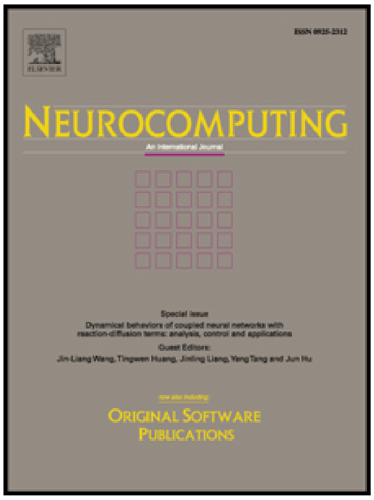
Communicated by Dr. Wang QI

Accepted Manuscript



LSTM-based Traffic Flow Prediction with Missing Data

Yan Tian, Kaili Zhang, Jianyuan Li, Xianxuan Lin, Bailin Yang

PII: S0925-2312(18)31029-4
DOI: https://doi.org/10.1016/j.neucom.2018.08.067
Reference: NEUCOM 19915

To appear in: *Neurocomputing*

Received date: 19 April 2018
Revised date: 28 July 2018
Accepted date: 19 August 2018

Please cite this article as: Yan Tian, Kaili Zhang, Jianyuan Li, Xianxuan Lin, Bailin Yang, LSTM-based Traffic Flow Prediction with Missing Data, *Neurocomputing* (2018), doi: https://doi.org/10.1016/j.neucom.2018.08.067

LSTM-based Traffic Flow Prediction with Missing Data

Yan Tian[1], Kaili Zhang[1], Jianyuan Li[2], Xianxuan Lin[1], Bailin Yang[1]

[1] *Schoor of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou 310014, P.R.China*

[2] *ENJOYOR Company Limited, Hangzhou 310030, P.R.China*

Abstract

Traffic flow prediction plays a key role in intelligent transportation systems. However, since traffic sensors are typically manually controlled, traffic flow data with varying length, irregular sampling and missing data are difficult to exploit effectively. To overcome this problem, we propose a novel approach that is based on Long Short-Term Memory (LSTM) in this paper. In addition, the multiscale temporal smoothing is employed to infer lost data and the prediction residual is learned by our approach. We demonstrate the performance of our approach on both the Caltrans Performance Measurement System (PeMS) data set and our own traffic flow data set. According to the experimental results, our approach obtains higher accuracy in traffic flow prediction compared with other approaches.

Keywords: Traffic Flow Prediction, Intelligent Transportation Systems, Deep Learning, LSTM

1. Introduction

Traffic flow prediction has been regarded as a vital and challenging topic in both academia and industry. The prediction is likely to help road users make better travel decisions, raise traffic operation efficiency, reduce carbon emissions, and alleviate traffic congestion. Moreover, its development can be applied to other time-series forecasting problems, such as crowd flow forecasting [1], clinical medical forecasting [2], weather

*Corresponding author
Email address: ybl@zjgsu.edu.cn (Bailin Yang[1])

forecasting [3], wind speed forecasting [4], electrical load forecasting [5], human behavior forecasting [6], and extreme event forecasting [7].

Methods that are based on the traditional machine learning make use of time-series approaches [8], probabilistic graphical models [9], and so on. However, none of these approaches can efficaciously explore nonlinear characteristics and multimodality in the data and their scalability is limited.

Recently, deep learning, especially Recurrent Neural Networks (RNNs) [10] and Long Short-term Memory (LSTM) [11], has been applied with success to time-series forecasting tasks. The inference accuracy can be improved when a large amount of annotated data is provided. According to the philosophy of the deep learning approach, if we have sufficient data, we can overcome the problems that conventional methods cannot conquer.

Nevertheless, due to missing data, irregular sampling, and varying length, the data remain difficult to explore with high efficiency. In a traffic environment, this problem becomes even worse because the traffic sensors are often controlled manually. Most approaches utilize only valid data to train the network model, which dramatically decreases the training set size. Some approaches take advantage of the mean to study the missing data or utilize a temporal smoothness constraint to infer the missing data [2]. However, these solutions often cause the compensation process to differ from the prediction models and the missing patterns to be explored inefficiently, thereby resulting in suboptimal analyses and predictions.

Traffic flow data have periodic characteristics; for example, each day, the traffic is heavy during commuting hours. This feature can be employed to infer missing values. The contribution of the periodic cue is difficult to determine. We may need a mechanism for dynamically learning the contribution ratio. In addition, multiple factors affect the traffic flow and the prediction may not be accurate if only the long-term dependency is utilized. A missing pattern is a type of temporal dependency information that can be utilized to compensate for the inference deviation.

In this paper, we develop a novel LSTM-based traffic flow forecasting method. The main contributions of this paper are as follows:

- We propose a new traffic flow prediction approach that not only acquires the long-term and short-term temporal dependencies of time-series observations but also utilizes the missing patterns to improve the prediction results.
- A new approach is presented for learning the prediction residual by explicitly combining the missing patterns based on the revised LSTM model.
- We construct a large database of traffic flow data so that the traffic flow prediction approach that is proposed in this paper can be evaluated. This database can also be used to evaluate other traffic flow prediction approaches. A link for downloading the data set can be found on our homepage [12].
- Experiments show that the proposed approach is competitive with other state-of-the-art approaches on traffic flow prediction.

The remainder of this paper is organized as follows: Section II reviews related studies on short-term traffic flow prediction, and section III introduces LSTM. Section IV presents the deep learning approach with LSTM as a building block for traffic flow prediction based on missing data. Section V discusses the experimental results and the concluding remarks are presented in Section VI.

2. Related Work

We review the machine-learning-based research on the time-series forecasting task, analyze the advantages and drawbacks of various methods, and formulate the missing value problem, which is a substantial challenge in this area of research.

Various machine learning methods have been employed in traffic flow prediction: time-series approaches, such as the Autoregressive Integrated Moving Average (ARIMA) [8]; probabilistic graphical models, such as Bayesian Network [13], Markov Chain [14], and Markov Random Fields (MRFs) [9]; and nonparametric approaches, such as Artificial Neural Networks (ANNs) [15], Support Vector Regression (SVR) [16], and Locally Weighted Learning (LWL) [17]. However, there are multiple reasons for fluctuation in the traffic flow and the patterns in the data are multimodal and difficult to learn. Moreover, these shallow network approaches require a high-dimensional space

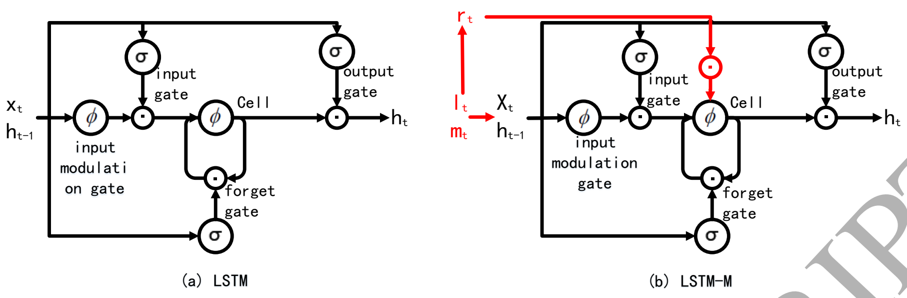


Figure 1: Architectures of (a) LSTM and (b) LSTM-M. In LSTM-M, masking vector $\mathbf{m}_t$ and time interval $\mathbf{l}_t$ are introduced to provide an initial prediction, which is denoted as $\mathbf{x}_t$, and learn the prediction residual in the block.

to model the complex mapping, which leads to the requirement of a huge amount of annotated data, and the overfitting problem becomes acute in the high-dimensional space.

By using a multilayer nonlinear structure, deep learning approaches have a strong ability to express multimodel patterns in data using a reduced number of dimension; in addition, the overfitting problem is alleviated.

Huang et al. [18] proposed using the Deep Belief Network (DBN) [19] for unsupervised traffic flow feature learning. Then a multitask regression layer was employed for supervised prediction. Later, Lv et al. [20] employed the Stacked Autoencoder (SAE) [21] to learn the generic traffic flow features and the model was trained in a greedy and layerwise fashion. To enhance the forecasting accuracy, Yang et al. [22] extended this approach and proposed a stacked autoencoder that was based on the Levenberg-Marquardt model and used the Taguchi method [23] to revise the optimization process and learn time-series features via a greedy and layerwise unsupervised learning approach. Recently, Polson et al. [24] combined the SAE model with an $L1$ regularization to identify the nonlinear spatiotemporal effects. Guo et al. [25] developed a fusion-based framework for improving the accuracy of traffic prediction.

In a time-series prediction task, the temporal relationship plays an important role. While DBN and SAE learn the implicit spatial relations of the data, they cannot model the temporal dependency explicitly. LSTM is designed to combine the short-term and long-term temporal information and exhibits superior time-series prediction performance. Ma et al. [26] used LSTM on remote microwave sensor data to capture the nonlinear traffic dynamic. To predict traffic flow under extreme conditions, Yu et al. [27] proposed a mixed deep LSTM approach that uses deep LSTM to model normal traffic and SAE to simulate interruptions from accidents. To study the temporal-spatial correlations in traffic flow, Zhao et al. [28] proposed a new LSTM network in which the two dimensions directly represented the temporal-spatial correlation. To obtain spatial-temporal traffic information within a transport network topology, Convolutional LSTM [29] and Graph Convolution Gated Recurrent Unit (GC-GRU) [30] were utilized to determine the temporal relation.

Although LSTM networks have achieved competitive results in traffic flow prediction, there has been little work on handling missing values in the LSTM network structure. It is proved that when a missing value is imputed via mean or temporal smoothing, it is impossible to distinguish whether the value is an imputed missing values or a true value. Simply concatenating the valid masking and the time interval vectors fails to exploit the temporal structure of the missing values. Recently, Che et al. [31] employed the mean and the last observation data in the LSTM framework to infer missing values, neglecting the pseudoperiodic characteristics of the traffic data. Cinar et al. [32] proposed utilizing an exponential function or a partition function in the attention weights to predict a missing value; however, the inference is not steady as the deviation is not modeled.

3. Long Short-Term Memory

To facilitate understanding of the method in this paper, we briefly introduce the mechanism of LSTM (the baseline of this method) and discuss why LSTM excels in sequence analysis.

Although RNNs have been proven successful in sequence prediction tasks, it can still be difficult to learn the long-term dependence, mainly due to the exploding/vanishing gradient problem that results from the gradient propagation of the recurrent network over many layers.

LSTM networks overcome this problem by incorporating memory units and the network learns when to forget previous memories and update memories. Fig. 1 presents an example of using such a recurrent process to generate descriptions.

We express a multivariate time series with D variables of length T as $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T)$, where $\mathbf{x}_t$ represents the t-th observations of all variables and x_t^d denotes the measurement of d-th variable of $\mathbf{x}_t$. The main contribution of the LSTM model is a memory cell $\mathbf{c}_t$ that contains information at time step t on the observations that have been obtained in this step. The cell is controlled by several gates and can either keep the value or reset the value according to the state of the gate. In particular, three gates are employed to control whether to forget the current cell value (forget gate $\mathbf{f}_t$), to read its input (input gate $\mathbf{i}_t$), and to output the new cell value (output gate $\mathbf{o}_t$); in addition, there is an input modulation gate, which is named $\tilde{\mathbf{c}_t}$. The gates and cell update and output are defined as follows:

$$\mathbf{i}_t = \sigma(\mathbf{W}_{xi}\mathbf{x}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1}), \tag{1}$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_{xf}\mathbf{x}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1}), \tag{2}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_{xo}\mathbf{x}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1}), \tag{3}$$

$$\tilde{\mathbf{c}_t} = \phi(\mathbf{W}_{xc}\mathbf{x}_t + \mathbf{W}_{hc}h_{t-1}), \tag{4}$$

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

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

where $\odot$ denotes the product operation and the $\mathbf{W}$ matrices are the network parameters. The LSTM networks are trained robustly as those gates deal well with exploding/vanishing gradients. The nonlinearities are the hyperbolic tangent $\phi()$ and the sigmoid $\sigma()$ and $\mathbf{h}_t$ is the hidden state.

4. LSTM Prediction with Missing Data

We analyze various patterns of missing data and design a novel prediction method that combines the characteristics of each pattern. In addition, we present a new approach for inferring the prediction residual by explicitly combining the missing pattern based on the revised LSTM model.

4.1. Patterns of Missing Data

There are many reasons for missing data in the traffic flow, such as malfunction of the sensor, manual system closure and errors in signal transmission.

Regardless of the cause, we find that, after statistical analysis, most missing observations can be divided into two categories, which are shown in Fig 2(A) and Fig 2(B):

I) short-period missing values, which can last less than 5 min. The invalid period may last less than 1 s in specific cases. These missing values are chiefly caused by unsteady equipment or a cluttered environment. Temporal smoothness is typically employed to deduce the missing values since the temporal information is rich in this situation;

II) long-period missing values, which can last hours, or even days. These missing values are principally caused by system closure. The mean values in the traffic flow are usually substituted for the invalid values. However, the prediction error is huge in this situation because of the scarcity of temporal information.

4.2. Multiscale Missing Value Prediction

To effectively handle missing data of both types in time series, we consider two important cases, especially in the traffic domain: I) if the value of the missing observation is close to that of its temporal neighbor, the missing observation belongs to the first class; II) if the input variables change periodically over time, the missing observation is of the second class. For instance, the traffic flow data repeat each week. We propose a new model, which is named LSTM-M, for managing missing data from the two categories, in which a long-period and a short-period mechanism are designed for modeling missing data in the input variables and hidden states are employed to capture the aforementioned properties.

Suppose a time series with missing values is denoted by $\mathbf{X} \in \mathbb{R}^{T\times D}$, and $s_t \in \mathbb{R}$ denotes the time-stamp when the t-th observation is acquired. We use a masking vector

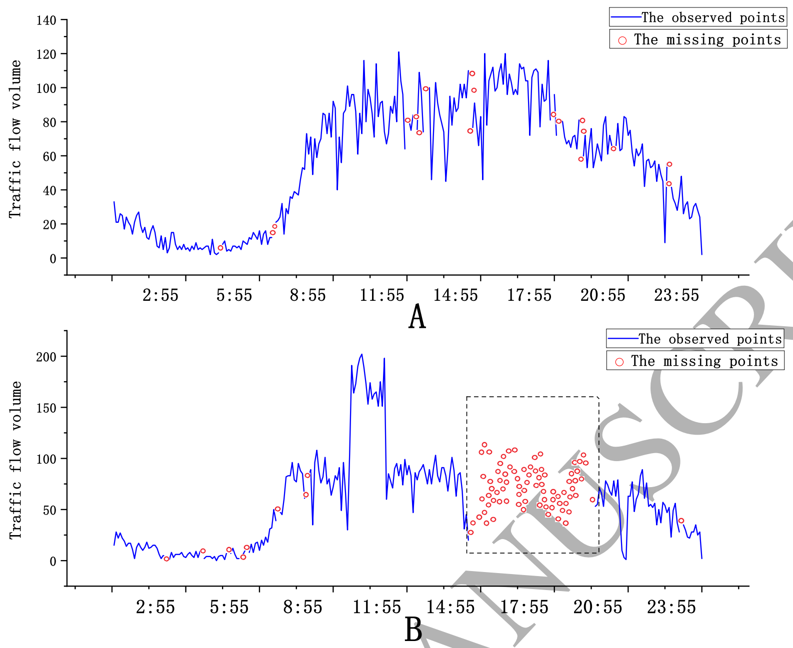


Figure 2: Patterns of the missing data: (A) short-period missing values and (B) long-period missing values.

$\mathbf{m}_t \in \{0, 1\}^D$ to represent the missing flag at time step t. The masking vector for x_t^d is obtained by

$$m_t^d = \begin{cases} 1 & \text{if } x_t^d \text{ is observed,} \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

For each variable d, we calculate the time interval l_t^d between its last observation and the current time

$$l_t^d = \begin{cases} s_t - s_{t-1} + l_{t-1}^d & \text{if } t > 1, m_{t-1}^d = 0, \\ s_t - s_{t-1} & \text{if } t > 1, m_{t-1}^d = 1, \\ 0 & \text{if } t = 1. \end{cases} \tag{8}$$

Then, we introduce weights $\mathbf{r}_t$ to control the impact, in consideration of the following: 1) each input variable in the traffic flow series has a unique meaning and time stamp and the weight should be flexible from 0 to 1 according to the time interval relative to the previous variables; 2) the missing patterns are of value in the prediction tasks and, thus, the weights should represent the patterns and be conducive to the inference tasks; and 3) as the missing patterns are undiscovered and possibly nonlinear, we use an exponential distribution to model the weights. Moreover, we make use of the training data in the traffic flow series to learn the weights rather than setting them as a priori:

$$\mathbf{r}_t = exp\{-max(\mathbf{0}, \mathbf{W}_r \mathbf{l}_t + \mathbf{b}_r)\}, \tag{9}$$

where $\mathbf{W}_r$ and $\mathbf{b}_r$ are parameters to be learned jointly with those in the LSTM network, and $\mathbf{W}_r$ is constrained to be diagonal to make the variable independent from the others. We choose the exponentiated negative rectifier to keep the weights monotonically decreasing in a proper range between 0 and 1 and keep other formulations monotonic with the weights in the same range. For example, the sigmoid function can be employed.

Our proposed time-series model for missing data incorporates two temporal prediction scales to obtain the missing data directly from the input values and implicitly in the RNN states. Considering a missing observation, we utilize an influence factor, which is denoted as m_t^d, to represent the weight of the previous observation and an influence factor, which is denoted as $1 - m_t^d$ to represent the weight of the periodic factor. Under this assumption, the observation can be expressed as

$$x_t^d = m_t^d x_t^d + (1 - m_t^d) r_t^d x_{t'}^d + (1 - m_t^d)(1 - r_t^d) x_{t''}^d, \tag{10}$$

where $x_{t'}^d$ is the previous observation of the d-th variable and $x_{t''}^d$ is the value in the previous period of the d-th variable ($t'' < t' < t$).

4.3. LSTM-based Residual Prediction

Sometimes, the model that is described in the previous subsection may not well forecast the missing data because the missing patterns are multimodel and cannot be exactly represented by short-time and long-time inferences. To capture a complex missing pattern in the data, we revise the traditional LSTM model and propose the LSTM-M model.

In the LSTM-M model, the influence factors, which are denoted as $\mathbf{r}_{c_t}$, are also employed on the cell state $\mathbf{c}_t$ to model the decay influence in the memory. In other words, the previous cell state, namely, $\mathbf{c}_{t-1}$, is weighted as $\mathbf{r}_{c_t} \odot \mathbf{c}_{t-1}$ before obtaining the new cell state, namely, $\mathbf{c}_t$, and the parameters in $\mathbf{r}_{c_t}$ are also jointly learned with those in the LSTM model.

In addition, as the missing pattern is intricate and multimodel, we simulate the residual between the predicted value and the ground-truth value in the LSTM unit by introducing a masking vector, which is denoted as $\mathbf{m}_t$, directly into the model. As a result, the update functions of the LSTM-M model are

$$\mathbf{i}_t = \sigma(\mathbf{W}_{xi}\mathbf{x}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{V}_i\mathbf{m}_t + \mathbf{b}_i), \tag{11}$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_{xf}\mathbf{x}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1} + \mathbf{V}_f\mathbf{m}_t + \mathbf{b}_f), \tag{12}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_{xo}\mathbf{x}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{V}_o\mathbf{m}_t + \mathbf{b}_o), \tag{13}$$

$$\tilde{\mathbf{c}_t} = \phi(\mathbf{W}_{xc}\mathbf{x}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1} + \mathbf{V}_c\mathbf{m}_t + \mathbf{b}_c), \tag{14}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t, \tag{15}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \phi(\mathbf{c}_t), \tag{16}$$

where $\mathbf{V}_i$, $\mathbf{V}_f$, $\mathbf{V}_o$, and $\mathbf{V}_c$ are new parameters for masking vector $\mathbf{m}_t$.

The framework of the LSTM-M model is illustrated in Fig 1(b). Masking vector $\mathbf{m}_t$ has two functions: I) it participates in the prediction of the values of the missing observations via the linear model that is described in Eq. (10); and II) it learns the prediction residual via the nonlinear functions in the LSTM unit. Although time interval $\mathbf{l}_t$ plays a similar role, it revises the previous cell state $\mathbf{c}_{t-1}$ rather than modifying the input modulation gate $\tilde{\mathbf{c}_t}$ because it is related to temporal information instead of instantaneous information.

5. Experimental Results

In this section, we compare the proposed approach with the state-of-the-art approaches in terms of effectiveness. The configurations, training details, measurements, results and discussion of the experiments are provided.

5.1. Hardware and Software Environments

We use a workstation with an Intel i7-4790 3.6GHz CPU, 32GB memory, and NVIDIA GTX Titan X graphics. Our algorithm utilizes the TensorFlow [33] to evaluate the performance and computational efficiency.

5.2. Training the Network

The LSTM-M networks in our experiments are trained via the widely used deep learning framework. Our prime architecture is comprised of 6 LSTM layers and 1 fully connected (FC) layer. The parameters are selected according to our engineering experience. The observation variable (input) has dimension 1 and the size of the hidden unit in the LSTM is 32 throughout this paper. In addition, the activation function for the LSTM layer is the tanh function. All the parameters in the LSTM layers and the FC layer are initialized from a uniform distribution over [-1.00, 1.00] and all bias terms are initialized to zero. We use the Adam algorithm [34] for optimization because the traffic flow data are noisy, and Adam is appropriate for problems with very noisy and/or sparse gradients. The learning rate is $r = 0.001$. Training is terminated when the maximum number of epochs (10 in our case) has been reached. Gradients are clipped if the norm of the parameter vector exceeds 5.0. The mini-batch size remains 32 and the other hyperparameters are optimized via cross-validation. Our model can be trained within 3 hours on a single Titan X GPU.

5.3. Data Sets

To evaluate the effectiveness of the approach that is presented in this paper, we utilize two data sets for experiments: the Caltrans Performance Measurement System (PeMS), which is widely used for traffic flow prediction tasks, and a data set that we built ourselves.

5.3.1. PeMS Data Set

The Caltrans Performance Measurement System (PeMS) is the most extensively used data set in traffic flow prediction. The data are collected from inductive loops and the objective is to predict the traffic flow on the road near the loop detector. Traffic data are collected every 30 s from over 15,000 individual detectors that are deployed state-wide in freeway systems across California. For highways with multiple detectors, we select data sequences with more than 1 vehicle is recorded in 15 min for training and testing. In this paper, the traffic flow series in the weekdays from January to March in 2013 are sorted for the experiments and the data in the last two weeks are selected as the testing set, while the other data are utilized as the training set. We deem two directions of the freeway to be independent and process them separately. In this paper, we mainly test the precision of our LSTM-M model in 5-min, 15-min, and 1-hour intervals. Hence, we export the traffic flow data for 5-min, 15-min and 1-hour intervals from the PeMS database. For highways with multiple detectors, traffic data that were collected by disparate detectors are averaged to obtain the mean flow for the freeway.

5.3.2. Our Traffic Flow Data Set

Our traffic flow data set consists of data that were collected from April 4th, 2015 to January 3rd, 2016, from spring to winter, including sunny, rainy and snowy days. Traffic data are gathered from numerous separated locations on an elevated road in HangZhou. Locations at which data are collected include the Shangcheng District (10 detectors), the Gongshu District (10 detectors), the Xihu District (8 detectors), and the Binjiang District (6 detectors). Each detector is deployed on a rod and tilted toward the road to capture vehicle information, including the license plate number, passage time, and vehicle speed. Then, we calculate the traffic flow data based on the license plate data, for example, the traffic value plus one when a new license plate is captured during a specified period. If no vehicle passes the observed road, the traffic flow in the specified period is zero, and the missing data are recorded as N/A. In this paper, the ratio of the training and testing data is fixed to 4.0.

We export these data and reorganize them into available traffic flow series. The traffic flow is defined as the total number of vehicles that pass through the specified place during a fixed period. For instance, the length of the time interval could be 5 min or 15 min and the data might be abnormal in some parts, for example, due to missing data or a flow burst event. In the stage of data collection, we find data are commonly missing, which may be related to the working condition of the detection equipment and the emergency situation of the road. The rates at which data are missing for various time intervals are listed in Table 1. The rate of missing data varies with the time interval. The larger the time interval is, the smaller the occurrence probability of overall data loss in this time period. However, the time interval cannot be excessively large because the traffic situation should be reflected accurately in a short period of time.

Table 1: Statistics of the rate of missing data for various time intervals.

Time interval	Missing rate
5 minutes	10%-32%
15 minutes	7%-26%

5.4. Evaluation Criteria

We use the mean absolute error (MAE), the mean relative error (MRE), and the root-mean-squared error error (RMSE) as the evaluation criteria to gauge the prediction accuracy, which are defined as

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|f_i - \hat{f}_i|, \quad (17)$$

$$MRE = \frac{1}{n}\sum_{i=1}^{n}\frac{|f_i - \hat{f}_i|}{f_i}, \quad (18)$$

$$RMSE = [\frac{1}{n}\sum_{i=1}^{n}|f_i - \hat{f}_i|^2]^{\frac{1}{2}}, \quad (19)$$

where n is the number of test sample, f_i is the real traffic flow in sample i, and $\hat{f}_i$ denotes the predicted traffic flow.

5.5. Experimental Results on Our Traffic Flow Data Set

We have demonstrated the performance of the proposed model on our traffic flow data set. Our traffic flow data set is demanding as the data contains numerous missing or invalid observations, which account for approximately 30% of the data set. We preprocess the missing data and use Eq. (10) to obtain an initial prediction. Then, we

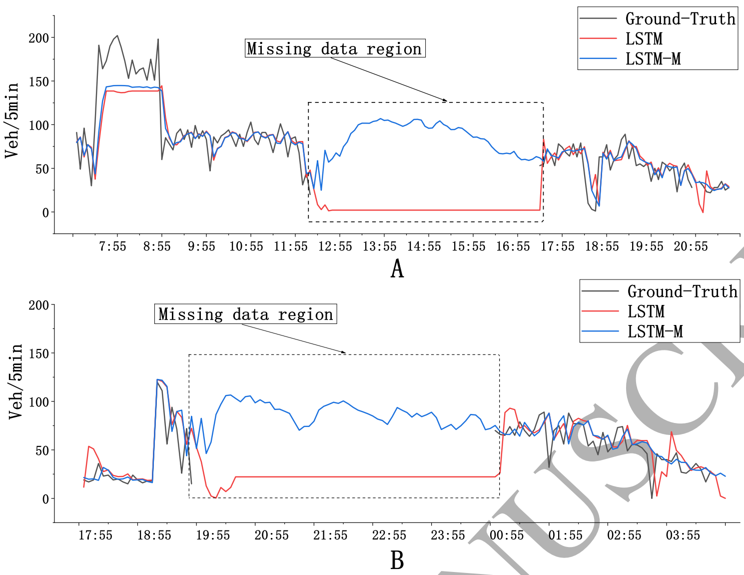


Figure 3: Experimental results for observation spots 'A' and 'B' in our data set. (A) Experimental results for observation spot 'A'. (B) Experimental results for observation spot 'B'.

utilize the LSTM-M model to compensate for the prediction residual. The traffic flow prediction results at observation spots 'A' and 'B' are shown in Fig 3(A) and Fig 3(B), respectively. We consider only the original traffic flow data; other factors, such as weather conditions, accidents, and the traffic flow density and speed, are not utilized in our approach. Our LSTM-M approach not only captures the short-term temporal pattern but also utilizes the pattern to improve the prediction results. However, the LSTM cannot capture the pattern, which is possibly due to the insufficient learning capability of the prediction residual of the nonlinear functions in the LSTM unit.

Performance comparisons between the LSTM and the LSTM-M approaches in short-time-interval sequences (5-min intervals) in peak (8-10 am and 17-19 pm) and off-peak times are presented in Table 2 and the results of the corresponding statistical tests of the LSTM-M approach are shown in Fig. 4.

Both models show high potential on our traffic flow data set. One reason for this lies in the low-traffic-flow conditions in this data set. When the traffic flow is low, any difference between the predicted value and the ground-truth value can cause a large

Table 2: Quantitative comparisons between our LSTM-M and LSTM during peak and off-peak times on the PeMS data set and our traffic flow data set.

Dataset		MAE		MRE		RMSE	
		LSTM	LSTM-M	LSTM	LSTM-M	LSTM	LSTM-M
PeMS	peak	18.47	16.76	7.76	6.29	27.16	25.32
	off-peak	14.32	12.28	5.43	4.87	22.54	21.29
	all day	15.92	13.88	6.45	5.12	25.49	22.67
Ours	peak	20.39	18.49	7.53	5.45	29.63	25.34
	off-peak	15.27	14.2	6.78	5.02	24.41	21.51
	all day	16.86	14.57	6.97	5.12	26.24	21.98

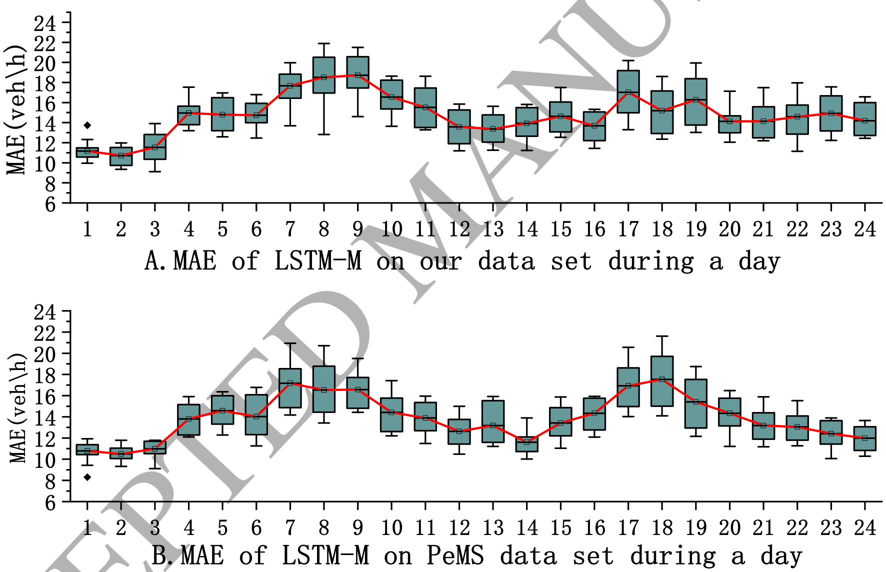


Figure 4: Statistical tests on the PeMS data set and our traffic flow data set. (A) Experimental results on our data set. (B) Experimental results on the PeMS data set.

variation in the MRE. Roads are mostly in the sparse and medium states and roads that are in the busy state contribute a small share of the data for the full day. According to Table 2, the MAE of the LSTM model is 16.86, while that of our LSTM-M model is 14.57 (improved by 2.29). The MRE of the LSTM model is 6.97%, whereas that of our

LSTM-M model is 5.12% (improved by 1.85%). In addition, the RMSE of the LSTM model is 26.24, whereas that of our LSTM-M model is 21.98 (improved by 4.26). Our LSTM-M approach has a lower error rate in comparison with the LSTM model, which is mainly because we explicitly model the prediction residual based on the pattern of the missing data.

5.6. Experimental Results on the PeMS Data Set

Our approach is also evaluated on the PeMS data set. For comparison with the other approaches, the traffic flow series in 5 min intervals are selected as the training and testing data. The method that was described in the previous subsection is utilized for data preprocessing and the evaluation results of the data preprocessing approach are listed in Table 3. 'Mean' refers to replacing the missing value by the mean of the sequence, and 'Temporal Smooth' refers to the completion of a missing observation using its temporal neighbor, 'Multiscale' refers to our multiscale missing value prediction approach.

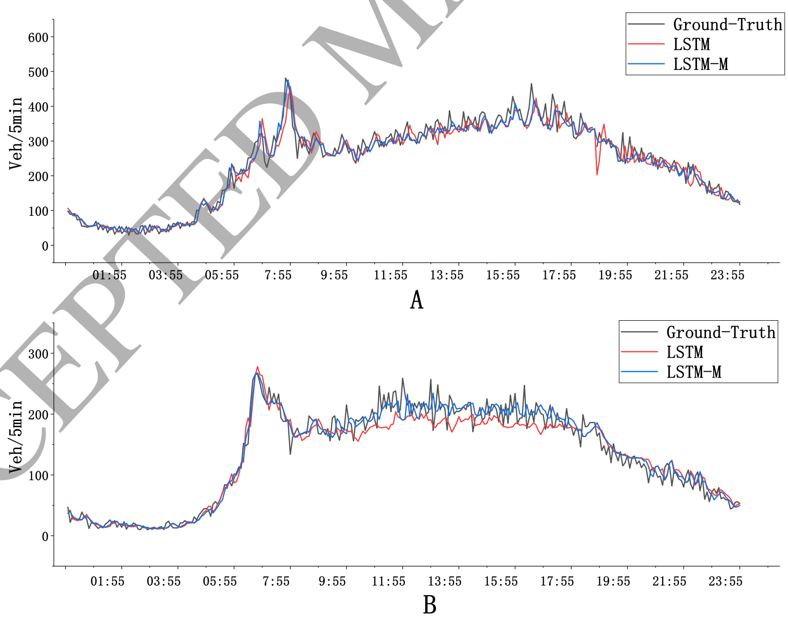


Figure 5: Experimental results on the PeMS data set. (A) Experimental results on freeway 'SR99-N'; (B) Experimental results on freeway 'US101-N'.

The traffic flow prediction results on freeways 'US101-N' and 'SR99-N' are presented in Fig 5(A) and Fig 5(B), respectively. The traffic in the PeMS data set is busier than that in our data set. The superior performance of our LSTM-M model in Fig 5(A) and Fig 5(B) is not as prominent as in Fig 3(A) and Fig 3(B) because few data in the PeMS database suffer from the missing value problem. Nevertheless, our LSTM-M model obtains smaller deviations between the predicted and ground-truth values compared to the LSTM model. Moreover, our LSTM-M model is more robust and stable than the LSTM model. The LSTM-M can capture the multimodel and nonlinear data patterns and can learn the residual between the initial prediction and the real flow, with steady inference.

Quantitative comparisons between the LSTM and the LSTM-M approaches on the PeMS data set are presented in Table 2. The average accuracy (one minus MRE) of the LSTM model is 93.2%-93.5%, while that of our LSTM-M model reaches almost 95.0%. For this reason, our LSTM-M approach achieves a gain of over 1.5% in the average accuracy compared with the LSTM model. Additionally, the MAE is comparatively large for heavy traffic flow (PeMS data set). The presence of more vehicles on the road may give rise to traffic bursts or accidents, and the traffic flow series may lead to fluctuations. Consequently, the task becomes more complicated in this scenario and large MAE is obtained when the traffic is heavy.

In addition, in terms of accuracy, we compare the proposed LSTM-M model and various state-of-the-art approaches: autoregressive Integrated Moving Average Model (ARIMA) is a classical algorithm of the time-series approach, support vector regression (SVR) [16] is a nonparametric regression approach, back propagation neural network (BPNN) [35] and radial basis function neural network (RBFNN) [36] are shallow neural networks, and stacked autoencoders (SAE) [20] and the LSTM network are deep neural networks. The experiment setup is the same as for our LSTM-M approach and the mean prediction errors (MAE, MRE and RMSE) on the testing data for freeways with 15-min and 60-min traffic flow sequences are listed in Table 3.

Deep neural networks, such as SAE, LSTM, and our LSTM-M, are superior to shallow neural networks (such as RBFNN and BPNN), because the deep architecture can learn more complex patterns than the shallow networks due to their ability to extract

Table 3: Quantitative comparisons among various preprocessing approaches and prediction approaches on the PeMS data set.

Model	15 min			60 min		
	MAE	MRE	RMSE	MAE	MRE	RMSE
Mean	34.87	6.89	50.12	122.09	6.18	182.74
Temporal Smooth	34.21	6.62	50.03	121.2	6.03	181.6
Multi-scale	**33.5**	**5.94**	**48.76**	**117.42**	**5.98**	**164.7**
BPNN [35]	60.8	10.9	94.1	202.8	9.8	321.5
SVR [16]	38.7	8.0	62.3	372.9	22.1	607.5
ARIMA [8]	38.5	7.6	60.4	353.6	21.8	619.8
RBFNN [36]	38.3	7.4	55.9	443.4	26.4	652.6
SAE [20]	34.1	6.75	50.0	122.8	6.21	183.9
LSTM [26]	34.4	6.68	49.87	120.4	6.12	178.9
LSTM-M	**32.2**	**5.24**	**47.04**	**114.6**	**5.47**	**154.8**

semantic representations from multiple layers.

Using back-propagation, the BPNN approach can adaptively learn the weights and biases of the network. Its MRE is 10.9% for the 15-min-interval traffic flow and 9.8% for the 60-min-interval traffic flow. ARIMA is a classical algorithm for time series analysis, in which an initial differencing step can be applied one or more times to eliminate the nonstationarity; its MRE is 7.6% for the 15-min-interval traffic flow and 21.8% for the 60-min-interval traffic flow. RBFNN uses a Gaussian function as the basis function to approximate the nonlinear analytical model. According to Table 3, the MRE of RBFNN is 7.4% for the 15-min-interval traffic flow, and 26.4% for the 60-min-interval traffic flow. Compared with RBFNN, SVR utilizes the radial basis kernel function to transform the traffic flow forecasting problem into a linear regression problem in Hilbert space and the MRE of SVR is 8.0% for the 15-min-interval data and 22.1% for the 60-min-interval data. The prediction errors of SVR and RBFNN decrease as the time interval increases because those approaches do not possess the memory cell and cannot capture the long-term temporal dependencies in the data series.

The MRE of SAE and the LSTM approach are less than 6.8% and their performances are similar. The SAE model uses a stacked autoencoder to extract high-level features and employs a logistic regression layer for prediction. The SAE model, through the utilization of the deep structure topology, performs well for traffic flow prediction as it can explore the implicitly multimodel pattern in noisy data. Additionally, the spatial relation is implicitly modeled and learned in SAE. Therefore, both spatial and temporal information are exploited to infer the prospective traffic flow. Naturally, RNNs are capable of learning temporal sequences as they have internal memory units for storing and processing previous information and LSTM is especially suitable for capturing the patterns in the traffic flow. This is because it incorporates memory units that allow the network to learn to forget previous hidden states and update hidden states when given new information at an appropriate time. SAE and LSTM have similar accuracy for the traffic flow forecasting problem; however, SAE cannot model the temporal relationship explicitly and LSTM cannot identify missing patterns and handle missing values in a data series.

For both short-time and long-time prediction of the traffic flow, our LSTM-M yields satisfactory results. Table 3 shows that the MAE improvement of LSTM-M reaches 2.20 in 15-min-interval series and 5.80 in 60-min-interval series compared with the LSTM approach and the MRE improvement of LSTM-M reaches 1.44 in 15-min-interval series and 0.65 in 60-min-interval series. Moreover, the RMSE improvement of LSTM-M reaches 2.83 in 15-min-interval series and 24.10 in 60-min-interval series. The LSTM-M approach for traffic flow prediction is a promising method because it has both a long- and short-term mechanisms for simulating missing data in the input variables and the residuals between the initial predictions and the ground-truth values, which are caused by the complex patterns in the missing data, are explicitly learned.

During the experiments, we observe that a subset of the time-series data, both in the long term and the short term, plays an important role in the inference stage, while the remaining data are weakly related to the predicted values. In the future, we plan to incorporate the attention mechanism into the traffic flow prediction task, localize the important data in the temporal space, and utilize these data for accurate inference.

6. Conclusion

We have proposed a novel approach for traffic flow prediction, which is called LSTM-M, that can infer the traffic flow even when values are missing from the data.

In this paper, a linear model is proposed for predicting the missing observations via the combination of temporal information of disparate scales. Based on a revised LSTM approach, we also learn the prediction residual. The experimental results on the PeMS data set and our constructed data set demonstrate that our approach outperforms several state-of-the-art methods in terms of accuracy.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (No. 61602407 and 61472363). Opening Foundation of Engineering Research Center of Intelligent Transport of Zhejiang Province (No. 2016ERCITZJ-KF02 and 2017ERCITZJ-KF04).

References

[1] J. Zhang, Y. Zheng, D. Qi, Deep spatio-temporal residual networks for citywide crowd flows prediction, in: AAAI, 2017, pp. 1655–1661.

[2] Z. Lipton, D. Kale, C. Elkan, et al., Learning to diagnose with lstm recurrent neural networks, in: ICLR, 2016, pp. 1456–1463.

[3] A. Grover, A. Kapoor, E. Horvitz, A deep hybrid model for weather forecasting, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015, pp. 379–386.

[4] A. Ghaderi, B. Sanandaji, F. Ghaderi, Deep forecast: Deep learning-based spatio-temporal forecasting, in: ICML, 2017, pp. 264–271.

[5] X. Qiu, Y. Ren, P. Suganthan, et al., Empirical mode decomposition based ensemble deep learning for load demand time series forecasting, Applied Soft Computing 54 (2017) 246–255.

[6] N. Phan, Y. Wang, X. Wu, et al., Differential privacy preservation for deep auto-encoders: an application of human behavior prediction, in: AAAI, 2016, pp. 1309–1316.

[7] N. Laptev, J. Yosinski, L. E. Li, S. Smyl., Time-series extreme event forecasting with neural networks at uber, in: ICML, 2017, pp. 384–391.

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

[9] M. Lippi, M. Bertini, P. Frasconi, Collective traffic forecasting, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 2010, pp. 259–273.

[10] P. Lingras, S. Sharma, M. Zhong, Prediction of recreational travel using genetically designed regression and time-delay neural network models, Transportation Research Record: Journal of the Transportation Research Board (1805) (2002) 16–24.

[11] S. Hochreiter, J. Schmidhuber., Long short-term memory., Neural Computation 9 (8) (1997) 1735–1780.

[12] [Online]. Available: http://www.yaletian.com/project/timeseriesprediction/.

[13] Z. Li, S. Jiang, L. Li, Y. Li, Building sparse models for traffic flow prediction: an empirical comparison between statistical heuristics and geometric heuristics for bayesian network approaches, Transportmetrica B: Transport Dynamics (2017) 1–17.

[14] D. Huang, Z. Deng, L. Zhao, B. Mi, A short-term traffic flow forecasting method based on markov chain and grey verhulst model, in: Data Driven Control and Learning Systems, 2017, pp. 606–610.

[15] K.-L. Li, C.-J. Zhai, J.-M. Xu, Short-term traffic flow prediction using a methodology based on arima and rbf-ann, in: Chinese Automation Congress, 2017, pp. 2804–2807.

[16] M. Castro-Neto, Y.-S. Jeong, M.-K. Jeong, L. D. Han, Online-svr for short-term traffic flow prediction under typical and atypical traffic conditions, Expert systems with applications 36 (3) (2009) 6164–6173.

[17] M. Shuai, K. Xie, W. Pu, G. Song, X. Ma, An online approach based on locally weighted learning for short-term traffic flow prediction, in: ACM SIGSPATIAL international conference on Advances in geographic information systems, 2008, pp. 45–53.

[18] W. Huang, G. Song, H. Hong, et al., Deep architecture for traffic flow prediction: Deep belief networks with multitask learning, TITS 15 (5) (2014) 2191–2201.

[19] G. E. Hinton, S. Osindero, Y.-W. Teh, A fast learning algorithm for deep belief nets, Neural Computation 18 (7) (2006) 1527–1554.

[20] Y. Lv, Y. Duan, W. Kang, et al., Traffic flow prediction with big data: a deep learning approach, TITS 16 (2) (2015) 865–873.

[21] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, in: ICML, 2008, pp. 1096–1103.

[22] H. Yang, T. Dillon, Y. Chen, Optimized structure of the traffic flow forecasting model with a deep learning approach, TNNLS 28 (10) (2017) 2371–2381.

[23] R. K. Roy, A primer on the Taguchi method, Society of Manufacturing Engineers, 2010.

[24] N. Polson, V. Sokolov, Deep learning for short-term traffic flow prediction, Transportation Research Part C: Emerging Technologies 79 (2017) 1–17.

[25] F. Guo, J. W. Polak, R. Krishnan, Predictor fusion for short-term traffic forecasting, Transportation Research Part C: Emerging Technologies 92 (2018) 90–100.

[26] X. Ma, Z. Tao, Y. Wang, et al., Long short-term memory neural network for traffic speed prediction using remote microwave sensor data, Transportation Research Part C: Emerging Technologies 54 (2015) 187–197.

[27] R. Yu, Y. Li, C. Shahabi, U. Demiryurek, Y. Liu, Deep learning: A generic approach for extreme condition traffic forecasting, in: SIAM International Conference on Data Mining, 2017, pp. 777–785.

[28] Z. Zhao, W. Chen, X. Wu, P. C. Chen, J. Liu, Lstm network: a deep learning approach for short-term traffic forecast, IET Intelligent Transport Systems 11 (2) (2017) 68–75.

[29] X. Cheng, R. Zhang, J. Zhou, W. Xu, Deeptransport: Learning spatial-temporal dependency for traffic condition forecasting, arXiv preprint arXiv:1709.09585.

[30] Y. Li, R. Yu, C. Shahabi, Y. Liu, Diffusion convolutional recurrent neural network: Data-driven traffic forecasting, in: ICLR, 2018, pp. 147–155.

[31] Z. Che, S. Purushotham, K. Cho, D. Sontag, Y. Liu, Recurrent neural networks for multivariate time series with missing values, Scientific reports 8 (1) (2018) 6085.

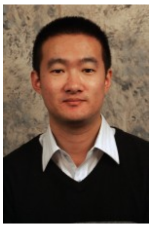
[32] Y. G. Cinar, H. Mirisaee, P. Goswami, E. Gaussier, A. Aït-Bachir, Period-aware content attention rnns for time series forecasting with missing values, Neurocomputing.

[33] M. Abadi, A. Agarwal, P. Barham, et al., Tensorflow: Large-scale machine learning on heterogeneous distributed systems, ArXiv preprint. [Online]. Available: https://arxiv.org/abs/1603.04467.

[34] D. Kingma, J. Ba, Adam: A method for stochastic optimization, in: ICLR, 2015, pp. 45–52.

[35] Y. Cai, Ipso-bpnn for short-term traffic flow prediction, Computer Engineering & Applications 48 (2012) 239–243.

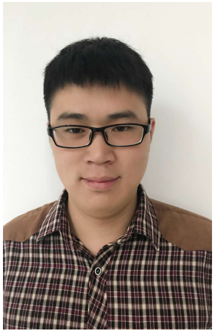
[36] H. Leung, T. Lo, S. Wang, Prediction of noisy chaotic time series using an optimal radial basis function neural network, TNN 12 (5) (2001) 1163–1172.



Yan Tian is received BSc in Communication Engineering from Hang-zhou Dianzi University, Hangzhou, China, Ph.D. degrees from Beijing University of Posts and Telecommunications, Beijing, China, in 2005 and 2011, respectively. Then he had a postdoctoral research fellow position (2012-2015) in the Department of Information and Electronic Engineering, Zhejiang University, Hangzhou, China. He is currently a Lecture of Computer Science and Technology in the School of Computer Science and Information Engineering, Zhejiang Gongshang University, China. His current interests are machine learning and pattern recognition, and he also works on image and video Analysis.

Kaili Zhang is received Bachelor's degree in Computer Science and Technology from Anyang Institute of Technology in 2016, Anyang, China. Master's degrees in Technology of Computer Application from Zhejiang Gongshang University, Hangzhou, China. Her research interests are in Computer Vision and image processing.

Jianyuan Li is received the B.Sc. in Applied Electronic Technology and the M.Sc. in Computer Science & Technology from Shan Xi Normal University, Xi'an, China, in 2001 and 2007, respectively. And he received the Ph.D. degree in Computer Science & Technology from Tongji University, Shanghai, China, in 2012. Then he had a postdoctoral research fellow position in Enjoyor Co. Ltd, Hangzhou, China, from 2014 to 2017. He is currently the Chief Technology Officer (CTO) in the research institute of Enjoyor Co. Ltd, Hangzhou, China. His research interests include machine learning and data mining.



Xianxuan Lin is received Bachelor's degree in Electronic Information Engineering from Yang-En University in 2014, Quan-zhou, China. Master's degrees in Technology of Computer Application from Zhejiang Gongshang University, Hangzhou, China. His research interests are in color processing of image and Sequential data prediction.

Bailin Yang is received the Doctor's degree in department of computer science from Zhejiang University in 2007. He is an professor in the department of computer and electronic engineering of Zhejiang Gongshang University. His research interests are in virtual reality, mobile graphics, data mining and mobile game.