Postprint of Short-Term Demand Forecasting Model for Public Bicycles Considering Variable Environmental Factors

Authors: Qiao Jian, Chen Shaobo, He Mengying

Date: 2022-04-07T15:01:57+00:00

Abstract

Existing short-term demand forecasting models for public bicycles have neglected the nature differences in how various environmental factors influence user demand, as well as the temporal dependencies of variable environmental factors. This paper distinguishes environmental factors into invariant factors that have been internalized within demand and variable factors that require separate consideration, and proposes a GCNN-LSTM-E model that employs Graph Convolutional Neural Networks (GCNN) to capture the non-Euclidean spatial dependencies of user demand, utilizes Long Short-Term Memory (LSTM) networks to capture the temporal dependencies of both user demand and variable environmental factors, and applies the influence of variable environmental factors to user demand through vector concatenation and fully connected networks. Experimental results demonstrate that the GCNN-LSTM-E model achieves optimal prediction performance at a 1-hour time granularity and outperforms all baseline models. This indicates that the model design is reasonable and effective, and that 1 hour is the most appropriate time granularity.

Full Text

Preamble

Vol. 39 No. 8
Application Research of Computers
ChinaXiv Partner Journal

Short-Term Demand Prediction Model for Public Bikes Considering Variable Environmental Factors

Qiao Jian†, Chen Shaobo, He Mengying
(School of Management, Northwestern Polytechnical University, Xi'an, Shaanxi 710072, China)

Abstract: Existing short-term demand prediction models for public bikes have overlooked both the differential nature of how various environmental factors influence user demand and the temporal dependency of variable environmental factors. This paper distinguishes environmental factors into invariant factors that have been internalized into demand and variable factors that require separate consideration. We propose a GCNN-LSTM-E model that employs Graph Convolutional Neural Networks (GCNN) to capture the non-Euclidean spatial dependencies of user demand, Long Short-Term Memory (LSTM) networks to capture the temporal dependencies of both user demand and variable environmental factors, and uses vector concatenation with fully connected networks to impose the influence of variable environmental factors on user demand. Experimental results demonstrate that the GCNN-LSTM-E model achieves optimal prediction performance at a 1-hour time granularity and outperforms all benchmark models, indicating that the model design is reasonable and effective, and that 1 hour is the most appropriate time granularity.

Keywords: public bike; short-term demand prediction; graph convolutional neural network; long short-term memory

---

0 Introduction

With the rapid growth in motor vehicle ownership, cities worldwide face increasing pressure from traffic congestion, environmental pollution, and energy consumption, prompting a vigorous shift toward public transportation development. Public bike systems (PBS), being green, energy-efficient, and environmentally friendly, play a crucial role in addressing the "first/last mile" problem, station connectivity, and short-distance commuting. As a one-way transportation system, PBS frequently experiences inventory imbalances across stations due to users renting and returning bikes at different locations, leading to significantly reduced demand fulfillment rates. Consequently, accurate short-term demand prediction is essential for timely inventory rebalancing. This makes short-term demand prediction for public bikes a critical research topic in public transportation.

By spatial granularity, short-term demand for public bikes can be categorized into city-level, cluster-level, and station-level predictions. City-level and cluster-level demand exhibit strong determinism and are relatively easy to predict, but their results cannot directly inform station-level inventory rebalancing. Station-level demand, while more random and challenging to predict, provides direct input for rebalancing operations. Station-level prediction models fall into two categories: statistical models and machine learning (ML) models. Common statistical models include Bayesian networks, ARIMA, multivariate regression, and

Poisson/negative binomial regression, while traditional ML models encompass support vector machines, gradient boosting trees, random forests, and neural networks. Statistical models offer better interpretability, whereas traditional ML models provide stronger predictive capabilities. However, both require manual feature engineering from raw data.

Deep learning (DL) models based on deep neural networks can automatically extract hidden features and possess superior learning and memory capabilities, making them the mainstream approach in traffic prediction in recent years. Zhang et al. pioneered DL models for short-term PBS demand prediction, attempting to use Convolutional Neural Networks (CNN) to capture spatial and temporal dependencies. While CNNs excel at image recognition due to their ability to compute convolutions on Euclidean spaces, the spatial dependencies in bike-sharing demand are non-Euclidean, and CNNs lack mechanisms to memorize past information, making them ill-suited for capturing temporal dependencies.

Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), derived from Recurrent Neural Networks (RNN), are specifically designed to capture temporal dependencies and have gained attention in traffic prediction. For instance, Wang et al. proposed LSTM and GRU models for single time-step PBS demand prediction; Liu et al. developed an LSTM model supporting multi-feature input and multi-time-step output; Xu et al. introduced an AM-LSTM model with attention mechanisms to address the non-strict periodicity of travel patterns; and Liu et al. proposed a bidirectional LSTM model with connections to both forward and backward hidden layers. While LSTM, GRU, and other DNNs excel at capturing temporal dependencies, they are not adept at modeling spatial dependencies.

Recently emerged Graph Convolutional Neural Networks (GCNN) compute graph convolutions on non-Euclidean spaces, perfectly aligning with the spatial dependency characteristics of bike-sharing demand. GCNN has thus replaced CNN as a mainstream DNN for PBS demand prediction. For example, Xiao et al. proposed the STGCN model using GCNN and gated CNN to capture spatial and temporal dependencies; Kim et al. combined GCNNs across three temporal granularities (hourly, daily, weekly) to capture spatiotemporal dependencies while incorporating weather and time-of-day factors through weighted sums with fully connected layer outputs; and He et al. employed multiple Graph Attention CNNs (GACNN) with attention mechanisms to capture traffic flows at different granularities and used a fully connected network to introduce weather and point-of-interest effects. Although GCNN matches the non-Euclidean nature of spatial dependencies, like CNN, it struggles with temporal dependencies. To address this, Lin et al. proposed the GCNNrec-DDGF model using separate GCNN and LSTM components for spatial and temporal dependencies, while Chai et al. introduced the MGCNN model employing multiple GCNNs to capture inter-station distances, interactions, and correlations, with LSTM processing the temporal dependencies in multi-graph convolution outputs.

User demand is influenced by numerous environmental factors. Some are invariant, such as points of interest (schools, stations, supermarkets), while others are variable, including temperature, rainfall, and wind conditions. Invariant environmental factors persist long-term, and their effects have already been internalized into user demand patterns; thus, they should not be explicitly considered in prediction models. Variable environmental factors are temporary, and their effects are not internalized, requiring explicit modeling to avoid compromising prediction accuracy. Most existing models ignore environmental factors entirely, while those that do consider them suffer from two key issues: (1) they fail to distinguish between invariant and variable factors, unnecessarily incorporating invariant factors that increase model complexity, and (2) they treat variable factors statically, overlooking their temporal dependencies.

This paper proposes a GCNN-LSTM-E model that considers only variable environmental factors. The model uses GCNN to capture non-Euclidean spatial dependencies of user demand, LSTM to capture temporal dependencies of both demand and variable environmental factors, and employs vector concatenation with fully connected networks to impose environmental influences on demand, thereby addressing the aforementioned limitations and improving prediction performance.

1 GCNN-LSTM-E Model Design

The model comprises three components: a demand feature capturer, an environmental feature capturer, and a feature fuser (Figure 1). The demand feature capturer learns and memorizes demand patterns along with their temporal and spatial dependencies. The environmental feature capturer learns and memorizes patterns and temporal dependencies of variable environmental factors. The feature fuser integrates variable environmental factors with user demand to reflect their influence.

Figure 1. Architecture of the GCNN-LSTM-E model

1.1 Demand Feature Capturer

User demand exhibits both spatial and temporal dependencies, with spatial dependencies being non-Euclidean—i.e., origin-destination relationships between stations are asymmetric many-to-many relationships. We represent this relationship using an adjacency matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$, where a_{ij} denotes the correlation between rental and return demands at stations i and j. Let user demand be a time series $\mathbf{x}_1, \dots, \mathbf{x}_T$, where $\mathbf{x}_t = [x_{t,1}, \dots, x_{t,N}]^\top$ represents demand at N stations at time t, and $x_{t,i}$ denotes demand at station i.

The demand feature capturer consists of a GCNN convolution block, an LSTM recurrent block, and a fully connected feedforward block to capture spatial and temporal dependencies. The GCNN block performs graph convolution operations on the input time series $\mathbf{x}_1, \dots, \mathbf{x}_T$ to extract non-Euclidean spatial dependencies, outputting results $\mathbf{A}\mathbf{x}_1, \dots, \mathbf{A}\mathbf{x}_T$ to the LSTM block. Here, $\mathbf{A}$ is a graph filter initialized through the following steps: (1) randomly generate a real symmetric adjacency matrix $\mathbf{A}$; (2) let $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$, where $\mathbf{I}_N$ is the identity matrix; (3) compute $\hat{\mathbf{A}} = \tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}$, where $\tilde{\mathbf{D}}$ is a diagonal matrix with elements $d_{ii} = \sum_j \tilde{a}_{ij}$.

The LSTM recurrent block captures temporal dependencies from the input sequence $\hat{\mathbf{A}}\mathbf{x}_1, \dots, \hat{\mathbf{A}}\mathbf{x}_T$. As shown in Figure 2, an LSTM cell outputs short-term state $\mathbf{h}_t$ and long-term state $\mathbf{c}_t$, controlled by a forget gate $\mathbf{f}_t$, input gate $\mathbf{i}_t$, and output gate $\mathbf{o}_t$. The forget gate determines which parts of the previous long-term state $\mathbf{c}_{t-1}$ to discard, the input gate controls what to add to $\mathbf{c}_t$, and the output gate regulates what to output to $\mathbf{h}_t$.

For inputs $\hat{\mathbf{A}}\mathbf{x}_t$ from the GCNN block, the LSTM cell updates the gates and cell state using equations (1)–(6) while computing the short-term state sequence $\mathbf{h}_1^d, \dots, \mathbf{h}_T^d$, thereby capturing temporal dependencies. Here, d denotes the number of hidden units.

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \cdot [\hat{\mathbf{A}}\mathbf{x}_t, \mathbf{h}_{t-1}] + \mathbf{b}_i) \quad (1) \tag{1}$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \cdot [\hat{\mathbf{A}}\mathbf{x}_t, \mathbf{h}_{t-1}] + \mathbf{b}_f) \quad (2) \tag{2}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o \cdot [\hat{\mathbf{A}}\mathbf{x}_t, \mathbf{h}_{t-1}] + \mathbf{b}_o) \quad (3) \tag{3}$$

$$\tilde{\mathbf{c}}_t = \tanh(\mathbf{W}_c \cdot [\hat{\mathbf{A}}\mathbf{x}_t, \mathbf{h}_{t-1}] + \mathbf{b}_c) \quad (4) \tag{4}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t \quad (5) \tag{5}$$

$$\mathbf{h}_t = \tanh(\mathbf{c}_t) \odot \mathbf{o}_t \quad (6) \tag{6}$$

where $\mathbf{W}_i, \mathbf{W}_f, \mathbf{W}_o, \mathbf{W}_c$ are weight matrices for the input, forget, output gates and cell state computation, respectively; $\mathbf{b}_i, \mathbf{b}_f, \mathbf{b}_o, \mathbf{b}_c$ are bias vectors; $[\cdot, \cdot]$ denotes vector concatenation; $\odot$ denotes element-wise multiplication; and $\sigma(\cdot)$ and $\tanh(\cdot)$ are the sigmoid and hyperbolic tangent functions.

Figure 2. Structure of an LSTM cell

The fully connected feedforward block computes the predicted demand $\hat{\mathbf{y}}_{t+1} \in \mathbb{R}^N$ (before incorporating environmental factors) from the LSTM hidden states $\mathbf{h}_1^d, \dots, \mathbf{h}_T^d$.

1.2 Environmental Feature Capturer

This component comprises an LSTM recurrent block and a fully connected feedforward block to capture temporal dependencies of variable environmental factors. Key environmental influences include points of interest, weather conditions, and air quality. Points of interest are invariant factors, while weather and air quality are variable. We posit that invariant factors, being long-term, have already been internalized into demand patterns, whereas variable factors like temperature, wind, rainfall, and air quality are temporary and not internalized, necessitating explicit modeling.

Let $\mathbf{x}_t^e \in \mathbb{R}^M$ denote the vector of temperature, wind, rainfall, and air quality at time t. Similar to the demand capturer, the environmental feature capturer uses its LSTM block to process the input sequence $\mathbf{x}_1^e, \ldots, \mathbf{x}_T^e$ and compute hidden states $\mathbf{h}_1^e, \ldots, \mathbf{h}_T^e$, capturing temporal dependencies of variable environmental factors. Here, e denotes the number of hidden units.

The fully connected feedforward block computes the predicted environmental factors $\hat{\mathbf{y}}_{t+1}^e \in \mathbb{R}^M$ from the LSTM hidden states, where M is the number of variable environmental factors.

1.3 Feature Fuser

This component consists of a concatenation block and a fully connected feedforward block to impose variable environmental influences on station-level demand. The concatenation block combines each station's predicted demand $\hat{y}_{i,t+1}^d$ with the predicted environmental factors $\hat{\mathbf{y}}_{t+1}^e$ into N vectors of length $M+1$: $\mathbf{c}_{i,t+1} = [\hat{y}_{i,t+1}^d, \hat{\mathbf{y}}_{t+1}^e]$, representing the application of M environmental factors to demand at N stations. The feedforward block computes the final demand prediction under environmental influence as $\hat{\mathbf{y}}_{t+1} = \mathbf{W}_c \cdot \mathbf{c}_{t+1}$, where $\mathbf{W}_c$ is a weight vector of length $M+1$.

During training, we minimize the absolute loss function $L = \sum_{i=1}^{N} |\hat{y}_{i,t+1} - y_{i,t+1}|$ using the Adam optimizer. To prevent overfitting, we specify an iteration threshold s. If validation performance does not improve after s iterations, training terminates and the current model is returned as final.

2.1 Data Preparation

We validate our model using New York City PBS data spanning January 1, 2017, to December 31, 2019, containing over 40 million transaction records. Preprocessing involved: (1) selecting stations operational throughout all three years, (2) computing hourly average rental/return demands, and (3) excluding stations with average hourly demand below 1, resulting in 395 stations over 26,280 hours. Figure 3 shows the hourly average demand distribution across these stations, with rental demands sorted in ascending order on the left and corresponding return demands on the right. Demand varies substantially across stations but remains remarkably consistent at individual stations. As shown in Figure 4, actual demand curves across three time granularities are relatively smooth. These observations indicate that most trips are round-trip journeys with strong determinism. The absence of direct competition contributes to this stability and motivated our dataset selection.

Figure 3. Distribution of average rental/return demands for 395 stations

Table 1 presents descriptive statistics of cumulative demand. With mean rental/return demand of approximately 3.75 per station per hour, the large gap between minimum and maximum values reveals that most stations have low demand, with over half averaging fewer than 3.75 hourly rentals/returns.

Table 1. Statistical description of cumulative demand

We also collected corresponding daily temperature, wind, precipitation, and air quality data for NYC using web scraping, aligning these with demand data across different time granularities.

2.2 Evaluation Metrics

We evaluate prediction accuracy using Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Coefficient of Determination (R^2):

$$\text{RMSE} = \sqrt{\frac{1}{MN}\sum_{i=1}^{N}\sum_{t=1}^{M}(\hat{y}_{i,t} - y_{i,t})^2} \quad (7) \tag{7}$$

$$\text{MAE} = \frac{1}{MN}\sum_{i=1}^{N}\sum_{t=1}^{M}|\hat{y}_{i,t} - y_{i,t}| \quad (8) \tag{8}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}\sum_{t=1}^{M}(\hat{y}_{i,t} - y_{i,t})^2}{\sum_{i=1}^{N}\sum_{t=1}^{M}(y_{i,t} - \bar{y})^2} \quad (9) \tag{9}$$

where M is the time duration, N is the number of stations, $\hat{y}_{i,t}$ and $y_{i,t}$ are predicted and actual values for station i at time t, and $\bar{y}$ is the mean. RMSE and MAE are scale-dependent, while R^2 is scale-independent. Lower RMSE and MAE indicate better performance, while higher R^2 is preferable.

2.3 Hyperparameters

We optimize hyperparameters via grid search, manually specifying search spaces in "start:step:end" format (e.g., "12:12:36" searches $\{12, 24, 36\}$). Using MAE on the validation set, we identify optimal values.

The GCNN-LSTM-E model has three hyperparameter categories: (1) time steps T for both recurrent blocks, hidden units d_1 in the demand LSTM, and hidden units d_2 in the environmental LSTM; (2) Adam learning rate α and batch size B; (3) iteration threshold s.

We split the dataset chronologically into training, validation (8 weeks each), and test sets. Table 2 shows search spaces, optimal values, and evaluation metrics.

Table 2. Search space and optimal value of hyperparameters, as well as evaluation index of the GCNN-LSTM-E model

2.4 Experimental Results

We compare our model against eight benchmarks: Historical Average (HA), GCNNrec-DDGF, LASSO, LSTM, MLP, SVR-linear, SVR-RBF, and XGBoost.

Table 3 presents prediction errors on the 1-hour rental demand test set. HA performs worst due to its simplicity. SVR-linear and LASSO improve through regression but remain limited by linear kernels. SVR-RBF's nonlinear kernel outperforms them. MLP surpasses SVR-RBF via deep learning, while LSTM further improves by capturing temporal dependencies. XGBoost's fine-tuning and regularization yield better performance than LSTM. GCNNrec-DDGF captures spatiotemporal dependencies, significantly outperforming XGBoost. GCNN-LSTM-E, which additionally captures temporal dependencies of variable environmental factors, achieves the best performance. Return demand results are similar.

Table 3. Prediction errors on the rental demand test dataset under the time granularity of 1h

Table 4 compares GCNN-LSTM-E and GCNNrec-DDGF across 1h, 0.5h, and 0.25h granularities. At 1h, GCNN-LSTM-E leads on all six metrics. At 0.5h, it leads on four metrics and ties on two. At 0.25h, it leads on two, ties on one, and trails on three, demonstrating overall superiority. Incorporating variable environmental factors' temporal dependencies clearly enhances prediction accuracy.

From equations (7)–(9), RMSE and MAE are scale-dependent while R^2 is scale-independent, enabling direct comparison across granularities. Table 4 shows R^2 is best at 1h granularity, though RMSE and MAE cannot be directly compared across scales.

Table 4. Experimental results on the rental/return demand test datasets under three kinds of time granularities

Figure 4 illustrates actual vs. predicted demand across three granularities. Finer granularities show lower accuracy, confirming that 1h yields the best RMSE and MAE. Smaller time units exhibit greater randomness and volatility, reducing prediction precision. Thus, 1h represents the most suitable granularity for balancing inventory management and prediction accuracy. Other models show consistent patterns.

Figure 4. Actual demand of New York PBS and predicted demand of the GCNN-LSTM-E model under three kinds of time granularities

3 Conclusion

As a one-way transportation mode, public bike systems require effective station inventory rebalancing to maintain high service levels. Accurate short-term demand prediction is prerequisite for effective rebalancing. However, most existing models ignore environmental factors, while those that do neglect the distinction between invariant and variable factors and overlook temporal dependencies of variable factors. This paper proposes the GCNN-LSTM-E model, which considers only variable environmental factors, uses GCNN for non-Euclidean spatial dependencies, LSTM for temporal dependencies of demand and environmental factors, and employs vector concatenation with fully connected networks to integrate environmental influences. Comparisons with eight benchmarks across three time granularities using real demand and environmental data demonstrate that GCNN-LSTM-E achieves optimal performance at 1h granularity, validating its design. The 1h granularity optimally balances inventory rebalancing needs and prediction accuracy.

Notably, New York's PBS faces no direct competition, unlike many Chinese systems competing with shared bike operators like Hellobike, Meituan, and Qingju. Demand stability varies across competitive landscapes. Our model was only tested on stable NYC data; future work should evaluate its generalizability across diverse competitive environments. Additionally, handling demand shocks from unexpected events remains a challenge.

Note: Figure translations are in progress. See original paper for figures.