relationships of the traffic flow and the temporal features of the parking data, respectively. Multiple data sources, including parking meter transactions, traffic conditions, and weather conditions have been imported into the model. Experiments show that an average mean absolute percent error (MAPE) of 10.6% can be achieved when predicting block-level parking occupancies 30 min in advance. Wu et al. [36] developed a parking recommendation sequence considering both the time to reach the parking lot and predicted successful parking probability. In our previous work [14], we also studied an SVR model to predict the availability of vacant parking spaces where the essential parameters of SVR were optimized with a fly optimization algorithm.

However, few of the aforementioned studies have focused on the long-term prediction of parking availability using DL approaches. Most have either estimated the real-time parking availability for parking lots without sensor devices or made short-term predictions, i.e., no longer than half an hour. Moreover, these studies have not provided comprehensive cross-model performance comparisons among various mainstream machine learning models.

The prediction model proposed in the present study belongs to the second approach, where the LSTM-NN model, which is noted for its remarkable capability of long-term predictions, is leveraged. More importantly, to predict as accurately as possible, the present study also involves an overall optimization upon the LSTM-NN model, including the essential parameters, the multistep approach, and the optimizer. In addition, another advantage of the present study over the existing work is that our study provides a

comprehensive benchmark in which five other commonly used machine learning models are involved.

Methodology

As mentioned in the "Related Work" section, vacant parking space availability prediction methods based on parking process analysis depend heavily on some additional assumptions on arrival and departure processes and are mainly theoretical, whereas methods that make predictions directly from historical observed data by machine learning are much more straightforward and practical for implementation. Moreover, machine learning approaches are more qualified to deal with irregular traffic volume fluctuations since the predictions are typically made from recently collected information that has a strong time correlation with the predicted value.

With this in mind, in this study, we used machine learning, or rather a DL approach, to make predictions. In the next section, we first introduce in detail the fundamentals of the LSTM-NN model and the intuition of using it to make predictions. Then we propose two multistep prediction methods for long-term predictions. Finally, the detail training process of the prediction method is provided.

Recurrent Neural and LSTM Networks

A recurrent neural network (RNN) is a kind of memory neural network that is widely used to solve time-series prediction problems and machine translation problems. Similar to a normal neural network, an RNN has an input layer, a hidden layer, and an output layer. The difference is that there is a circular network of cyclic links between hidden units. This makes the final output dependent not only on the current input but also on the output of the previous hidden layer. In other words, the most significant characteristic of the RNN is that it can introduce previous hidden layer information to the current task.

The forward propagation structure of RNN is given in Figure 1, where $\mathbf{x}_t$ is the m-dimensional input vector in the training set at time step t; $\mathbf{y}_t$ is the corresponding label (the actual value) in the training set at time step t; $\mathbf{h}_t$ is the q-dimensional hidden state vector at time step t; $\mathbf{y}_t^*$ is the prediction vector at time step t; and L_t is the value of the loss function at time step t.

The mathematical model of RNN can be defined as follows:

$$\mathbf{h}_t = \sigma(\mathbf{W}\mathbf{h}_{t-1} + \mathbf{U}\mathbf{x}_t + \mathbf{b}_h), \tag{1}$$

$$\mathbf{y}_t^* = \sigma(\mathbf{V}\mathbf{h}_t + \mathbf{b}_{y^*}), \tag{2}$$

where $\mathbf{W} \in \mathbb{R}^{q\times q}$, $\mathbf{U} \in \mathbb{R}^{q\times m}$, and $\mathbf{V} \in \mathbb{R}^{1\times q}$ are the weight matrices; $\mathbf{b}_h \in \mathbb{R}^q$ is the hidden bias vector; $\mathbf{b}_{y^*}$ is the output bias vector; and σ is the activation function typically represented by the sigmoid function, i.e., $\sigma(x) = 1/1 + e^{-x}$.

However, the traditional RNN is vulnerable to the vanishing gradient problem when long-term dependencies are involved in learning tasks. In fact, the propagation of a neural network gets stuck because it is so difficult for the network to learn useful information from the former layers when the vanishing gradient problem occurs. That is, information decays through time. In view of this, Hochreiter et al. [23] introduced the long short-term memory unit to the RNN structure. An embedded new structure called a *memory cell* was introduced that enabled the LSTM-NN model to overcome the vanishing gradient problem so that it could efficiently learn long-term dependencies in the data and make long-term time-series predictions more accurately [18], [37]. Figure 2 presents the architecture of an LSTM memory unit that includes three gates, i.e., the forget gate, the input gate, and the output gate.

The Forget Gate

The purpose of the forget gate is to determine the extent to which a previous value remains in the memory unit. The output of the forget gate is as follows:

$$\mathbf{F}_t = \sigma(\mathbf{W}_{fx}\mathbf{x}_t + \mathbf{W}_{fh}\mathbf{h}_{t-1} + \mathbf{b}_f), \tag{3}$$

where $\mathbf{W}_{fx} \in \mathbb{R}^{q\times m}$ and $\mathbf{W}_{fh} \in \mathbb{R}^{q\times q}$ are the weight matrices; and $\mathbf{b}_f \in \mathbb{R}^q$ is the bias vector of the forget gate. From (3), it can be observed that $\mathbf{h}_{t-1}$ and $\mathbf{x}_t$ are combined and then enter the forget gate. With the sigmoid activate function, a weight range from 0 to 1 is imposed upon $\mathbf{C}_{t-1}$, which is the value of the memory unit of the previous time step. The closer the value is to 1, the greater the amount of information that should be retained.

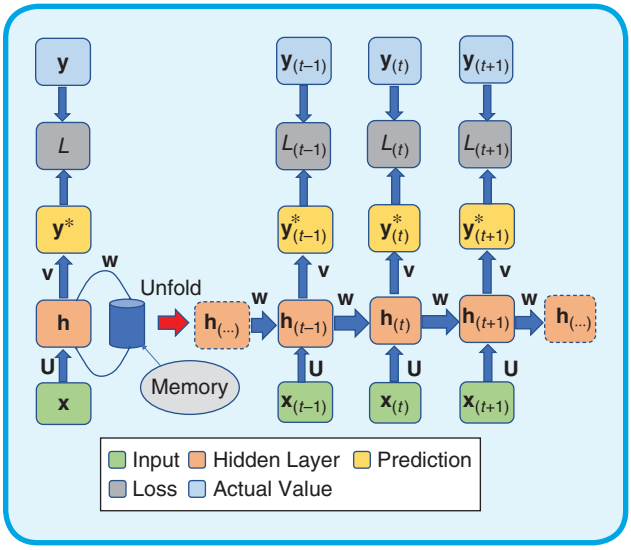


FIG 1 The forward propagation structure of RNN. $\mathbf{x}_t$ is the input, $\mathbf{h}_t$ is the hidden state matrix, $\mathbf{y}_t$ is the actual value, L_t is the loss function, and $\mathbf{y}_t^*$ is the prediction.

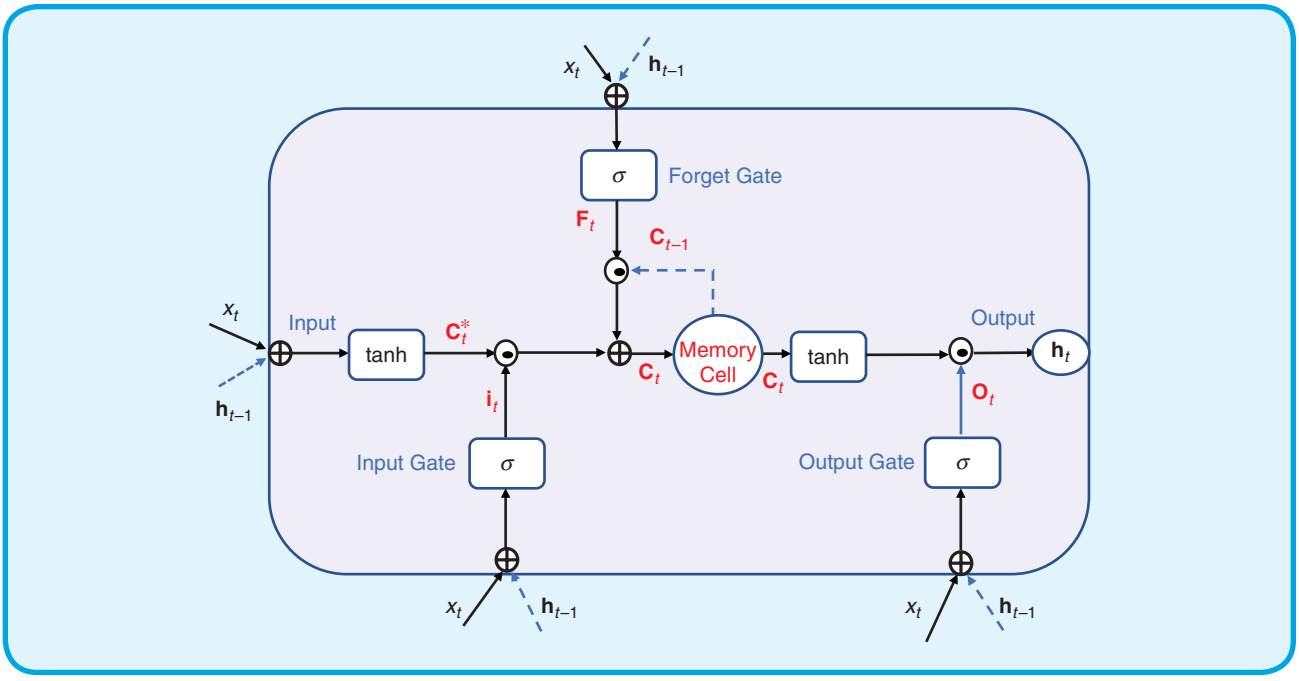


FIG 2 The architecture of the LSTM memory unit.

The Input Gate

The input gate determines the extent that new values enter the memory unit, as shown in (4).

$$\mathbf{i}_t = \sigma(\mathbf{W}_{ix}\mathbf{x}_t + \mathbf{W}_{ih}\mathbf{h}_{t-1} + \mathbf{b}_i), \tag{4}$$

where $\mathbf{W}_{ix} \in \mathbb{R}^{q\times m}$ and $\mathbf{W}_{ih} \in \mathbb{R}^{q\times q}$ are the weight matrices; and $\mathbf{b}_i \in \mathbb{R}^q$ is the bias vector of the input gate.

$\mathbf{h}_{t-1}$ and $\mathbf{x}_t$ are blocked by a tanh layer, which outputs a vector of new candidate values $\mathbf{C}_t^*$, as shown in

$$\mathbf{C}_t^* = \tanh[(\mathbf{W}_{cx}\mathbf{x}_t + \mathbf{W}_{ch}\mathbf{h}_{t-1}) + \mathbf{b}_c], \tag{5}$$

where $\mathbf{W}_{cx} \in \mathbb{R}^{q\times m}$ and $\mathbf{W}_{ch} \in \mathbb{R}^{q\times q}$ are the weight matrices; and $\mathbf{b}_c \in \mathbb{R}^q$ is the bias vector of the tanh layer.

Then, $\mathbf{i}_t$ and $\mathbf{C}_t^*$ are combined as follows:

$$\mathbf{C}_t = \mathbf{F}_t \odot \mathbf{C}_{t-1} + \mathbf{i}_t \odot \mathbf{C}_t^*, \tag{6}$$

where $\mathbf{F}_t \odot \mathbf{C}_{t-1}$ represents the retained information of the previous state; $\mathbf{i}_t \odot \mathbf{C}_t^*$ represents the newly introduced information; and the operator $\odot$ represents the element-wise matrix multiplication.

The Output Gate

The output gate controls the extent to which the value in the memory unit is used to compute the output activation of the LSTM architecture. The output of the output gate is

$$\mathbf{o}_t = \sigma(\mathbf{W}_{ox}\mathbf{x}_t + \mathbf{W}_{oh}\mathbf{h}_{t-1} + \mathbf{b}_o), \tag{7}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{C}_t), \tag{8}$$

where $\mathbf{W}_{ox} \in \mathbb{R}^{q\times m}$ and $\mathbf{W}_{oh} \in \mathbb{R}^{q\times q}$ are the weight matrices; and $\mathbf{b}_o \in \mathbb{R}^q$ is the bias vector of the output gate. Finally, by substituting (8) into (2), the prediction of the LSTM-NN model, i.e., $\mathbf{y}_t^*$, is obtained.

It should be noted that there is only one output in this work. Hence, instead of vectors, we use the scalars y_t and y_t^* to denote the actual value and the prediction.

Multistep Prediction Methods

There are two fundamental prediction approaches. One is the single-step prediction approach, and the other is the multistep prediction approach.

The Single-Step Prediction Approach

The single-step prediction approach predicts $y_{t_c+\delta}^*$ by using m historical observations $[x_{t_c-(m-1)\delta}, x_{t_c-(m-2)\delta}, \ldots, x_{t_c}]$, where t_c represents the current time point and δ is the time step size. For instance, in the case of $\delta = 5$ and $m = 5$, the single-step prediction predicts the vacant parking space availability at $t+5$ by the historical vacant parking spaces information at $[t-20, t-15, \ldots, t]$.

The Multistep Prediction Approach

Like the single-step prediction approach, the multistep prediction approach predicts $y_{t_c+h\delta}^*(h > 1)$ on the basis of m historical observations $[x_{t_c-(m-1)\delta}, x_{t_c-(m-2)\delta}, \ldots, x_{t_c}]$. Two different multistep prediction strategies are described in this article, i.e., the iterative and the direct multistep prediction strategies. In the iterative multistep prediction strategy, $y_{t_c+\delta}^*$ is first predicted with $[x_{t_c-(m-1)\delta}, x_{t_c-(m-2)\delta}, \ldots, x_{t_c}]$. Then, $y_{t_c+2\delta}^*$ is predicted by $[x_{t_c-(m-2)\delta}, x_{t_c-\delta}, \ldots, x_{t_c}, y_{t_c+\delta}^*]$ in

a iterative way. $y^*_{t_c+2\delta}$ is further used to predict $y^*_{t_c+3\delta}$, and so on. By contrast, the direct multistep prediction strategy outputs $y^*_{t_c+h\delta}$ directly with the historical observations $[x_{t_c-(m-1)\delta}, x_{t_c-(m-2)\delta}, \ldots, x_{t_c}]$, as given in Figure 3.

Multistep prediction methods impose nonnegligible effects upon the accuracy of predictions. The iterative multistep prediction strategy introduces cumulative errors during iterations, which may increase the prediction error. In the direct multistep prediction strategy, however, the time correlation between the historical information and the predicted value is so weak that it also leads to considerable prediction error. In view of this, in this study, the optimal multistep prediction strategy was determined by grid searching. The details are provided in the section "Model Optimizations."

The Training Process

As explained in the subsection "Multistep prediction methods" in this section, both of the two multistep prediction approaches are based on the single-step prediction approach. Therefore, in this section, we first introduce the training process of the single-step prediction approach of the LSTM-NN model, which can be described as follows:

- Step 1. Prepare data and initiate some essential parameters including the epochs, the number of units in the hidden layer, and the input historical data length m.
- Step 2. Preprocess the historical parking data.
  - Step 2.1. Normalize the data.
  - Step 2.2. Divide the normalized data into the training set and the test set.
  - Step 2.3. Reconstruct data according to the given m as follows:

$$\begin{bmatrix} x_{t_c-(m-1)\delta} & x_{t_c-(m-2)\delta} & \cdots & x_{t_c} & \overbrace{x_{t_c+\delta}}^{y} \\ x_{t_c-(m-2)\delta} & x_{t_c-(m-3)\delta} & \cdots & x_{t_c+\delta} & x_{t_c+2\delta} \\ x_{t_c-(m-3)\delta} & x_{t_c-(m-4)\delta} & \cdots & x_{t_c+2\delta} & x_{t_c+3\delta} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{t_c} & x_{t_c+\delta} & \cdots & x_{t_c+(m-1)\delta} & x_{t_c+m\delta} \end{bmatrix}.$$

- Step 3. Train the LSTM-NN model.
  - Step 3.1. First, randomly initialize the weight matrices and the bias vectors, including $\mathbf{W}_{fx}, \mathbf{W}_{fh}, \mathbf{W}_{ix}, \mathbf{W}_{ih}, \mathbf{W}_{cx}, \mathbf{W}_{ch}, \mathbf{W}_{ox}, \mathbf{W}_{oh}$ and $\mathbf{b}_f, \mathbf{b}_i, \mathbf{b}_c, \mathbf{b}_o$.
  - Step 3.2. Forward propagation: Calculate (y^*) by (3)–(8), then calculate the loss function which is defined as follows:

$$L(t) = \frac{1}{T}\sum_{t=1}^{T}(y_t - y_t^*)^2. \tag{9}$$

  - Step 3.3. Back propagation. Calculate the following gradients: $\partial L(t)/\partial W_{fx}, \partial L(t)/\partial W_{fh}, \partial L(t)/\partial W_{ix}, \partial L(t)/\partial W_{ih}, \partial L(t)/\partial W_{cx}, \partial L(t)/\partial W_{ch}, \partial L(t)/\partial W_{ox}, \partial L(t)/\partial W_{oh}$ and $\partial L(t)/\partial \mathbf{b}_f, \partial L(t)/\partial \mathbf{b}_i, \partial L(t)/\partial \mathbf{b}_c, \partial L(t)/\partial \mathbf{b}_o$. In this work, the back propagation is trained by a back propagation through time algorithm [23].
  - Step 3.4. Minimize $L(t)$ by a proper gradient descent optimization algorithm.

In the present study, two gradient descent optimization algorithms—namely, the root mean square propagation optimization (RMSProp) algorithm [38] and the adaptive moment estimation (ADAM) algorithm [39]—are considered to dynamically optimize the weights of the LSTM-NN model so as to minimize $L(t)$. The RMSProp algorithm is an adaptive learning rate method that maintains per-parameter learning rates that are adapted to the average of recent magnitudes of the gradients for the weight. The RMSProp algorithm has shown an excellent adaptation of learning rate in different applications and is capable of dealing with online and nonstationary issues. The ADAM algorithm is an update of the RMSProp algorithm that adds bias-correction and momentum to the RMSProp algorithm. The ADAM algorithm can handle sparse gradients on noisy problems and is computationally efficient as well.

Thus far, the ADAM algorithm might be a better choice. However, that does not mean it is the end-all and be-all. Actually, we have seen some studies that have used the RMSProp rather than the ADAM algorithm [40], [41]. In general, the choice of an optimizer is generally dependent on the model and the data set. The practical procedure is most likely to be to try a bunch of different optimizers and choose the one that provides the most consistent performance [41], [42]. With this in mind, the

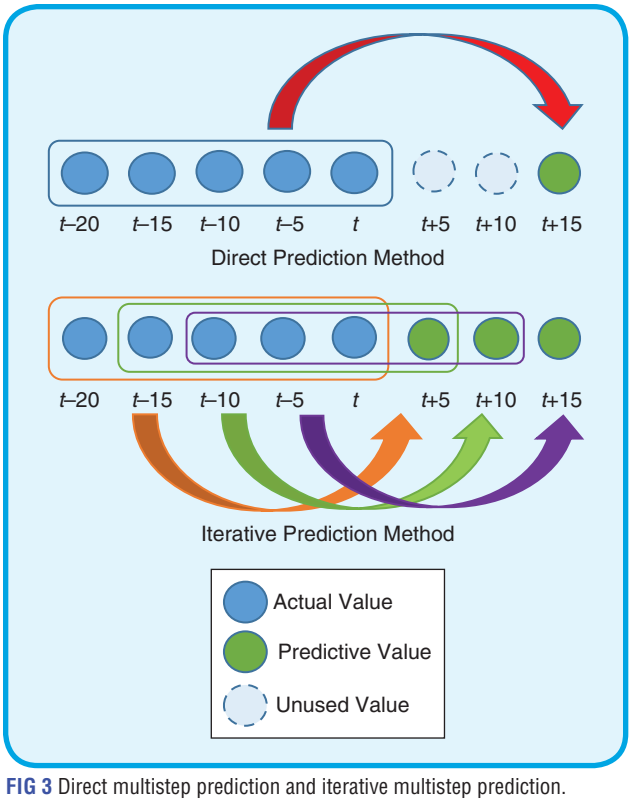


FIG 3 Direct multistep prediction and iterative multistep prediction.

training losses and the convergence rates of the two algorithms are compared in this article. The learning rates of the RMSProp and the ADAM algorithms are both 0.001, which are the default rates. The results are presented in Figure 4. The description of the data used in the comparisons is presented in detail in the next subsection, "Data Description." It can be observed that the ADAM algorithm slightly outperforms the RMSProp algorithm in both performance indicators. We, therefore, selected the ADAM algorithm as the optimizer in our proposed LSTM-NN model.

To achieve the best training results, the optima of the length of the input vectors m, the number of memory units of the hidden layer, and the multistep prediction method are obtained by using a grid search method, which is an exhaustive searching through a specified feasible value space. The details for model optimization are presented in the subsection "Model optimizations" in the section "Model Optimizations."

Data Description

The data used in the present study were collected from two parking lots. One set of data was from the parking lot located at (28.006187, 120.684459) in Wenzhou University, China, denoted as P-WZU, from 7 a.m., 20 March 2018 to 9:50 p.m., 20 April 2018, as presented in Figure 5(a). The time stamps of each vehicle entering and leaving the parking lot were recorded. The other set of data was collected at the parking lot located at (34.01444, –118.49372), Santa Monica, California [43], denoted as P-SM, from 7 a.m., 11 May 2018 to 9:50 p.m., 11 June 2018, as given in Figure 5(b). In this data set, the number of vacant parking spaces was collected every 5 min.

For the data collected in the P-WZU parking lot, the following preprocessing was required to obtain the number of vacant parking spaces every 5 min:

$$f(t+5) = f(t) + \{F_i[t\sim(t+5)] - F_o[t\sim(t+5)]\}, \qquad (10)$$

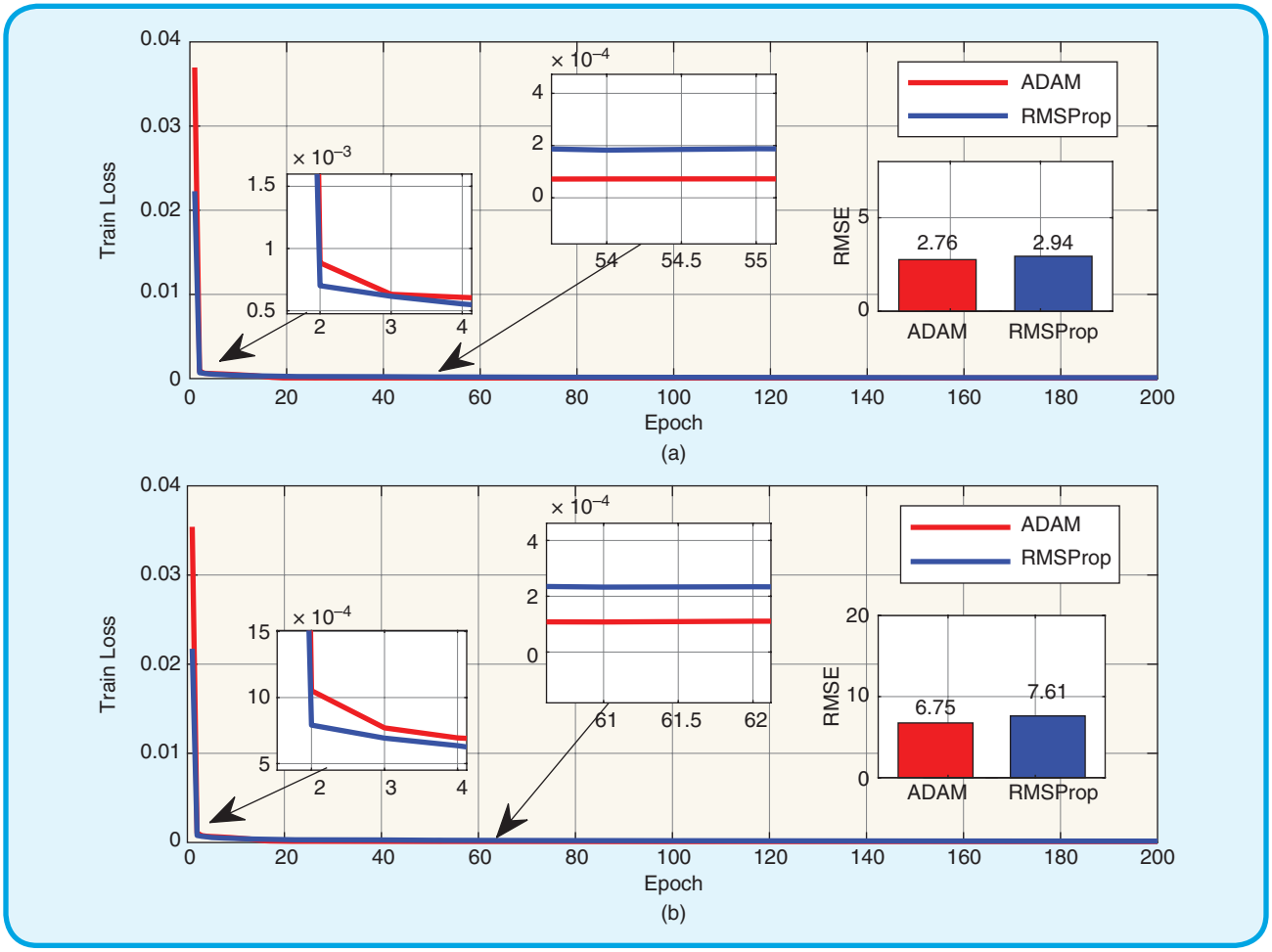


FIG 4 The RMSE of the training loss and the convergence rate of the 5-min prediction. (a) The P-WZU parking lot. (b) The P-SM parking lot. The length of the input historical data $m = 5$, the number of units in the hidden layer $N_h = 100$, the batch_size (i.e., the frequency with which samples are read during an iteration) is 32, and the activation function of the output layer is linear. The data that are included in the training set is 67%, and the remaining 33% are in the test set.

where $f(t)$ denotes the number of vacant parking spaces in the parking lot at time t; and $F_i\{t\sim(t+5)\}$ and $F_o\{t\sim(t+5)\}$ represent the accumulative number of vehicles respectively entering and leaving the parking lot in the time period from t to $t+5$.

The P-WZU lot is a private parking lot and the traffic volume fluctuates mildly and regularly. By contrast, as a large public commercial parking lot, the parking space availability of the P-SM parking lot fluctuates much more significantly, as illustrated in Figure 5.

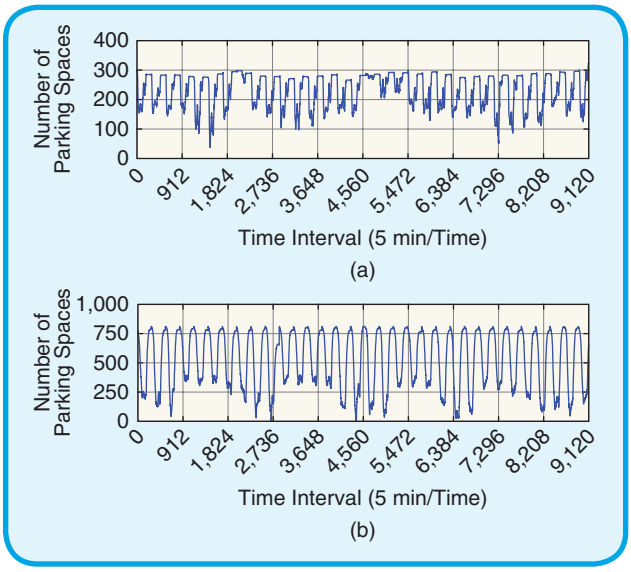


FIG 5 (a) The data of the P-WZU parking lot. (b) The data of the P-SM parking lot.

Model optimizations

Performance Indicators

In this article, three indicators—the mean absolute error (MAE), the mean absolute percent error (MAPE), and the root mean squared error (RMSE)—are used to measure the fitness between the predictions and the actual values. The MAE, MAPE, and RMSE are defined as follows:

$$\text{MAE} = \frac{1}{T}\sum_{t=1}^{T}|y_t - y_t^*|, \tag{11}$$

$$\text{MAPE} = \frac{100\%}{T}\sum_{t=1}^{T}\frac{|y_t - y_t^*|}{y_t}, \tag{12}$$

$$\text{RMSE} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(y_t - y_t^*)^2}, \tag{13}$$

respectively, where y_t^* is the predicted value at time step t; y_t is the actual value at time step t; and T is the number of time steps.

Model Optimizations

In this study, all the involved prediction models, including the proposed LSTM-NN model and the five benchmark models, were optimized mainly by the grid search approach guided by the MSE since the MSE is used as the loss function during the training process. Throughout the grid search process, 67% of the data were used as the training set and the rest were used as the test set. The vacant parking space availabilities after 5, 15, 30, 45, and 60 min were then respectively forecast.

The LSTM-NN Model

For the LSTM-NN model, the following parameters were optimally determined by a grid search:

- the length of the input vectors m chosen from the space $\{1,2,\ldots,12\}$
- the number of the hidden units in the hidden layer chosen from the space $\{10,20,\ldots,100\}$.

The grid search process was applied both to the direct multistep prediction and the iterative multistep prediction models. For each search, an average result of 100 experiments was considered. In addition, the training processes was performed 200 epochs with a batch_size of 32. The activation function of the output layer was linear. The optimal parameters of the LSTM-NN model for various prediction tasks are given in Table 1. Table 1 also lists the values of the other two indicators, namely MAE and MAPE, with the objective of presenting the performances from various perspectives.

It can be observed that for the P-WZU parking lot, the optimal parameters—denoted as (m, N_h, single-step/direct multistep/iterative multistep)—are [(8,100, single-step), (9, 30, direct multistep), (10, 30, direct multistep), (12, 20, iterative multistep), and (12, 70, iterative multistep)] for 5-, 15-, 30-, 45-, and 60-min predictions, respectively. Meanwhile, for the P-SM parking lot, (11, 30, single-step) was optimal for 5-min single-step predictions and (11, 30, iterative multistep) was the best decision for all the multistep predictions.

The Benchmark Models

The benchmark models were also optimized by grid search. The detail parameters for optimization are listed in Table 2. In addition, the other parameters for the benchmark models are listed as follows:

- GRU-NN: The batch_size was 32, the activation function of the output layer was linear, and the number of epochs was 200.
- SAE: There were three hidden layers. The batch_size was 32, the activation function of the output layer was linear, and the number of epochs was 200.
- SVR: The kernel function was linear.
- BPNN: The number of epochs was 1,000. The learning rate was 0.01. The activation function was the purelin function.

Similarly, a grid search was applied to both multistep predictions for all the benchmark models and the best pa-

Table 1. The results of the grid search for the LSTM-NN model.

	P-WZU Parking Lot						P-SM Parking Lot					
		Best Values						Best Values				
Task	Multistep	m	N_h	MAE	MAPE(%)	RMSE	Multistep	m	N_h	MAE	MAPE(%)	RMSE
5 min	Single-step	8	100	1.8998	1.04	2.9408	Single-step	11	30	3.9610	1.87	5.7730
15 min	Direct	9	30	4.0712	2.26	6.5230	Direct	11	70	8.0371	3.90	11.4901
	Iteration	12	70	4.1364	2.29	6.5443	Iteration	11	30	7.9560	3.74	11.3505
30 min	Direct	10	30	7.9760	4.22	11.6859	Direct	11	70	13.5717	6.41	19.2717
	Iteration	11	20	7.4160	4.04	11.7577	Iteration	11	30	13.4706	6.09	18.8193
45 min	Direct	9	80	11.8496	6.24	17.1734	Direct	11	10	19.3751	8.76	27.5592
	Iteration	12	20	11.0812	6.11	17.0129	Iteration	11	30	19.3825	8.24	27.1090
60 min	Direct	9	20	15.4423	8.34	22.6943	Direct	10	30	25.4131	12.30	36.8763
	Iteration	12	70	14.8386	8.26	22.4047	Iteration	11	30	25.7805	10.39	36.1273

Table 2. The parameters for optimization.

Model	Parameter	Search Range
GRU-NN	m	[1, 12]
	N_h	{10, 20, 30, 40, 50, 60, 70, 80, 90, 100}
SAE	m	[1, 12]
	N_h	{100, 200, 300, 400, 500}
SVR	m	[1, 12]
	C	{0.2, 0.4, 0.6, 0.8, 1, 1.2, 1.4, 1.6, 1.8, 2}
BPNN	m	[1, 12]
	N_h	{10, 15, 20, 25, 30, 35, 40, 45, 50}
KNN	m	[1, 12]
	K	{5, 7, 9, 11, 13, 15, 17, 19, 21}

where m is the length of the input vectors; N_h is the number of the hidden units; C is the penalty factor for the SVR model; and K is the number of neighbors for the KNN model.

rameter combinations were selected for the benchmark. The results of the grid search are given in Table 3. The results show that the direct multistep prediction was better in all the prediction tasks.

Results and Analysis

The Analysis of the Prediction Results

The predictions made by the proposed iterative multistep LSTM-NN model on the vacant parking space availability of the two parking lots (dash-dot curves) are presented in Figures 6 and 7, respectively. Considering that predictions output by the neural network models are vulnerable to fluctuation since they partially depend on the random initial weights, 100 prediction tasks were performed for each scenario and the average was adopted. The actual values are depicted with blue curves in Figures 6 and 7.

The results are mainly three-fold. First, for the short-term tasks, the iterative multistep LSTM-NN model made amazingly accurate predictions. The MAPE of the 5-min predictions to achieve 1.04 and 1.87% for the P-WZU and that in P-SM parking lots, respectively, are presented in Table 4; both of the accuracy rates were above 98%. The accuracy of the mid-term predictions was also encouraging. The MAPEs of the 15-min and the 30-min predictions were 2.26 and 4.22% for the P-WZU parking lot, and 3.74 and 6.09% for the P-SM parking lot. Compared to the recent work reported in [16] that had a MAPE of 10.6% in the 30-min prediction, our study achieved significant improvement. However, the longer the prediction period, the lower the accuracy. The MAPEs of the 45- and 60-min predictions increased to 6.11 and 8.26% for the P-WZU parking lot and to 8.24 and 10.39% for the P-SM parking lot. This is natural since the long-term predictions suffer from a weak time correlation between the historical information and the predictions, as discussed in the subsection "The Multistep Prediction Approach" in the section "Recurrent Neural and LSTM Networks."

Second, it can be observed that there appeared to be obvious gaps between the predictions and the actual values during the flat peak period in the long-term predictions, as illustrated in Figure 6(e). However, actually, the maximum gap was only about 11.5, with an absolute percentage error of 4.2%. Instead, the worst cases were seen with the violent fluctuations of the actual values, as emphasized by the green dotted circles in Figure 6(e). The violent fluctuations in long-term predictions result in growing uncertainties, the accumulation of errors, and the lack of internal information, which introduce dramatic errors, i.e., the absolute percentage error can reach above 30% and make an accurate prediction extremely difficult. However, fortunately, this kind of absurd prediction accounts for only less than 4% of all the 60-min predictions.

Table 3. The results of the grid search for the benchmark models.

		GRU-NN		SAE		SVR		BPNN		KNN	
Parking Lot/Task		m	N_h	m	N_h	m	C	m	N_h	m	K
P-WZU	5 min	6	20	7	100	5	2.0	8	20	2	11
	15 min	9	30	5	100	5	1.4	8	30	4	5
	30 min	11	80	12	500	8	2.0	7	15	6	11
	45 min	9	30	12	200	6	1.8	8	20	5	13
	60 min	9	100	9	300	8	1.6	7	15	5	15
P-SM	5 min	8	90	7	400	5	2.0	11	20	3	9
	15 min	10	40	8	200	12	2.0	12	25	5	9
	30 min	11	80	11	200	12	1.8	12	15	6	15
	45 min	12	40	12	200	12	1.8	11	35	6	15
	60 min	9	10	9	100	12	1.4	11	30	6	15

All multistep prediction methods are direct multistep prediction.

Consequently, in most cases, our predictions were convincing. Third, it can be determined from Table 4 that the predictions for the P-WZU parking lot were more accurate than those for the P-SM parking lot due to its mild traffic volume fluctuations, as previously presented in Figure 5.

Table 4 lists the comparisons of prediction errors between the proposed iterative multistep LSTM-NN model and the benchmark models. On the one hand, it can be observed that the LSTM-NN model achieved the best performance in all the predictions from the perspective of the RMSE. By this observation, it can be concluded that the LSTM-NN model outperformed all the benchmark models since the MSE was adopted as the loss function when training all the models. On the other hand, however, it can be observed that performance comparison results are a little inconsistent in different indicators. The LSTM-NN model is slightly inferior in a few tasks with regard to MAE and MAPE. Specifically, the values of MAE and MAPE of the 5-, 15-, and 30-min predictions made by the LSTM-NN model for the P-WZU parking lot were slightly higher (less than 2%) than those of predictions made by the GRU-NN model, and the value of the MAE of the 60-min prediction made by the LSTM-NN model for the P-SM parking lot was larger than that of the KNN model by nearly 1.3 parking spaces (less than 0.15% of the capacity of the P-SM parking lot). This kind of inconsistency is common and has already been reported in some studies [18], [44]. In the present study, we trained the models to minimize the MSE, but not the MAE nor the MAPE. So it makes sense that, although the LSTM-NN model does the best in terms of the MSE, it loses a little in the MAE and the MAPE in a few cases.

The Computational Complexity Analysis

The computational complexity of the LSTM-NN model was presented in [23]. In detail, the computational complexity of the LSTM per time step is $O(w)$, where w is the total number of parameters codetermined by the number of output units, the number of hidden units, and the number of input units.

We also measured the CPU times needed by the proposed LSTM-NN model and all the benchmark models per prediction. The results are listed in Table 5. All the experiments were performed on an Intel Core i5-6500 CPU with 3.20 GHz, 16 GB RAM, and Windows 7 Professional operating system. The values were averaged from the CPU times of 30,000 predictions. It is natural that the LSTM-NN model would consume more CPU time than the BPNN model, the SVR model, and the KNN model since it involves circular networks. However, the CPU time of less than 10 μs of the LSTM-NN model can fully satisfy the requirements of the prediction system.

Conclusion

In this article, we proposed a vacant parking space availability prediction model by taking advantage of the inherent strong points of the LSTM-NN model in time-series prediction. The only information needed for the suggested model is sufficient historical information on parking space availability. The key parameters of the LSTM-NN model are optimally determined by the grid search method. Two multistep prediction methods—i.e., the iterative multistep prediction and the direct multistep prediction—are recommended and integrated into the LSTM-NN model to deal with the long-term prediction. We applied the prediction model to two different parking lots: one is the private parking lot of Wenzhou University, and the other is a large public parking lot at Santa Monica. To verify the effectiveness of the advocated LSTM-NN model, it was extensively benchmarked with five other models—namely, the GRU-NN, SAE, SVR, BPNN, and KNN models—whose key parameters were also optimized by a grid search. The results show that

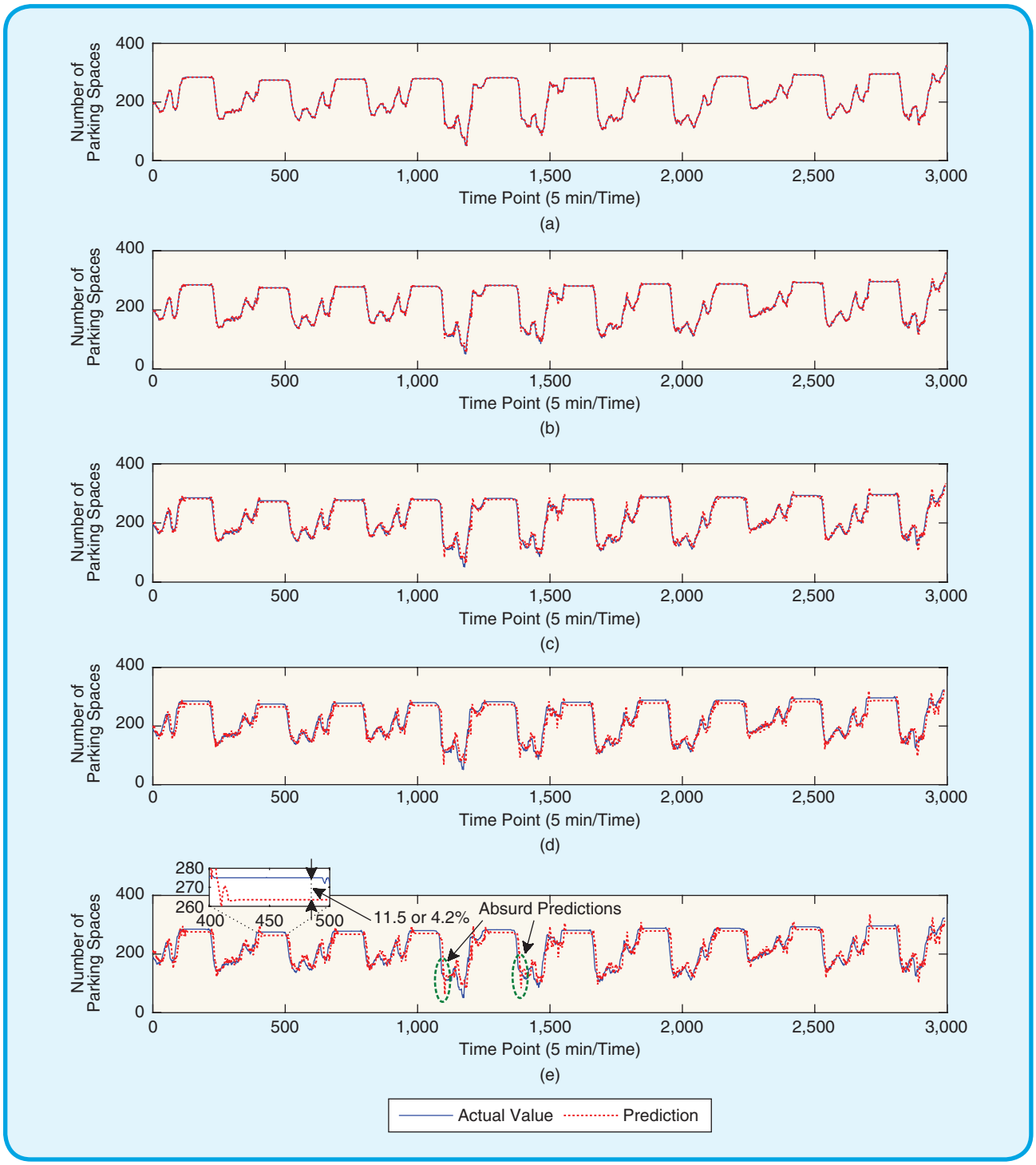


FIG 6 The prediction made by the LSTM-NN model for the P-WZU parking lot. The (a) 5-, (b) 15-, (c) 30-, (d) 45-, and (e) 60-min predictions.

the proposed multistep LSTM-NN model outperforms all the other models, especially in term of long-term predictions for large parking lots with marked traffic fluctuation.

Considering further improvements to the prediction accuracy and the application of the LSTM-NN prediction model, we suggest the following three interesting directions for future research:

- It will be potentially effective to consider more relative factors such as the weather, the locations of parking lots, holiday information, and so on, to further improve the prediction accuracy since all these fac-

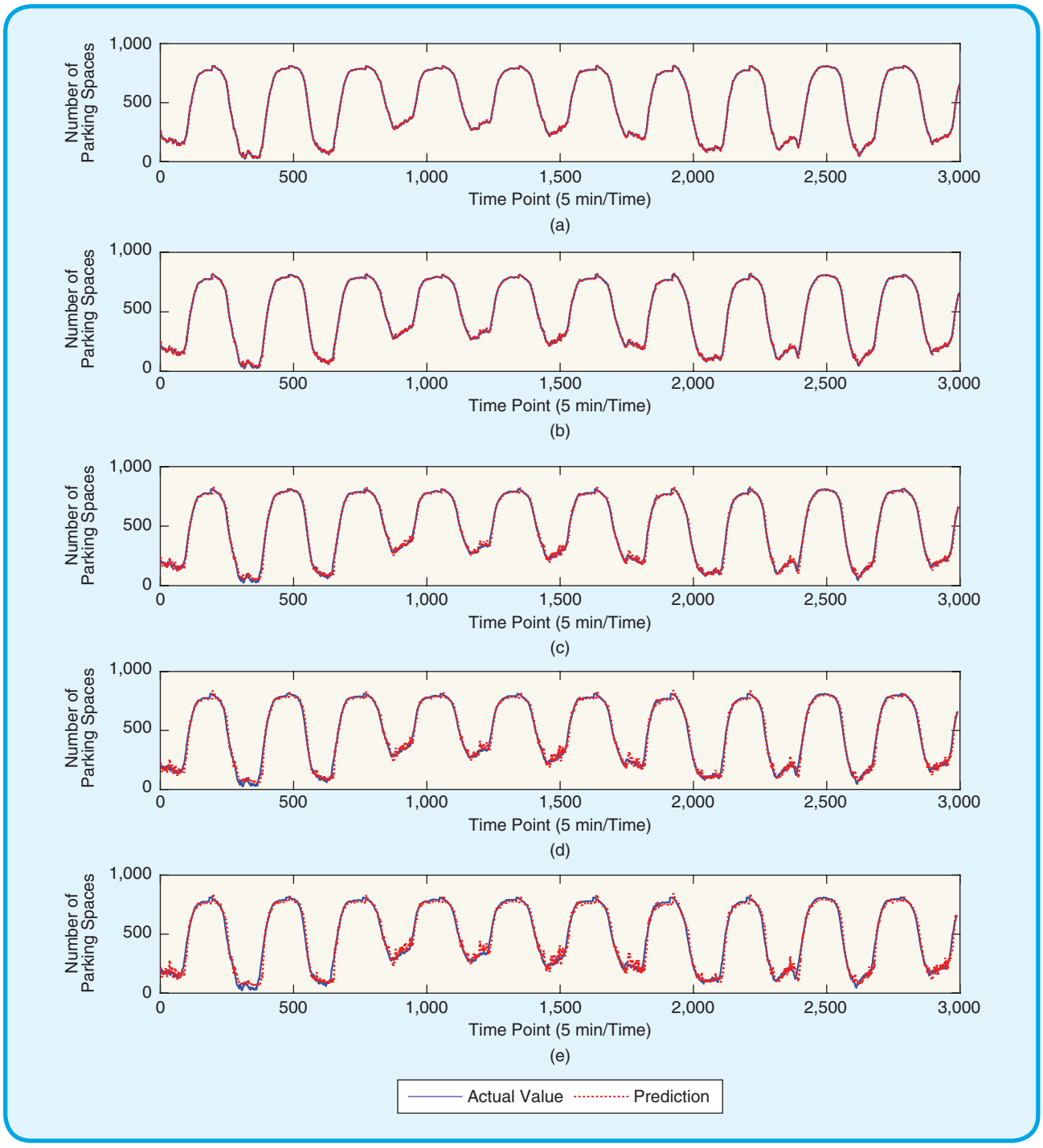


FIG 7 The prediction made by the LSTM-NN model for the P-SM parking lot. The (a) 5-, (b) 15-, (c) 30-, (d) 45-, and (e) 60-min predictions.

tors have strong correlation with the occupation of the parking lots.

- Long-term predictions imply that there is a large time interval between the historical data and the time point to be predicted. It will be interesting to study how to deal with the weak time correlation between the historical information and the predictions.
- Moreover, considering that curbside parking shows some features that differ from those of aggregated parking lots, it will be very meaningful to clarify whether the model proposed in this article is also effective at predicting parking space availability for curbside parking. Although the answer is very likely yes, since the problem is also a time-series prediction, we cannot verify it

Table 4. The performance comparisons among the LSTM-NN, GRU-NN, SAE, SVR, and KNN models.

		P-WZU Parking Lot					P-SM Parking Lot				
Methods	Index	5 min	15 min	30 min	45 min	60 min	5 min	15 min	30 min	45 min	60 min
LSTM-NN	MAE	1.8998	4.0712	7.9760	**11.0812**	**14.8386**	**3.9610**	**7.9560**	**13.4760**	**19.3825**	25.7805
	MAPE (%)	1.04	2.26	4.22	**6.11**	**8.26**	**1.87**	**3.74**	**6.09**	**8.24**	**10.39**
	RMSE	**2.9408**	**6.5230**	**11.6859**	**17.0129**	**22.4047**	**5.7730**	**11.3505**	**18.8193**	**27.1090**	**36.1273**
GRU-NN	MAE	**1.8289**	**4.0731**	**7.8210**	11.6757	15.7144	4.0095	8.0369	13.7732	19.4533	26.9216
	MAPE (%)	**1.02**	**2.24**	**4.13**	6.27	8.28	1.92	3.72	6.35	8.72	11.61
	RMSE	2.9552	6.5769	11.7958	17.1582	22.5020	5.8328	11.5788	19.3925	27.8878	36.9212
SAE	MAE	2.5601	4.6007	8.1674	12.6573	15.8561	4.8770	9.0227	14.4722	18.6743	26.6520
	MAPE (%)	1.41	2.53	4.37	6.64	8.41	2.42	4.18	7.08	9.22	11.94
	RMSE	3.7578	7.2443	12.1538	17.7494	22.9650	7.0436	12.9150	20.2818	27.7739	36.8098
SVR	MAE	1.8864	5.4605	10.0356	13.8804	17.0156	10.1219	10.4375	14.9415	21.5783	28.6769
	MAPE (%)	1.05	2.79	5.02	7.08	8.86	3.50	5.40	6.75	9.15	11.66
	RMSE	3.0471	7.1577	12.8839	18.3019	23.4130	12.5486	14.4053	20.0761	28.6902	38.1432
BPNN	MAE	1.9059	4.4099	8.0111	12.0122	15.9940	4.2179	8.2111	14.3228	20.8814	28.0337
	MAPE (%)	1.05	2.37	4.24	6.39	8.53	1.95	4.06	6.41	8.93	11.58
	RMSE	2.9727	6.7771	12.2270	17.8872	23.3361	6.0052	11.6663	19.7064	28.4814	38.0331
KNN	MAE	2.2355	4.7122	7.7909	11.2680	15.0981	4.5865	8.8958	14.2145	19.4964	**24.4746**
	MAPE (%)	1.21	2.56	4.29	6.17	8.26	2.12	4.32	7.24	9.74	12.00
	RMSE	3.6739	7.3238	12.0931	17.1115	22.4564	6.6630	12.9907	21.1948	29.8713	39.1978

The values in bold indicate the best performances.

Table 5. CPU times per prediction.

	LSTM-NN	GRU-NN	SAE	BPNN	SVR	KNN
CPU time (μs)	9.47	9.67	44.23	5.93	1.7	6.97

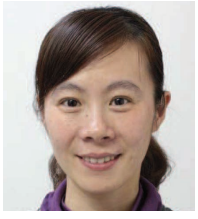
at this time because of the lack of current ground truth data on curbside parking.

Acknowledgments

This work was supported in part by the Zhejiang Provincial Natural Science Foundation of China (LZ20F010008 and LQ16G010006), the Fundamental Scientific Research Project of Wenzhou City (G20190021), and the Graduate Scientific Research Foundation of Wenzhou University. Junkai Fan and Qian Hu contributed equally to this work.

About the Authors

Junkai Fan (435589127@qq.com) earned his B.S. degree from the Department of Computer Science, Fuyang Normal University, China, in 2015 and his M.S. degree in computer software and theory from the College of Mathematics, Physics and Electronic Information Engineering, Wenzhou University, China, in 2019. His current research interests include machine learning, deep learning, and intelligent transportation.



Qian Hu (huqian@wzu.edu.cn) earned her B.S. degree in communication engineering from the Nanjing Institute of Posts and Telecommunications, China, in 2001 and her M.S. degree in circuits and systems from Zhejiang University, Hangzhou, in 2005. She is currently a lecturer with the College of Mathematics, Physics and Electronic Information Engineering, Wenzhou University. Her research interests are in the areas of wireless communications and artificial intelligence.

Yingying Xu (yyxu@wzu.edu.cn) earned her Ph.D. degree in communication and information systems from Beijing Jiaotong University in 2014. She is a lecturer in the Computer Science Department, Wenzhou University, China. Her research interests include data mining, intelligent transportation, and machine learning.

Zhenzhou Tang (mr.tangzz@gmail.com) earned his B.S. degree in electronic engineering and his M.S. degree in communications and information systems from Zhejiang University, China, in 2001 and 2004, respectively. He earned his Ph.D. degree in communications and information systems from Dalian University of Technology in 2015. In March 2004, he joined Wenzhou University and is currently a professor of the College of Computer Science and Artificial Intelligence. He is now serving as an associate editor of *IEEE Access*. His current research interest includes artificial intelligence, 5G communications, heterogeneous networks, cooperative communications, and network coding. He is a Senior Member of IEEE.

References

[1] T. Giuffrè, S. M. Siniscalchi, and G. Tesoriere, "A novel architecture of parking management for smart cities," *Procedia Soc. Behav. Sci.*, vol. 53, pp. 16–28, Oct. 2012. doi: 10.1016/j.sbspro.2012.09.856.

[2] D. C. Shoup, "Cruising for parking," *Transp. Policy*, vol. 13, no. 6, pp. 479–486, 2006. doi: 10.1016/j.tranpol.2006.05.005.

[3] D. Ayala, O. Wolfson, B. Xu, B. DasGupta, and J. Lin, "Pricing of parking for congestion reduction," in *Proc. 20th Int. Conf. Advances Geographic Information Systems*, 2012, pp. 43–51. doi: 10.1145/2424321.2424328.

[4] S. R. Rizvi, S. Zehra, and S. Olariu, "MAPark: A multi-agent auction-based parking system in Internet of Things," *IEEE Intell. Transp. Syst. Mag.*, early access. doi: 10.1109/MITS.2019.2953524.

[5] X.-S. Zhan, J. Wu, T. Jiang, and X.-W. Jiang, "Optimal performance of networked control systems under the packet dropouts and channel noise," *ISA Trans.*, vol. 58, pp. 214–221, Sept. 2015. doi: 10.1016/j.isatra.2015.05.012.

[6] J.-W. Hu, X.-S. Zhan, J. Wu, and H.-C. Yan, "Optimal tracking performance of NCSs with time-delay and encoding-decoding constraints," *Int. J. Control, Autom. Syst.*, vol. 18, no. 4, pp. 1012–1022, 2019. doi: 10.1007/s12555-019-0300-5.

[7] T. P. Hong, A. C. Soh, H. Jaafar, and A. J. Ishak, "Real-time monitoring system for parking space management services," in *Proc. 2013 IEEE Conf. Systems, Process and Control (ICSPC)*, pp. 149–153. doi: 10.1109/SPC.2013.6735122.

[8] C.-F. Yang, Y.-H. Ju, C.-Y. Hsieh, C.-Y. Lin, M.-H. Tsai, and H.-L. Chang, "iParking—a real-time parking space monitoring and guiding system," *Veh. Commun.*, vol. 9, pp. 301–305, July 2017. doi: 10.1016/j.vehcom.2017.04.001.

[9] G. Amato, F. Carrara, F. Falchi, C. Gennaro, C. Meghini, and C. Vairo, "Deep learning for decentralized parking lot occupancy detection," *Expert Syst. Appl.*, vol. 72, pp. 327–334, Apr. 2017. doi: 10.1016/j.eswa.2016.10.055.

[10] A. Klappenecker, H. Lee, and J. L. Welch, "Finding available parking spaces made easy," *Ad Hoc Netw.*, vol. 12, pp. 243–249, Jan. 2014. doi: 10.1016/j.adhoc.2012.03.002.

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

[12] L. Peng and H. Li, "Searching parking spaces in urban environments based on non-stationary Poisson process analysis," in *Proc. 2016 IEEE 19th Int. Conf. Intelligent Transportation Systems (ITSC)*, pp. 1951–1956. doi: 10.1109/ITSC.2016.7795871.

[13] J. Xiao, Y. Lou, and J. Frisby, "How likely am I to find parking?—A practical model-based framework for predicting parking availability," *Transp. Res. B, Methodol.*, vol. 112, pp. 19–39, June 2018. doi: 10.1016/j.trb.2018.04.001.

[14] J. Fan, Q. Hu, and Z. Tang, "Predicting vacant parking space availability: An SVR method with fruit fly optimisation," *IET Intell. Transp. Syst.*, vol. 12, no. 10, pp. 1414–1420, Nov. 2018. doi: 10.1049/iet-its.2018.5031.

[15] C. Badii, P. Nesi, and I. Paoli, "Predicting available parking slots on critical and regular services by exploiting a range of open data," *IEEE Access*, vol. 6, pp. 44,059–44,071, Aug. 2018. doi: 10.1109/ACCESS.2018.2864157.

[16] S. Yang, W. Ma, X. Pi, and S. Qian, "A deep learning approach to real-time parking occupancy prediction in transportation networks incorporating multiple spatio-temporal data sources," *Transp. Res. C, Emerg. Technol.*, vol. 107, pp. 248–265, Oct. 2019. doi: 10.1016/j.trc.2019.08.010.

[17] D. Wu et al., "Deep dynamic neural networks for multimodal gesture segmentation and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 8, pp. 1583–1597, 2016. doi: 10.1109/TPAMI.2016.2537340.

[18] J. Pang, J. Huang, Y. Du, H. Yu, Q. Huang, and B. Yin, "Learning to predict bus arrival time from heterogeneous measurements via recurrent neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 9, pp. 1–11, 2018. doi: 10.1109/TITS.2018.2873747.

[19] J. Li and Z. Wang, "Real-time traffic sign recognition based on efficient CNNs in the wild," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 3, pp. 1–10, 2018. doi: 10.1109/TITS.2018.2843815.

[20] M. Lee, Y. Xiong, G. Yu, and G. Y. Li, "Deep neural networks for linear sum assignment problems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 6, pp. 962–965, 2018. doi: 10.1109/LWC.2018.2843359.

[21] Y. Wang, W. Zhou, J. Luo, H. Yan, H. Pu, and Y. Peng, "Reliable intelligent path following control for a robotic airship against sensor faults," *IEEE/ASME Trans. Mechatronics*, vol. 24, no. 6, pp. 2572–2582, 2019. doi: 10.1109/TMECH.2019.2929224.

[22] Y. Wang, H. Shen, and D. Duan, "On stabilization of quantized sampled-data neural-network-based control systems," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3124–3135, 2016. doi: 10.1109/TCYB.2016.2581220.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

[24] R. Zhao, D. Wang, R. Yan, K. Mao, F. Shen, and J. Wang, "Machine health monitoring using local feature-based gated recurrent unit networks," *IEEE Trans. Ind. Electron.*, vol. 65, no. 2, pp. 1539–1548, 2017. doi: 10.1109/TIE.2017.2733438.

[25] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, 2014. doi: 10.1109/TITS.2014.2345663.

[26] W.-C. Hong, Y. Dong, F. Zheng, and S. Y. Wei, "Hybrid evolutionary algorithms in a SVR traffic flow forecasting model," *Appl. Math. Comput.*, vol. 217, no. 15, pp. 6733–6747, 2011. doi: 10.1016/j.amc.2011.01.073.

[27] R. Zhang, Y. Xu, Z. Y. Dong, W. Kong, and K. P. Wong, "A composite k-nearest neighbor model for day-ahead load forecasting with limited temperature forecasts," in *Proc. 2016 IEEE Power and Energy Society General Meeting (PESGM)*, pp. 1–5. doi: 10.1109/PESGM.2016.7741097.

[28] M. Caliskan, A. Barthels, B. Scheuermann, and M. Mauve, "Predicting parking lot occupancy in vehicular ad hoc networks," in *Proc. 2007 IEEE 65th Vehicular Technology Conf. - VTC2007-Spring*, pp. 277–281. doi: 10.1109/VETECS.2007.69.

[29] F. Caicedo, C. Blazquez, and P. Miranda, "Prediction of parking space availability in real time," *Expert Syst. Appl.*, vol. 39, no. 8, pp. 7281–7290, 2012. doi: 10.1016/j.eswa.2012.01.091.

[30] F. Bock and M. Sester, "Improving parking availability maps using information from nearby roads," *Transp. Res. Procedia*, vol. 19, pp. 207–214, Dec. 2016. doi: 10.1016/j.trpro.2016.12.081.

[31] F. Yu, J. Guo, X. Zhu, and G. Shi, "Real time prediction of unoccupied parking space using time series model," in *Proc. 2015 Int. Conf. Transportation Information and Safety (ICTIS)*, pp. 370–374. doi: 10.1109/ICTIS.2015.7232145.

[32] F. Bock, S. Di Martino, and A. Origlia, "A 2-step approach to improve data-driven parking availability predictions," in *Proc. 10th ACM SIGSPATIAL Workshop Computational Transportation Science*, 2017, pp. 13–18. doi: 10.1145/3151547.3151550.

[33] Y. Ji, D. Tang, P. Blythe, W. Guo, and W. Wang, "Short-term forecasting of available parking space using wavelet neural network model," *IET Intell. Transp. Syst.*, vol. 9, no. 2, pp. 202–209, 2014. doi: 10.1049/iet-its.2013.0184.

[34] T. Rajabioun and P. A. Ioannou, "On-street and off-street parking availability prediction using multivariate spatiotemporal models," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2913–2924, 2015. doi: 10.1109/TITS.2015.2428705.

[35] Y. Rong, Z. Xu, R. Yan, and X. Ma, "Du-parking: Spatio-temporal big data tells you realtime parking availability," in *Proc. 24th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2018, pp. 646–654. doi: 10.1145/3219819.3219876.

[36] E. H.-K. Wu, J. Sahoo, C.-Y. Liu, M.-H. Jin, and S.-H. Lin, "Agile urban parking recommendation service for intelligent vehicular guiding system," *IEEE Intell. Transp. Syst. Mag.*, vol. 6, no. 1, pp. 35–49, 2014. doi: 10.1109/MITS.2013.2268549.

[37] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-term residential load forecasting based on LSTM recurrent neural network," *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 841–851, 2017. doi: 10.1109/TSG.2017.2753802.

[38] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning*, vol. 4, no. 2; pp. 26–31, 2012. [Online]. Available: http://www.cs.toronto.edu/~hinton/coursera/lecture6/lec6e.mp4

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, arXiv:1412.6980

[40] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara, "Predicting human eye fixations via an LSTM-based saliency attentive model," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5142–5154, 2018. doi: 10.1109/TIP.2018.2851672.

[41] M. Turkoglu, D. Hanbay, and A. Sengur, "Multi-model LSTM-based convolutional neural networks for detection of apple diseases and pests," *J. Ambient Intell. Human. Comput.*, 2019. doi: 10.1007/s12652-019-01591-w.

[42] C. Lai, W. Chien, L. T. Yang, and W. Qiang, "LSTM and edge computing for big data feature recognition of industrial electrical equipment," *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 2469–2477, 2019. doi: 10.1109/TII.2019.2892818.

[43] "Santa Monica Open Data Portal," provided by the Mobility Division, Planning and Community Development Department, 2020. Accessed: July 2018. [Online]. Available: https://data.smgov.net/

[44] J. Mackenzie, J. F. Roddick, and R. Zito, "An evaluation of HTM and LSTM for short-term arterial traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 5, pp. 1847–1857, 2019. doi: 10.1109/TITS.2018.2843349.

ITS