Transportation Research Part C 129 (2021) 103228

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Real-time spatiotemporal prediction and imputation of traffic status based on LSTM and Graph Laplacian regularized matrix factorization☆

Jin-Ming Yang [a], Zhong-Ren Peng [b,*], Lei Lin [c]

[a] Center for Intelligent Transportation Systems and Unmanned Aerial Systems Applications Research, School of Naval Architecture, Ocean and Civil Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

[b] International Center for Adaptation Planning and Design, College of Design, Construction and Planning, University of Florida, Gainesville, FL 32611-5706, USA

[c] Goergen Institute for Data Science, University of Rochester, Rochester, NY 14620, USA

ARTICLE INFO

Keywords:
Spatiotemporal data
Traffic data prediction
Missing data imputation
Temporal matrix factorization
Deep learning
Intelligent transportation system

ABSTRACT

Accurate prediction of traffic status in real time is critical for advanced traffic management and travel navigation guidance. There are many attempts to predict short-term traffic flows using various deep learning algorithms. Most existing prediction models are only tested on spatiotemporal data assuming no missing data entries. However, this ideal situation rarely exists in real world due to sensor or network transmission failure. Missing data is a nonnegligible problem. Previous studies either remove time series with missing entries or impute missing data before building prediction models. The former may cause insufficient data for model training, while the latter adds extra computational burden and the imputation accuracy has direct impacts on the prediction performance. In this study, we propose an online framework that can make spatiotemporal predictions based on raw incomplete data and impute possible missing values at the same time. We design a novel spatial and temporal regularized matrix factorization model, namely LSTM-GL-ReMF, as the key component of the framework. The Long Short-term Memory (LSTM) model is chosen as the temporal regularizer to capture temporal dependency in time series data and the Graph Laplacian (GL) serves as the spatial regularizer to utilize spatial correlations among network sensors to enhance prediction and imputation performance. The proposed framework integrating with the LSTM-GL-ReMF model are tested and compared with other state-of-the-art matrix factorization models and deep learning models on three uni-variate and multi-variate spatiotemporal traffic datasets. The experimental results show our approach has a robust and accurate performance in terms of prediction and imputation accuracy under various data missing scenarios.

1. Introduction

Spatiotemporal data are a collection of spatially correlated time series such as traffic volume data and speed data collected from multiple sensors at a road network. In the era of big data, the dimension and scale of spatiotemporal data are rapidly expanding with the growth of the deployed sensor types and quantities. These large-scale spatiotemporal data can be utilized to provide wealthy

☆ This article belongs to the Virtual Special Issue on IG005572: VSI:Machine learning.

* Correspondence to: Gainesville, FL 32611-5706, USA.

E-mail addresses: yangjm67@sjtu.edu.cn (J.-M. Yang), zpeng@ufl.edu (Z.-R. Peng), Lei.Lin@rochester.edu (L. Lin).

https://doi.org/10.1016/j.trc.2021.103228
Received 18 August 2020; Received in revised form 24 April 2021; Accepted 15 May 2021
Available online 15 June 2021

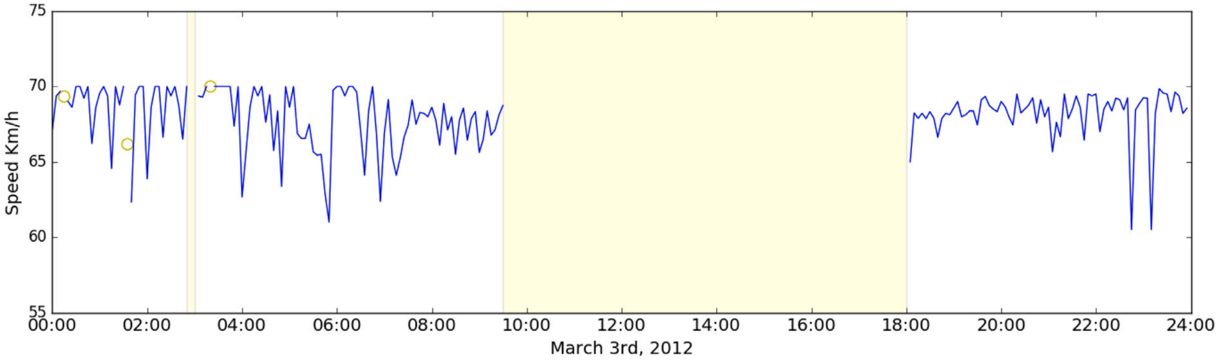


Fig. 1. Illustration of two data missing patterns.

information for decision support in constructing smart city. For example, online spatiotemporal traffic status prediction is critical in many intelligent transportation system (ITS) applications such as real-time adaptive traffic signal control system (Mirchandani and Head, 2001), vehicle navigation system (Lee et al., 2006) and predictive bus control framework (Andres and Nair, 2017). As another example, spatiotemporal prediction in air quality can also help with environment policy making (Paltsev et al., 2005). Hence, this topic has attracted wide attention in the past decade (Tong et al., 2017; Zhang et al., 2019; Che et al., 2018; Lin et al., 2018; Cui et al., 2019; Lin et al., 2015).

Time series prediction models mainly consider temporal correlations in individual time series such as Autoregressive model (AR) (Yule, 1926) and Autoregressive Integrated Moving Average (ARIMA) (Stellwagen and Tashman, 2013; Lin et al., 2013). Deep learning models such as Long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) and Gated recurrent neural network (GRU) (Chung et al., 2014) that can learn and preserve long term temporal correlations have also been applied for time series prediction. In contrast, there are also spatial correlations in spatiotemporal data except the temporal dependency. For example, data collected by nearby loop sensors tend to have similar temporal characteristics. Hence, spatiotemporal prediction models aim to utilize spatiotemporal correlations among sensors to improve model performance. As one of the emerging research areas, graph convolutional networks (GCN) that can capture the spatial correlations among sensors through adjacency matrix have shown promising results in spatiotemporal traffic state prediction (Lin et al., 2018; Cui et al., 2019).

Most existing studies mainly work on spatiotemporal data assuming no missing data entries (Chung et al., 2014; Yule, 1926; Stellwagen and Tashman, 2013; Lin et al., 2018; Cui et al., 2019). However, in reality, due to factors such as unstable power grids, sensor failures, data transmission network failures and periodic equipment overhauls, spatiotemporal data often suffer from the data missing issue. For example, the freeway Performance Measurement System (PeMS) maintained by the California Department of Transportation (Caltrans) has a missing sample rate of about 15% (Chen et al., 2003). The Los Angeles County freeway speed dataset METR-LA has a data missing rate of about 8.11% (Cui et al., 2020). There are basically two data missing patterns, namely the point-wise data missing(PM) where data is randomly lost at discrete time slots and continuous data missing (CM) where data is lost for a continuous period. Fig. 1 illustrates the METR-LA speed data collected by a loop sensor in March 3rd. Point-wise missing (PM) entries are marked with hollow dots. And continuous missing entries are marked with yellow panels.

There are several approaches to overcome the data missing issue in spatiotemporal prediction: omitting the missing entries, imputing data first before building prediction models (Hu et al., 2017), forward-fill that use predicted values or last observations to fill in missing entries along with prediction making (Cui et al., 2020), Bayesian modeling that treat massing data as random variables and infer them from their corresponding conditional distributions (Sun and Chen, 2019; Ma and Chen, 2018) and dynamic factor models such as state-space models (Zhang et al., 2014) and matrix factorization models (Yu et al., 2016). The removal of missing values may lead to insufficient data for model training, and cause potential overfitting. Except for a few models such as those based on Bayes and probability, the data imputation process before model building would introduce extra computational burden for most deep learning models. Further, a data imputation model with low accuracy will also jeopardize the spatiotemporal prediction accuracy. Fill in missing entries using previous predicted values could lead to severe error accumulation, especially when data is lost for a continuous long time. In addition, the MCMC sampling in Bayesian approaches do not run quickly on large datasets and may not converge easily for complex models (Ma and Chen, 2018).

Matrix factorization (MF) models have been applied for spatiotemporal data imputation (Salakhutdinov and Mnih, 2008; Lee and Seung, 2001) and prediction (Yu et al., 2016; Gultekin and Paisley, 2019; Sun and Chen, 2019) for partially observed data. MF models can capture global spatial correlations among sensors in the factorization process as well as temporal correlations through linear autoregressive (AR) regularizer (Yu et al., 2016; Gultekin and Paisley, 2019; Sun and Chen, 2019). In this study, we propose a framework based on MF which is able to make spatiotemporal predictions using raw incomplete data and perform online data imputation with real-time data collection simultaneously. We innovatively design a spatial and temporal regularized matrix factorization model, namely LSTM-GL-ReMF, as the key component of the framework. In LSTM-GL-ReMF, its temporal regularizer depends on the state-of-the-art Long Short-term Memory (LSTM) model (Hochreiter and Schmidhuber, 1997), and the spatial regularizer is designed based on Graph Laplacian (GL) spatial regularization (Cai et al., 2010). These regularizers enable the incorporation of complex spatial and temporal dependence into matrix factorization process for more accurate online prediction and imputation performance.

The proposed framework is tested on two spatiotemporal traffic datasets and a high-dimensional spatiotemporal air pollutant dataset, namely the Seattle Traffic Speed dataset, Metr-LA Freeway Speed dataset and Shanghai Pollutant Concentration dataset. Experiments demonstrate the proposed LSTM-GL-ReMF framework outperforms seven benchmark models, which include MF models TRMF (Yu et al., 2016) and BTMF (Sun and Chen, 2019), state-of-the-art deep learning models LSTM (Hochreiter and Schmidhuber, 1997), GRU-D (Che et al., 2018), GCN-DDGF (Lin et al., 2018), TGC-LSTM (Cui et al., 2019) and SGMN (Cui et al., 2020), in both prediction and imputation accuracy under various data missing scenarios. The main contributions of this paper are summarized as follows:

- We propose a novel LSTM-GL-ReMF model that captures nonlinear temporal dynamics and ensures the local spatial smoothness of spatiotemporal data, and then extend it to multi-variate version using tensor CP decomposition method.
- We propose an effective and efficient alternating method to solve the LSTM and Graph Laplacian regularized matrix factorization problem.
- We extensively compare LSTM-GL-ReMF with other state-of-the-art MF and deep learning models on three traffic datasets under various data missing scenarios, and show our model has robust and accurate performance.

The remainder of this paper is organized as follows. Section 2 introduces related work of this study. In Section 3, we introduce the proposed LSTM-GL-ReMF model and the online prediction and imputation framework. Section 4 introduces the experimental results conducted on the two spatiotemporal datasets under various data missing scenarios. Section 5 summarizes the study and discuss future research directions.

2. Related work

This section will first introduce relevant works in spatiotemporal prediction, then discuss previous prediction models based on incomplete data. Finally we will focus on matrix factorization studies that have been applied in both data imputation and prediction.

2.1. Spatiotemporal prediction models

Spatiotemporal prediction models such as GCN-DDGF (Lin et al., 2018), TGC-LSTM (Cui et al., 2019), ST-MGCN (Geng et al., 2019) and DAL (Qi et al., 2018) not only utilize the temporal dependence in time series, but also capture the spatial correlation among sensors. Graph Convolutional Neural Network with Data-driven Graph Filter (GCN-DDGF) (Lin et al., 2018) and Traffic Graph Convolutional Recurrent Neural Network (TGC-LSTM) (Cui et al., 2019) are recently proposed graph convolutional network (GCN) based models. GCN-DDGF is able to automatically learn the spatial dependence while TGC-LSTM defines spatial correlation based on traffic road network connectivity. Both models rely on LSTM to capture the temporal dependency. Spatiotemporal multi-graph convolution network (ST-MGCN) (Geng et al., 2019) further proposes multi-graph convolution to capture region correlations. Adjacency graph, funtionality graph and connectivity graph are used to incorporate spatial correlation, land use similarity and transportation connectivity of regions respectively. However, the construction of multi-graph requires additional information about land using, road type and road network structure. Deep Air Learning (DAL) (Qi et al., 2018) is a deep learning model that captures both spatial and temporal correlations by spatiotemporal semi-supervised learning which considers the spatial loss between the observations at the same time and the temporal loss between the observations at the same location.

2.2. Prediction models based on incomplete data

To work on incomplete data, one approach is to conduct data imputation first before building prediction models. A variety of methods have been developed in past decades to perform missing data imputation. These methods include: matrix/tensor factorization models which approximate partially observed data matrix/tensor by low rank matrices (Chen et al., 2019a,b), matrix/tensor completion models which minimize the rank of data matrix/tensor by singular value thresholding (Chen et al., 2020; Liu et al., 2012), KNN models which fill in the missing entries using the average of the k nearest neighbors (Zhang, 2012; Crookston and Finley, 2008), Expectation Maximization (EM) models which iteratively estimate missing entries and update distribution parameters (Su et al., 2008), etc. Based on these imputation methods, many prediction models for datasets with missing entries are proposed. Hu et al. (2017) uses matrix factorization to impute missing data first before applying LSTM network to make traffic data predictions. Sridevi et al. (2011) proposes to use an autoregressive based missing data imputer—ARLSimpute to patch missing data first before using linear or quadratic predictor to make data prediction. Purwar and Singh (2015) proposed a multiple imputation ensemble approach to patch missing data before using perceptron to make prediction.

On the other hand, there are also studies that construct prediction models directly on incomplete data. Anava et al. (2015) proposes a prediction model for data with missing values based on AR. The sequential model makes use of previous predictions to “fill in” the missing entries before making the next prediction. GRU-D (Che et al., 2018) integrated a decay mechanism for Gated Recurrent Unit deep learning model that explicitly captures data missing patterns apart from temporal modeling. It takes two data missing patterns, i.e. masking and time interval, into GRU units to mark missing entries and their last observations. However, the model has not been tested under different data missing ratios.

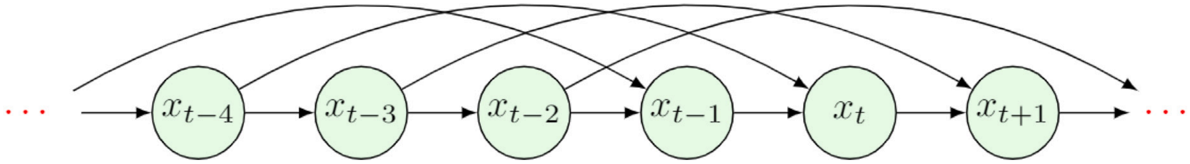


Fig. 2. Temporal correlation diagram.

2.3. Matrix Factorization models

Matrix Factorization (MF) is a kind of low rank process which is widely used in recommendation systems (Purushotham et al., 2012; Bokde et al., 2015) and missing data imputation (Salakhutdinov and Mnih, 2008; Lee and Seung, 2001). MF models have been applied for both prediction and imputation of spatiotemporal data. MF models such as Bayesian Probabilistic Matrix Factorization (BPMF) (Salakhutdinov and Mnih, 2008) and Non-negative Matrix Factorization (NMF) (Lee and Seung, 2001) patch missing entries through approximating the initial incomplete matrix by two low-rank submatrices. Tensor factorization (TF) models such as Bayesian Gaussian Tensor Canonical Polyadic (CP) Decomposition (Chen et al., 2019b) and Bayes augmented tensor factorization (Chen et al., 2019a) have also been exploited to perform data imputation.

For spatiotemporal prediction, temporally regularized MF models such as TRMF (Yu et al., 2016), OLMF (Gultekin and Paisley, 2019) and BTMF (Sun and Chen, 2019) incorporate temporal dependence into the factorization process. The three mentioned temporally regularized MF models all use linear Autoregressive model (AR) as temporal regularizer. TRMF (Yu et al., 2016) introduces a vector AR (VAR) regularizer for temporal modeling. BTMF (Sun and Chen, 2019) implements matrix AR (MAR) for temporal regularization which utilizes more parameters than VAR, and thus can capture more complex temporal information. The construction of BTMF under the Bayesian network makes the model more robust and does not require careful hyper-parameter tuning. However, the Gibbs sampling process in BTMF is highly time consuming which means that this kind of method is not very suitable for large-scale datasets. There are also MF models that explicitly models the local spatial correlation by implementing spatial regularizers such as Graph Laplacian (GL) (Cai et al., 2010), Directed Auto-Regression (DAR) (Takeuchi et al., 2017) and High-Order Fused LASSO (HOFL) (Takeuchi et al., 2017). GL uses spatial topology to penalize adjacent data discrepancies. DAR can automatically learn the link weights of topology graph. HOFL further encourages parameters in a given group to take identical values by $L1$ penalizing the discrepancies of parameters in the same group.

Although there have been numerous MF models for spatiotemporal prediction and data imputation, current MF studies either impute missing values in the history data matrix $Y \in \mathbb{R}^{m\times T}$ or make rolling prediction of future data vectors $y'_{t+1} \in \mathbb{R}^m, t+1 > T$. The proposed online MF framework focus on making prediction of data vector y'_{t+1} of the next time step $t+1$ and imputing the possible missing values in the newly observed data vector $y_t \in \mathbb{R}^m$ simultaneously on an online basis. Besides, the spatial and temporal regularizers in previous MF models can be further enhanced to capture more complex spatial and temporal correlations in the data. Different from the current temporal matrix factorization models such as TRMF (Yu et al., 2016) and BTMF (Sun and Chen, 2019) that use AR temporal regularization, the proposed LSTM and Graph Laplacian regularized matrix factorization model(LSTM-GL-ReMF) innovatively develop a neural network temporal regularizer based on the state-of-the-art recurrent network Long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997). Comparing to the AR regularization (Yu et al., 2016; Sun and Chen, 2019; Gultekin and Paisley, 2019) which are only able to extract linear temporal dependencies, the LSTM network based temporal regularizer is able to learn complex long term and short term non-linear temporal correlations. A novel Graph Laplacian spatial regularizer (Cai et al., 2010) is also implemented in the proposed spatially and temporally regularized matrix factorization model to address local spatial dynamics.

3. Methodology

In this section, we will first briefly introduce Temporal Regularized Matrix Factorization model (TRMF) (Yu et al., 2016), which serves as a basis to understand our methodology. Then we introduce the proposed LSTM and Graph Laplacian Regularized Matrix Factorization (LSTM-GL-ReMF) model and the online spatiotemporal data prediction and imputation framework.

3.1. Temporal regularized matrix factorization

Spatiotemporal data with M locations (sensors) and T time steps can be organized in a data matrix $\boldsymbol{Y} \in \mathbb{R}^{M\times T}$ in which each row and column corresponds to a sensor and a time step respectively. In conventional matrix factorization process, matrix $\boldsymbol{Y}$ which may be incomplete can be approximated by two low rank feature matrices: $\boldsymbol{Y} \approx \boldsymbol{W}\boldsymbol{X}^\top$. $\boldsymbol{W} \in \mathbb{R}^{M\times r}$ is the low rank spatial feature matrix ($r \ll \text{rank}(\boldsymbol{Y})$) and $\boldsymbol{X} \in \mathbb{R}^{T\times r}$ is the low rank temporal feature matrix. Each row vector of $\boldsymbol{W}$ ($\boldsymbol{w}_i \in \mathbb{R}^r, i = 1, 2, \ldots, M$) is the feature vector of its corresponding sensor. And each row vector of $\boldsymbol{X}$ ($\boldsymbol{x}_t \in \mathbb{R}^r, t = 1, 2, \ldots, T$) is the feature vector of its corresponding time step.

Different from conventional matrix factorization models, TRMF models learn and utilize temporal dependency existing in temporal features while performing matrix factorization. In the example in Fig. 2, x_t represents the feature vector at time step t. The

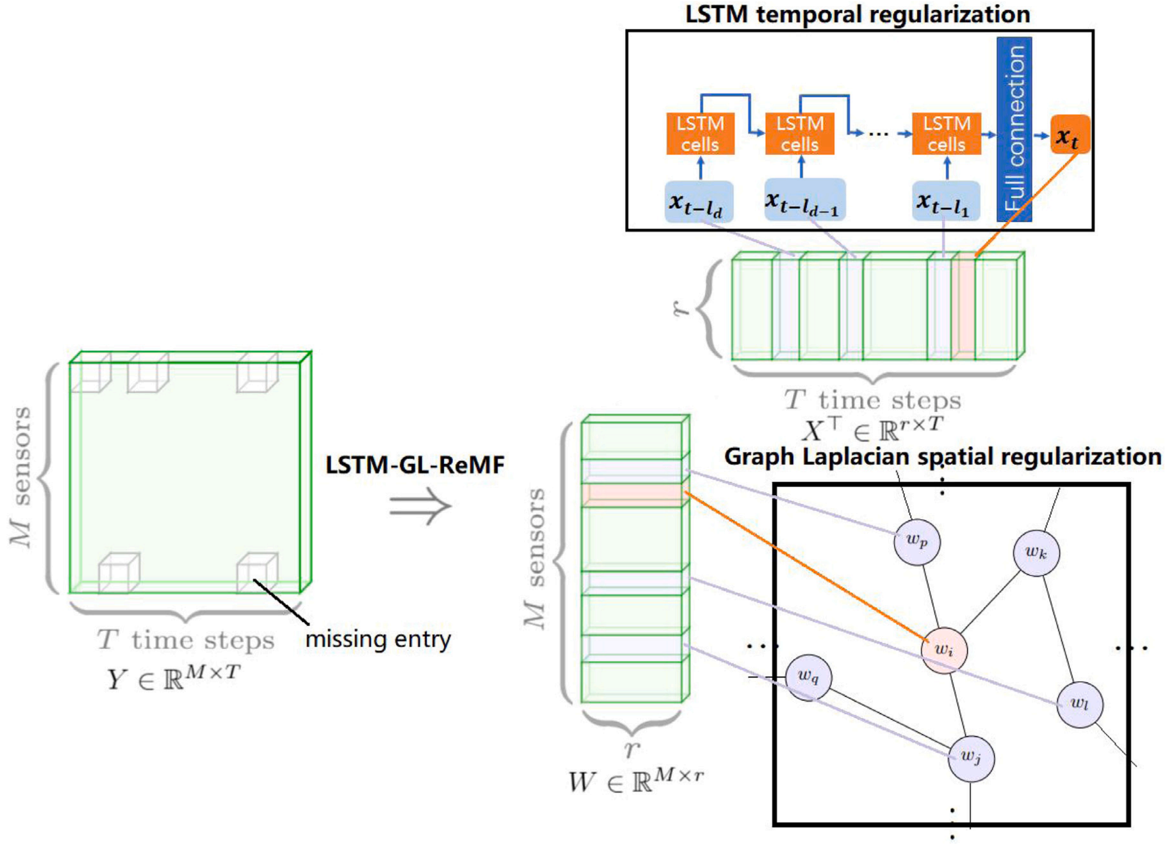


Fig. 3. Illustration of LSTM-GL-ReMF Model.

directed edges indicate that each feature vector is strongly related to the vectors at certain previous time steps. TRMF describes the temporal correlation as a temporal regularizer $x_t \approx f(x_{t-l_1}, x_{t-l_2}, \ldots, x_{t-l_d})$. $\mathcal{L} = \{l_1, l_2, \ldots, l_d\}$ is the time lag set that indicates the temporal correlation topology (e.g., $\mathcal{L} = \{1, 4\}$ in Fig. 2). Temporal regularizer learns the temporal dependence from the factorized temporal feature matrix $\boldsymbol{X}$. In return, the learnt temporal dependence is then used to regularize the matrix factorization process so that better feature matrices can be learned. Autoregressive (AR) model is one of the most commonly used temporal regularizers (Yu et al., 2016).

TRMF can be formulated as follows:

$$\min_{W,X,\theta} \frac{1}{2} \sum_{(i,t)\in\Omega} \left(y_{it} - \boldsymbol{w}_i^\top \boldsymbol{x}_t\right)^2 + \frac{\lambda_w \eta}{2} \|\boldsymbol{W}\|_F^2 + \frac{\lambda_x \eta}{2} \|\boldsymbol{X}\|_F^2 + \lambda_x \mathcal{R}_t (X \mid \theta) \tag{1}$$

where the first term is the sum of squared residual error, $\| \ \|_F$ is the Frobenius norm for over-fitting prevention, Ω is the index set of observed entries, $\mathcal{R}_t (X \mid \theta)$ is the temporal regularizer such as AR, and θ represents the learnable temporal regularization parameters such as AR coefficients. λ_w, λ_x and η are preset regularization coefficients which indicate the importance of Frobenius norm and temporal regularizers.

3.2. LSTM and Graph Laplacian regularized matrix factorization

On the basis of TRMF, we propose a novel LSTM and Graph Laplacian regularized matrix factorization (LSTM-GL-ReMF). The LSTM network temporal regularizer can capture the non-linear temporal dependency, and the GL spatial regularizer (Cai et al., 2010) is able to deal with spatial dependency among adjacency sensors. The schematic diagram of the proposed LSTM-GL-ReMF is shown in Fig. 3. Like TRMF, the data matrix $\boldsymbol{Y} \in \mathbb{R}^{M\times T}$ is factorized into temporal feature matrix $\boldsymbol{X} \in \mathbb{R}^{T\times r}$ and spatial feature matrix $\boldsymbol{W} \in \mathbb{R}^{M\times r}$. Furthermore, LSTM-GL-ReMF replaces the AR temporal regularizer in TRMF with the LSTM network temporal regularizer and also includes an GL spatial regularizer.

In Fig. 3, the temporal regularizer consists of one LSTM layer with r LSTM hidden units in each cell and one fully connected layer with r units. It takes historical temporal feature vectors $\boldsymbol{x}_{t-l_1}, \boldsymbol{x}_{t-l_2}, \ldots, \boldsymbol{x}_{t-l_d}$ as input and make a forecast of $\boldsymbol{x}_t' = f_\theta(\boldsymbol{x}_{t-l_1}, \ldots, \boldsymbol{x}_{t-l_d})$,

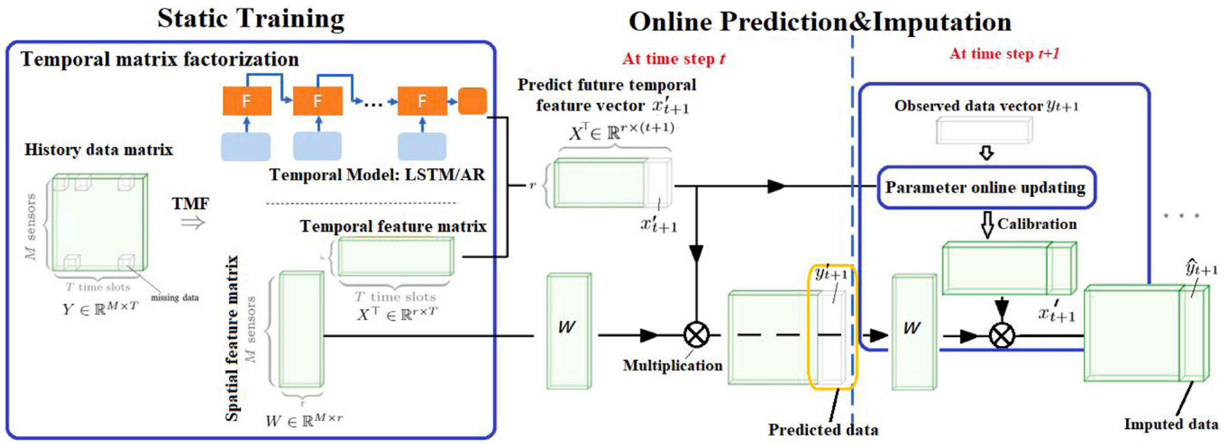

Static Training
Temporal matrix factorization
F
F
F
Temporal Model: LSTM/AR
History data matrix
M sensors
T time slots
missing data
$Y \in \mathbb{R}^{M\times T}$
TMF
Temporal feature matrix
T time slots
$X^\top \in \mathbb{R}^{r\times T}$
Spatial feature matrix
M sensors
r
$W \in \mathbb{R}^{M\times r}$
Online Prediction&Imputation
At time step t
Predict future temporal feature vector x'_{t+1}
$X^\top \in \mathbb{R}^{r\times(t+1)}$
r
x'_{t+1}
W
Multiplication
y'_{t+1}
Predicted data
At time step t+1
Observed data vector y_{t+1}
Parameter online updating
Calibration
x'_{t+1}
W
$\hat{y}_{t+1}$
Imputed data

Fig. 4. Illustration of the online prediction and imputation framework.

where $f_\theta(\,)$ denotes the LSTM network, θ denotes the trainable weights and biases in the network, $\mathcal{L} = \{l_1, l_2, \dots, l_d\}$ is the time lag set indicating the temporal correlation topology. The temporal regularization can be formulated as follows:

$$\mathcal{R}_t(\boldsymbol{X}|\theta) = \frac{1}{2}\sum_{t=l_d+1}^{T} \|\boldsymbol{x}_t - \boldsymbol{x}'_t\|_2^2 \tag{2}$$

To incorporate local spatial dependence information, we implement Graph Laplacian (GL) (Rao et al., 2015; Cai et al., 2010) as spatial regularizer. Graph Laplacian spatial regularizer penalizes the discrepancies between spatial feature vectors and their neighbors in spatial topology. For example, spatial feature vector $\boldsymbol{w}_i$ is adjacent with $\boldsymbol{w}_p$, $\boldsymbol{w}_k$ and $\boldsymbol{w}_j$ as shown in Fig. 3. GL would minimize the differences between spatial feature vector $\boldsymbol{w}_i$ and $\boldsymbol{w}_p$, $\boldsymbol{w}_k$, $\boldsymbol{w}_j$ respectively. If spatiotemporal data has undirected spatial topology graph $G = (V^w, E^w)$, the GL spatial regularizer can be formulated as:

$$\mathcal{R}_s = tr(\boldsymbol{W}^\top \mathbf{Lap}(E^w)\boldsymbol{W}) = \frac{1}{2}\sum_{i,j} E^w_{i,j}\|\boldsymbol{w}_i - \boldsymbol{w}_j\|_2^2 \tag{3}$$

where $\mathbf{Lap}(E^w) = \boldsymbol{D}^w - E^w$ is the graph Laplacian, $\boldsymbol{D}^w$ is the degree diagonal matrix of vertices V^w, and $E^w_{i,j}$ is the weight on edge (i,j) connecting sensor i and sensor j.

The LSTM-GL-ReMF can be formulated as follows:

$$\begin{aligned}\min_{W,X,\theta} \frac{1}{2}\sum_{(i,t)\in\Omega}\left(y_{it} - \boldsymbol{w}_i^\top \boldsymbol{x}_t\right)^2 + \frac{\lambda_w \eta}{2}\|\boldsymbol{W}\|_F^2 + \frac{\lambda_x \eta}{2}\|\boldsymbol{X}\|_F^2 \\ + \frac{\lambda_w}{2}\sum_{i,j} E^w_{i,j}\|\boldsymbol{w}_i - \boldsymbol{w}_j\|_2^2 + \frac{\lambda_x}{2}\sum_{t=l_d+1}^{T}\|\boldsymbol{x}_t - f_\theta(\boldsymbol{x}_{t-l_1}, \dots, \boldsymbol{x}_{t-l_d})\|_2^2\end{aligned} \tag{4}$$

where the first term is the sum of squared residual error, the second and the third term are $L2$ penalties for over-fitting prevention, the fourth term is the graph Laplacian spatial regularizer and the last term is the LSTM temporal regularizer. Ω denotes the index set of observed entries. Hyper-parameters λ_w, λ_x and η are regularization coefficients which balance the importance of spatial and temporal regularizations and their corresponding $L2$ penalties.

3.3. Online prediction and imputation framework

This section introduces the online spatiotemporal data prediction and imputation framework. Traditional MF frameworks either focus on imputing missing values in the history data matrix $Y \in \mathbb{R}^{m\times T}$ or focus on making rolling prediction of future data vectors $y'_{t+1} \in \mathbb{R}^m, t+1 > T$ separately. The proposed online framework combines the two tasks on an online basis, that is, making prediction of data vector y'_{t+1} of the next time step and imputing the possible missing entries in the newly observed data vector $y_t \in \mathbb{R}^m$ simultaneously. The online data imputation process is not simply using the previously predicted values to fill in the data blanks, but using data observed at the current time step t to estimate the rest in y_t with no observation. This online framework can be applied to various kinds of temporal matrix factorization models including the proposed LSTM-GL-ReMF model and traditional temporal MF models (Yu et al., 2016; Gultekin and Paisley, 2019). As shown in Fig. 4, the framework basically consists of two steps: static training and dynamic prediction and imputation.

3.3.1. Alternating method for static training

The static training step decouples historical spatiotemporal data matrix as spatial and temporal feature matrices based on the proposed LSTM-GL-ReMF model. Due to the introducing of the LSTM regularizer, Eq. (4) does not have a tractable closed form solution. To address this issue, we propose an alternating algorithm to optimize feature matrices and neural network parameters iteratively and compare it with fully back-propagation in section Section 4.6. In the proposed alternating method, each iteration consists of two steps: update feature matrix $\boldsymbol{X}$ and $\boldsymbol{W}$ using vector based alternative least square method, and then update the LSTM network parameters θ using Adam through back-propagation (BP).

Keep the LSTM network parameters θ fixed, calculate $\boldsymbol{x}'_t = f_\theta(\boldsymbol{x}_{t-l_1}, \boldsymbol{x}_{t-l_2}, \dots, \boldsymbol{x}_{t-l_d})$ for $t = \{l_d+1, \dots, T\}$, the first optimization step is formulated as:

$$\min_{\boldsymbol{W},\boldsymbol{X}} \frac{1}{2} \sum_{(i,t)\in\Omega} \left(y_{it} - \boldsymbol{w}_i^\top \boldsymbol{x}_t\right)^2 + \frac{\lambda_w \eta}{2} \|W\|_F^2 + \frac{\lambda_x \eta}{2} \|X\|_F^2 + \frac{\lambda_w}{2} \sum_{i,j} E_{i,j}^w \|\boldsymbol{w}_i - \boldsymbol{w}_j\|_2^2 + \frac{\lambda_x}{2} \sum_{t=l_d+1}^{T} \|\boldsymbol{x}_t - \boldsymbol{x}'_t\|_2^2 \tag{5}$$

Eq. (5) can be solved using alternative least squares (ALS) method as used in Yu et al. (2016) by making the first order derivative of the objective function with respect to each feature vector equals zero. Assume at iteration p, we can update the relevant parameters as following:

Updates for spatial feature vectors $\boldsymbol{w}_i, i = 1, 2, \dots, M$:

$$\boldsymbol{w}_i^{(p+1)} = \left(\sum_{t:(i,t)\in\Omega} \boldsymbol{x}_t^{(p)} \boldsymbol{x}_t^{(p)\top} + \lambda_w \eta \boldsymbol{I} + \lambda_w \sum_j E_{i,j}^w \boldsymbol{I}\right)^{-1} \left(\sum_{t:(i,t)\in\Omega} y_{it} \boldsymbol{x}_t^{(p)} + \lambda_w \sum_j E_{i,j}^w \boldsymbol{w}_j^{(p)}\right) \tag{6}$$

Updates for temporal feature vectors $\boldsymbol{x}_t, t = 1, 2, \dots, l_d$:

$$\boldsymbol{x}_t^{(p+1)} = \left(\sum_{i:(i,t)\in\Omega} \boldsymbol{w}_i^{(p+1)} \boldsymbol{w}_i^{(p+1)\top} + \lambda_x \eta \boldsymbol{I}\right)^{-1} \left(\sum_{i:(i,t)\in\Omega} y_{it} \boldsymbol{w}_i^{(p+1)}\right) \tag{7}$$

Updates for temporal vectors $\boldsymbol{x}_t, t = l_d+1, l_d+2, \dots, T$:

$$\boldsymbol{x}_t^{(p+1)} = \left(\sum_{i:(i,t)\in\Omega} \boldsymbol{w}_i^{(p+1)} \boldsymbol{w}_i^{(p+1)\top} + \lambda_x \eta \boldsymbol{I} + \lambda_x \boldsymbol{I}\right)^{-1} \left(\sum_{i:(i,t)\in\Omega} y_{it} \boldsymbol{w}_i^{(p+1)} + \lambda_x \boldsymbol{x}_t'^{(p+1)}\right) \tag{8}$$

where $\boldsymbol{x}_t'^{(p+1)} = f_{\theta^{(p)}}(\boldsymbol{x}_{t-l_1}^{(p+1)}, \boldsymbol{x}_{t-l_2}^{(p+1)}, \dots, \boldsymbol{x}_{t-l_d}^{(p+1)})$.

The second step to optimize LSTM network parameters is as follows. The weights and biases in the LSTM regularizer network can be updated through back propagation by batch gradient descent (Hochreiter and Schmidhuber, 1997).

$$\min_{\theta^{(p+1)}} \sum_{t=l_d+1}^{T} \|\boldsymbol{x}_t^{(p+1)} - f_{\theta^{(p+1)}}(\boldsymbol{x}_{t-l_1}^{(p+1)}, \dots, \boldsymbol{x}_{t-l_d}^{(p+1)})\|_2^2 \tag{9}$$

The algorithm conducts two steps iteratively until it reaches the following convergence criteria:

$$C = \frac{\|\boldsymbol{W}^{(p+1)} \boldsymbol{X}^{(p+1)^\top} - \boldsymbol{W}^{(p)} \boldsymbol{X}^{(p)^\top}\|_F^2}{\|\boldsymbol{W}^{(p)} \boldsymbol{X}^{(p)^\top}\|_F^2} \le \epsilon \tag{10}$$

where the super scripts p and $p+1$ indicate the iteration indices, and ϵ is set to be 1×10^{-4} in this study.

3.3.2. Online prediction and imputation

In the online prediction and imputation part in Fig. 4, at time step t, the framework make prediction of future temporal feature vector $\boldsymbol{x}'_{t+1}$ using the proposed LSTM network temporal regularizer by:

$$\boldsymbol{x}'_{t+1} = f_\theta(\boldsymbol{x}_{t+1-l_1}, \boldsymbol{x}_{t+1-l_2}, \dots, \boldsymbol{x}_{t+1-l_d}) \tag{11}$$

where $f_\theta()$ denotes the LSTM network temporal regularizer network with parameter set θ. Spatiotemporal prediction of time step $t+1$ can then be made by combining the predicted temporal feature vector $\boldsymbol{x}'_{t+1}$ and spatial feature matrix $\boldsymbol{W}$ as:

$$\boldsymbol{y}'_{t+1} = \boldsymbol{W} \boldsymbol{x}'_{t+1} \tag{12}$$

Once reaching time step $t+1$, parameters can be updated using the newly observed data vector y_{t+1}. Instead of concatenating the current observation data vector y_t into the history data matrix (or a sub-matrix) and then factorizes the extended matrix to update model parameters as does in TRMF (Yu et al., 2016) and BTMF (Sun and Chen, 2019), in this framework, the spatial feature matrix W and LSTM parameters θ are fixed, and the temporal feature vector x_{t+1} can be updated using the current observation data vector y_{t+1} only which greatly accelerates the parameter online updating process. The online parameter updating problem is formulated as

Eq. (13). Spatial feature matrix $\boldsymbol{W}$ and the LSTM network temporal regularizer parameters θ are fixed in this online temporal MF operation. The online LSTM regularized MF operation is formulated as follows:

$$\min_{\boldsymbol{x}_{t+1}} \frac{1}{2} \sum_{i \in \Omega_{t+1}} \left(y_{i,t+1} - \boldsymbol{w}_i^\top \boldsymbol{x}_{t+1}\right)^2 + \frac{\lambda_x}{2} \left(\boldsymbol{x}_{t+1} - \boldsymbol{x}'_{t+1}\right)^\top \left(\boldsymbol{x}_{t+1} - \boldsymbol{x}'_{t+1}\right) + \frac{\lambda_x \eta}{2} \boldsymbol{x}_{t+1}^\top \boldsymbol{x}_{t+1} \tag{13}$$

where Ω_{t+1} is the index set of non-missing entries in the data vector $\boldsymbol{y}_{t+1}$.

Solve Eq. (13) by the least square method and derive the updating equation for temporal feature vector $\boldsymbol{x}_{t+1}$ as:

$$\boldsymbol{x}_{t+1} = \left(\sum_{i \in \Omega_{t+1}} \boldsymbol{w}_i \boldsymbol{w}_i^\top + \lambda_x \boldsymbol{I} + \lambda_x \eta \boldsymbol{I}\right)^{-1} \left(\sum_{i \in \Omega_{t+1}} y_{i,t+1} \boldsymbol{w}_i + \lambda_x \boldsymbol{x}'_{t+1}\right) \tag{14}$$

The calibrated temporal feature vector $\boldsymbol{x}_{t+1}$ will then be used for future temporal feature vector prediction by Eq. (11). If real-time observation data vectors have missing values, online data imputation can be made by combining the calibrated temporal feature vector $\boldsymbol{x}_{t+1}$ with the statically trained spatial feature matrix $\boldsymbol{W}$ by:

$$\hat{\boldsymbol{y}}_{t+1} = \boldsymbol{W} \boldsymbol{x}_{t+1} \tag{15}$$

where the missing entries in $\boldsymbol{y}_{t+1}$ can be filled by the ones in $\hat{\boldsymbol{y}}_{t+1}$.

The online prediction and imputation framework can be easily extended to other spatial temporal regularized matrix and tensor factorization models like TRMF (Yu et al., 2016) and OLMF (Gultekin and Paisley, 2019). The proposed MF model can also be further extended for higher dimensional spatiotemporal data following the tensor CP decomposition method (Hitchcock, 1927) by replacing the feature vector updating Eqs. (6)–(8) as shown in Appendix A.

4. Experiments

To demonstrate the performance of the proposed framework and LSTM-GL-ReMF model, in this section, we conducted online prediction and imputation experiments on three spatiotemporal traffic datasets, namely the Seattle Traffic Speed dataset, Metr-LA Freeway Speed dataset (Jagadish et al., 2014) and a multi-variate traffic dataset PeMSD8 (Guo et al., 2019). The source code and datasets of our experiments can be found at https://github.com/Vadermit/TransPAI.

4.1. Experiment setting

We evaluate our LSTM-GL-ReMF model and model variants, traditional MF/TF models and state-of-the-art deep learning models on the three datasets under various data missing scenarios. We simulate two data missing patterns as observed in real-world, one is point-wise missing (**PM**) where data is missing at discontinuous time steps and the other is continuous missing (**CM**) where data is missing for a continuous period of time. We then randomly select entries from the dataset following **PM** or **CM** and set them to zero at a certain missing rate. The last 30% percent of data are selected as test set and the rest are used for model training. For deep learning models in which model over-fitting is critical, 10% of the training data are further extracted as validation set to early stop the model. While MF models can avoid over-fitting simply by setting a relatively small rank of feature matrices so no validation is used in MF models.

In the online prediction and imputation experiments, the training sets are used to train model parameters. And the test set data is fed in sequential at each time step to simulate real-time data observation. At each time step, after obtaining the observation data, the model predicts the data of the next time step, and patches missing values in the possibly incomplete real-time observation data. Root mean square error (RMSE) and mean absolute percentage error (MAPE) are used to evaluate prediction and imputation accuracy.

$$\text{MAPE} = \frac{1}{n(\Omega_e)} \sum_{i \in \Omega_e} \left| \frac{y_i - \hat{y}_i}{y_i} \right|, \quad \text{RMSE} = \sqrt{\frac{1}{n(\Omega_e)} \sum_{i \in \Omega_e} \left(y_i - \hat{y}_i\right)^2}. \tag{16}$$

$n(\Omega_e)$ is the size of the index set Ω_e. For prediction performance evaluation, Ω_e denotes the index set of non-missing entries in the ground truth (the Metr-LA speed dataset has 8.11% data missing initially). While for imputation performance evaluation, Ω_e denotes the index set of the manually masked entries.

4.2. Baseline models

To highlight the superiority of the framework and the LSTM-GL-ReMF model proposed in this paper, we choose the following state-of-the-art models as baseline models.

- TRMF: Temporal regularized matrix factorization (Yu et al., 2016) implements AR temporal regularization in the matrix factorization process. Two model variants: TRMF-GRMF and TRMF-ALS are tested in this experiment. TRMF-GRMF solves the MF problem by GRMF algorithm (Yu et al., 2016) which transform the AR regularization to a Graph Laplacian first and then update each feature sub-matrix as a whole. In the online phrase, TRMF-GRMF factorizes a small data matrix to update parameters at each prediction step. TRMF-ALS implements similar vector based least square methods as proposed in this study for parameter static and online updating.

- BTMF/BTTF: Bayesian temporal matrix/tensor factorization (Sun and Chen, 2019) is the Bayesian probability version of TRMF. Parameters of BTMF/BTTF are be obtained from their inferred posterior distribution by Gibbs sampling. Instead of the vector AR used by TRMF, BTMF implements matrix AR which enables it to capture more complex temporal correlations. Similar to TRMF-GRMF, in the online stage, BTMF factorizes a small data matrix/tensor at each time step to update parameters once observations come in. The small data matrix/tensor used in the online prediction and imputation process of TRMF-GRMF and BTMF consists of data from the last two days from the current time step.
- LSTM: Long short-term memory (Hochreiter and Schmidhuber, 1997) is a kind of recurrent neural network designed for sequential data that is able to learn long term dependencies.
- GRU-D: A deep learning model based on Gated Recurrent Unit (Che et al., 2018), it is a kind of recurrent neural network designed for time series with missing values by taking the interval between two observations into consideration. It not only captures the temporal dynamics but also data missing patterns to achieve better prediction performance.
- SGMN: Spectral Graph Markov Network is a deep learning model designed to make traffic forecasting with missing data. It considers the traffic network as a graph and define the transition between network-wide traffic states at consecutive time steps as a graph Markov process (Cui et al., 2020). Missing entries are filled in using previously predicted values in the Markov process.
- DCRNN: Diffusion Convolutional Recurrent Neural Network (Li et al., 2017) is a deep learning framework for traffic forecasting that captures the spatial dependency using diffusion convolution, and the temporal dependency using the encoder–decoder architecture.
- GCGRNN: Graph Convolutional Neural Network with Data-driven Graph Filter with recurrent neural network (Lin et al., 2018) is a deep learning model that incorporates graph convolutional neural network and LSTM. It is able to learn the spatial connection topology automatically and no prior graph information is needed.
- STGCN: Spatio-Temporal Graph Convolutional Networks (Yu et al., 2017) is a deep learning framework that implements graph convolution and 1-D gated convolution for spatial and temporal dependencies respectively.
- ASTGCN: Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting (Guo et al., 2019) is a deep learning framework that incorporates both spatio-temporal attention and graph convolutional network to capture spatio-temporal correlations.

4.3. Model variants

In order to verify the effectiveness of each component of the proposed model and to verify its generalization ability to multi-variate prediction, several model variants are tested in the experiments. These model variants are:

- LSTM-ReMF: LSTM-ReMF is a simplified variant of LSTM-GL-ReMF which only implements LSTM as the temporal regularizer and neglects the graph Laplacian spatial regularization.
- BiLSTM-ReMF/BiLSTM-GL-ReMF: Bidirectional LSTM (and Graph Laplacian) regularized matrix factorization implements two LSTM networks to model both the forward and backward temporal dynamics.
- Linear-LSTM-ReMF: To verify the effectiveness of non-linearity in the temporal model LSTM, nonlinear activation functions tanh and sigmoid are removed in Linear-LSTM-ReMF comparing to LSTM-ReMF.
- LSTM-ReTF, LSTM-GL-ReTF: Tensor factorization extensions of LSTM-ReMF and LSTM-GL-ReMF using the tensor CP decomposition method (Hitchcock, 1927).

4.4. Experiments on two uni-variate traffic speed datasets

Two uni-variate traffic speed datasets are tested in this study, namely the Seattle Traffic Speed dataset and the Metr-LA Los Angeles Freeway Speed dataset (Jagadish et al., 2014). The Seattle Traffic Speed dataset consists of speed data collected by 323 loop sensors located in Seattle from 1st Nov, 2015 to 31th Dec, 2015. And the Metr-LA Speed dataset consists of speed data collected from 207 loop sensors located in Los Angeles from 1st Mar, 2012 to 30th April, 2012. The time interval for data collection in these two datasets is 5 min. The spatial connection of sensors are represented by 0–1 adjacency matrices.

The last 30% of data are used for testing and the rest are used for model training. Rank r of feature sub-matrices and LSTM units number are both set to be 60. To be fair, the number of RNN units in deep learning baselines are also set to be 60. Considering the recent temporal dynamics and the strong daily periodicity of traffic data, the data time lag $\mathcal{L}$ is set to be $\{1, 2, 288\}$ except for GRU-D, TGC-LSTM and SGMN. For these three deep learning models, a continuous time lag set of $\{1, 2, \dots, 10\}$ is used since the temporal continuity matters in their model designs. Hyper-parameters, namely the regularization coefficients λ_w, λ_x and η are chosen using grid search with sliding window cross validation. In the hyper-parameter tuning process, the original dataset is partitioned into three non-overlapping sub-datasets, each containing 20 or 21 days' of data. The hyper-parameter set with which the model obtains the minimum average RMSE on all sub-datasets are set to be the final hyper-parameters. We first compare the online prediction and imputation performance of the proposed LSTM-GL-ReMF and its variants with traditional matrix factorization approaches such as TRMF and BTMF.

The online prediction and imputation performances of matrix factorization models on Seattle dataset and Metr-LA dataset are shown in Tables 1 and 2.

Table 1
Online prediction and imputation errors on Seattle speed dataset.

	LSTM-GL-ReMF	LSTM-ReMF	BiLSTM-GL-ReMF	Linear-LSTM-ReMF	TRMF-ALS	TRMF-GRMF	BTMF
Online prediction error: MAPE/RMSE							
Original,	**7.64**/4.43	**7.64**/**4.42**	7.83/4.50	8.01/4.59	8.36/4.96	8.20/4.83	7.70/4.59
10%, PM	7.83/4.51	**7.77**/**4.49**	7.98/4.57	8.18/4.70	8.48/5.03	8.27/4.87	7.80/4.65
20%, PM	7.94/4.60	7.91/4.59	**7.89**/**4.55**	8.35/4.81	9.14/5.38	8.30/4.87	7.97/4.73
40%, PM	8.39/4.81	**8.16**/4.75	8.21/**4.70**	8.58/4.95	9.18/5.43	8.50/4.97	9.07/5.31
10%, CM	7.81/4.52	**7.76**/**4.51**	7.93/4.58	8.07/4.67	8.23/4.92	8.62/5.06	7.80/4.59
20%, CM	8.02/**4.62**	8.00/**4.62**	8.03/4.64	8.06/4.72	8.15/4.89	9.47/6.35	**7.96**/4.63
40%, CM	8.37/4.84	8.37/4.97	**8.23**/**4.83**	8.68/5.04	8.59/5.20	14.53/13.15	8.35/4.88
Online imputation error: MAPE/RMSE							
10%, PM	7.12/4.24	7.46/4.55	7.10/4.24	7.49/4.57	7.48/4.55	**6.29**/**3.83**	7.15/4.33
20%, PM	7.20/4.31	7.61/4.68	7.16/4.29	7.63/4.70	7.63/4.68	**6.48**/**3.93**	7.24/4.38
40%, PM	7.45/4.43	8.01/4.92	7.35/4.38	8.04/4.95	8.01/4.91	**6.95**/**4.16**	7.60/4.50
10%, CM	7.77/**4.78**	7.91/5.02	**7.74**/4.82	7.87/4.98	7.92/5.08	11.50/6.41	7.89/4.77
20%, CM	**7.67**/4.64	7.83/4.87	7.68/4.68	7.85/4.90	7.90/4.95	14.05/10.33	7.73/**4.55**
40%, CM	8.13/4.90	8.61/5.32	**8.11**/**4.89**	8.54/5.30	8.76/5.46	23.51/19.94	8.35/4.92

Best results are bold marked.

Table 2
Online prediction and imputation errors on Metr-LA speed dataset.

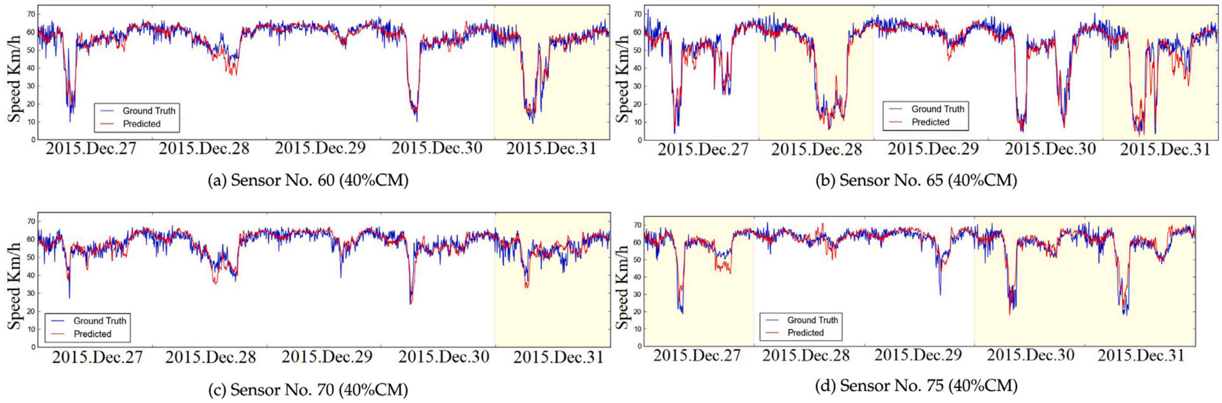
	LSTM-GL-ReMF	LSTM-ReMF	BiLSTM-GL-ReMF	TRMF-ALS	TRMF-GRMF	BTMF
Online prediction error: MAPE/RMSE						
Original,	7.76/4.89	**7.42**/4.79	7.51/**4.75**	8.15/5.09	9.26/5.94	7.97/5.20
10%, PM	7.99/5.01	**7.52**/4.83	7.68/**4.82**	8.21/5.12	9.28/5.96	7.99/5.23
20%, PM	8.13/5.11	**7.63**/**4.87**	7.85/4.97	8.20/5.12	9.39/6.00	8.75/5.62
40%, PM	8.77/5.52	8.15/**5.12**	**8.11**/**5.12**	8.19/5.15	9.72/6.15	10.13/6.37
10%, CM	8.00/5.04	**7.54**/**4.83**	7.70/4.88	8.02/5.06	9.74/6.35	7.85/5.12
20%, CM	8.19/5.16	**7.67**/**4.95**	7.97/5.04	8.05/5.10	10.78/7.53	8.01/5.20
40%, CM	9.07/5.66	**8.36**/**5.40**	8.54/**5.40**	8.61/5.47	16.04/14.10	9.05/5.86
Online imputation error: MAPE/RMSE						
10%, PM	6.93/4.53	6.89/4.51	**6.85**/**4.48**	6.93/4.53	7.12/4.54	7.06/4.60
20%, PM	7.11/4.61	7.05/4.60	**7.02**/**4.58**	7.04/4.60	7.42/4.70	7.33/4.73
40%, PM	7.51/4.85	7.30/4.75	7.32/4.76	**7.20**/**4.70**	8.04/5.06	7.92/5.04
10%, CM	7.29/4.96	**7.20**/4.94	7.21/**4.92**	7.37/5.05	13.77/9.37	7.26/4.87
20%, CM	7.76/**5.16**	**7.65**/5.22	7.70/5.19	7.86/5.30	15.48/11.52	7.98/5.30
40%, CM	8.89/5.75	**8.78**/5.74	8.80/**5.73**	9.11/5.91	25.59/21.15	10.25/6.55

Best results are bold marked.

Results in Tables 1 and 2 demonstrate the superior online prediction and imputation accuracy and stability of the proposed LSTM based MF models over other AR based MF counterparts under most data missing scenarios. (1) Comparing the performance of LSTM-ReMF and TRMF-ALS in which the only difference is the temporal regularizer, LSTM network temporal regularization could greatly improve the online prediction and imputation accuracy. (2) There are a few cases where TRMF-GRMF and BTMF achieve better imputation accuracy than the proposed LSTM regularized models. This may be because that TRMF-GRMF and BTMF utilize much more data in the online prediction & imputation phase than that used in the proposed models and TRMF-ALS. For parameter online updating, TRMF-GRMF and BTMF factorize a data matrix consists of data collected from the last two days at each time step which would take hundreds of times longer to calculate than our models. Detailed computation time costs are listed in Appendix B.

The effectiveness of the modules in the proposed LSTM-GL-ReMF model can be analyzed from the comparisons between several model variants. (1) The better performance of LSTM-ReMF over Linear-LSTM-ReMF implies that the non-linearity of temporal correlations captured by the LSTM network and is essential to improving the prediction performance. (2) Comparing the performance of LSTM-ReMF with LSTM-GL-ReMF, the Graph Laplacian spatial regularizer improves the online imputation accuracy on Seattle dataset but also reduces the online prediction accuracy in most cases. This is probably because the speed of adjacent road segments does not necessarily have such a strong similarity. The Graph Laplacian regularization in LSTM-GL-ReMF could over-averages the data and lead to a performance degradation. This data over-averaging problem can be alleviated by introducing more detailed spatial information such as the distances between sensors as will be introduced in Section 4.5. (3) Comparing to LSTM-GL-ReMF which only models the forward temporal dynamics, BiLSTM-GL-ReMF implements another LSTM for backward temporal dependencies. BiLSTM makes the model more stable when the data missing rate rises up, but it may also over-averaged the data and jeopardize the model performance when the data missing rate is low.

We provide some visualizations of the predictions made by LSTM-GL-ReMF on the Seattle speed dataset with 40% data continuous missing in Fig. 5. The red lines are predicted speed values and the blue ones are the ground truth. The yellow panels are data masked as zero simulating data continuous missing. As can be seen, the red lines and blue lines fit well with each other, which indicates the



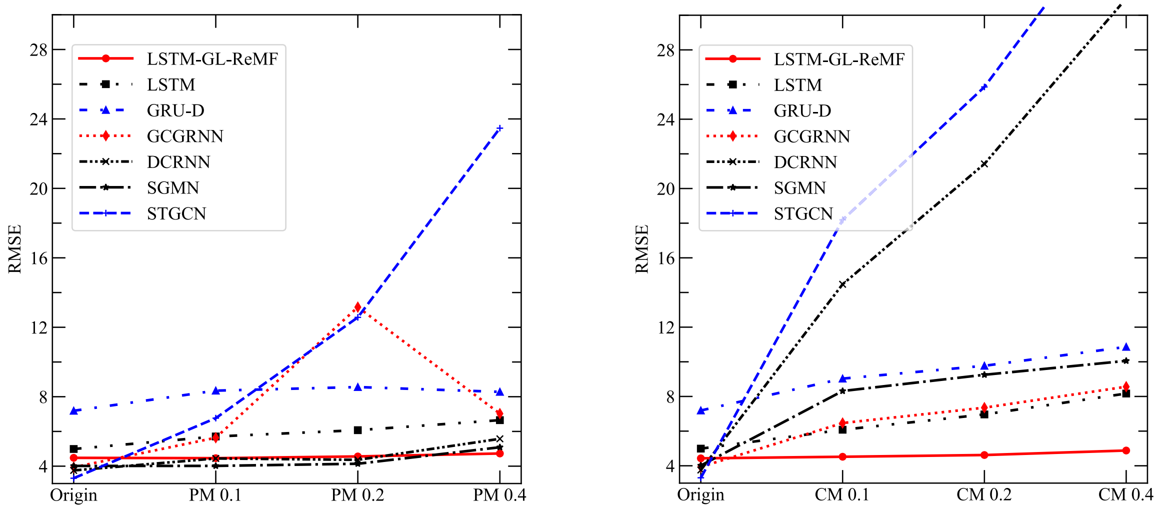
(a) Sensor No. 60 (40%CM)

(b) Sensor No. 65 (40%CM)

(c) Sensor No. 70 (40%CM)

(d) Sensor No. 75 (40%CM)

Fig. 5. Speed prediction made by LSTM-ReMF under 40% CM scenario.



(a) Prediction RMSE under point-wise data missing

(b) Prediction RMSE under continuous data missing

Fig. 6. Prediction error on Seattle Speed dataset.

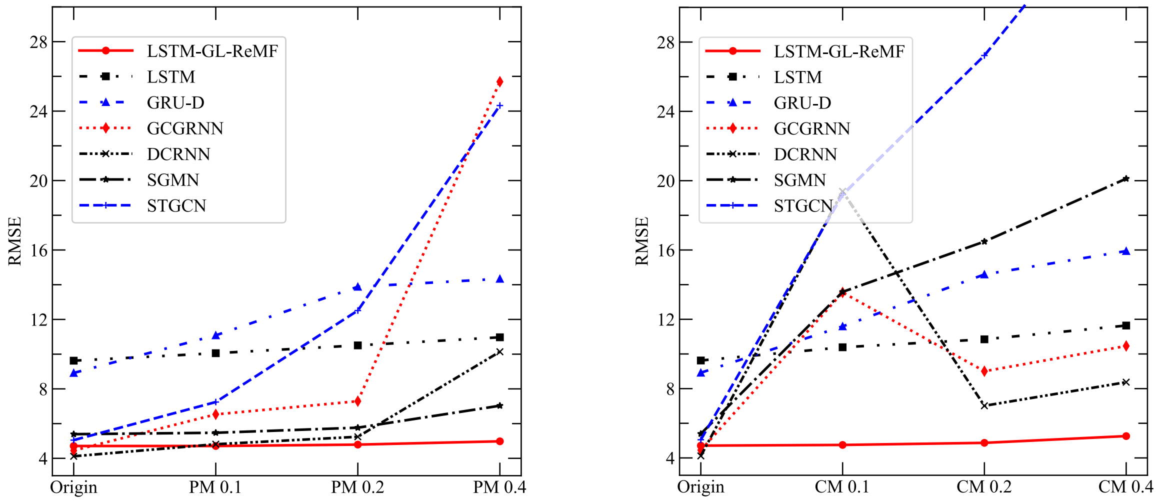
effectiveness of our approach. Even data are missing for continuous days at some sensors, LSTM-GL-ReMF can still correctly predict traffic congestion and sudden speed changes.

Six deep learning models are tested under data point-wise missing (PM) and continuous missing (CM) with missing rate range from 0 to 40%. All models are trained and tested using incomplete data without data imputation preprocessing. The prediction RMSE under different data missing patterns and missing rates are shown in Figs. 6 and 7.

From Fig. 6 it can be seen that: (1) under two data missing patterns, the prediction RMSE of the proposed LSTM-GL-ReMF model remains almost constant at a low level as the data missing rate increased from 0 to 40%; (2) deep learning models such as DCRNN (Li et al., 2017), STGCN (Yu et al., 2017) and SGMN (Cui et al., 2020) achieve better prediction performance when there is no data missing; (3) under the data point-wise missing scenarios, DCRNN and SGMN achieve better prediction performance with data missing rate no higher than 30% on Seattle speed dataset as shown in Fig. 6(a); (4) all six deep learning baselines fail to make proper prediction when data is missing for a continuous long period as demonstrated in Figs. 6(b) and 7(b). However, continuous data missing scenarios are quite common in real-world traffic spatio-temporal datasets as introduced in Section 1. Therefore, compared with the deep learning baselines, the proposed framework with LSTM-GL-ReMF model can more stably adapt to a wider range of real-world spatiotemporal traffic data prediction scenarios.

4.5. Experiments of multi-variate traffic data prediction

To verify the generalization ability of the proposed models to multi-variate spatiotemporal traffic data, we further validate our tensor factorization model variants namely the LSTM (and Graph Laplacian) Regularized Tensor Factorization models (LSTM-GL-ReTF) on PEMSD8 dataset (Guo et al., 2019). PEMSD8 is a multi-variate traffic dataset which consists of traffic flow, speed and



(a) Prediction RMSE under point-wise data missing

(b) Prediction RMSE under continuous data missing

Fig. 7. Prediction RMSE on Metr-LA Freeway Speed dataset.

Table 3
Online prediction and imputation error on PeMSD8 dataset.

	LSTM-GL-ReTF	LSTM-ReTF	BTTF	ASTGCN	DCRNN	GCGRNN
Online prediction error: MAPE/RMSE						
Original,	22.61/17.77	25.04/22.92	30.66/16.46	14.33/31.13	**9.77/14.68**	18.01/23.23
10%, PM	21.08/17.71	25.47/20.35	21.53/16.82	16.91/42.33	**9.12/15.31**	18.72/28.16
20%, PM	24.22/18.19	27.69/18.61	25.91/**16.94**	25.89/78.17	**11.86**/19.99	24.98/37.90
40%, PM	**21.50**/18.29	27.67/20.25	23.43/**17.74**	44.01/136.61	27.24/45.57	41.25/69.86
10%, CM	**27.30**/18.22	23.10/19.24	29.78/**16.91**	29.29/79.10	28.84/62.64	30.69/40.45
20%, CM	**23.41**/18.74	24.70/22.57	28.91/**18.24**	34.46/99.27	29.51/106.08	34.01/48.87
40%, CM	**21.23/20.80**	26.05/27.79	25.37/22.40	49.40/151.03	45.45/100.64	48.80/79.98
Online imputation error: MAPE/RMSE						
10%, PM	**21.43/18.01**	23.99/18.25	22.48/20.47	–	–	–
20%, PM	**23.45/18.39**	27.27/18.62	27.39/20.74	–	–	–
40%, PM	**21.87/19.46**	27.17/19.91	25.16/21.80	–	–	–
10%, CM	19.80/**20.08**	**17.76**/20.61	22.54/21.00	–	–	–
20%, CM	**19.16/22.17**	19.98/24.20	23.54/23.67	–	–	–
40%, CM	**18.74/23.53**	21.12/28.16	22.44/28.22	–	–	–

Best results are bold marked.

occupancy data collected by 170 detectors in San Bernardino in August 2016. The time interval for data collection is also 5 min. Spatial connection weight E^u_{ik} for adjacent sensor i and sensor k is calculated according to their road distance D_{ik} by Eq. (17) as introduced in Yu et al. (2017). In our experiment, we set $\sigma = \sqrt{5} \times \text{std}(D)$ and $\epsilon = 0.5$.

$$E^u_{ik} = \begin{cases} \exp(-\frac{D^2_{ik}}{\sigma^2}), i \neq k \text{ and } \exp(-\frac{D^2_{ik}}{\sigma^2}) > \epsilon \\ 0, \text{otherwise} \end{cases} \tag{17}$$

The dataset is organized in 3-D tensor $\mathcal{Y} \in \mathbb{R}^{170\times3\times8928}$. Data of the first 20 days are used as training set and the last 10 days are used for testing. Ranks of feature sub-matrices and neural network hidden layer units numbers are set to be 20, time lag set $\mathcal{L}$ is set to be $\{1, 2, 288\}$. One AR regularized tensor factorization base baseline BTTF (Sun and Chen, 2019) and three multi-variate graph based deep learning baselines: ASTGCN (Guo et al., 2019), DCRNN (Li et al., 2017) and GCGRNN (Lin et al., 2018) are tested on this dataset. For deep learning baselines, data normalization is performed separately for traffic flow, speed and occupancy data. While for tensor factorization models, no data preprocessing is performed. The online prediction and imputation results are shown in Table 3.

The value scale of the three kinds of data in the PEMSD8 dataset differ greatly. The flow value is up to thousands, the speed value is no more than 100, while the occupancy value is decimal between 0 and 1. Errors of small value data such as occupancy could be neglected in the overall RMSE. So in PeMSD8 dataset, the mean absolute percentage error (MAPE) can better reflect the

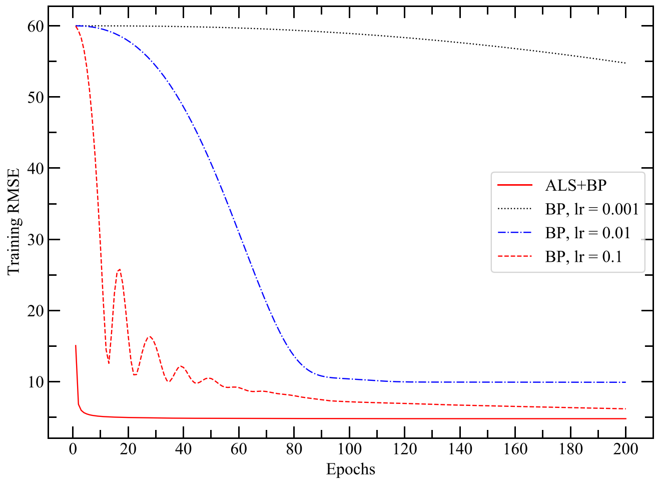


Fig. 8. Static training loss of LSTM-GL-ReMF on Metr-LA with 20% data point-wise missing.

overall prediction performance on PeMSD8 dataset. From Table 3 it can be seen that: (1) LSTM-GL-ReTF outperforms LSTM-ReTF in the prediction task on PeMSD8 dataset with sensor distance information incorporated in the spatial connection weight matrix $\boldsymbol{E}^u$; (2) LSTM-GL-ReTF acquires lower prediction and imputation MAPE than its AR regularized TF counterpart BTTF (Sun and Chen, 2019); (3) Under data point-wise missing scenarios, deep learning model DCRNN (Li et al., 2017) has better prediction performance than other models. However, as the missing rate of data increases, its prediction error increases rapidly. Under data continuous missing scenarios, all three multi-variate deep learning models fail to make proper predictions.

According to the three experiments, the proposed LSTM regularized MF/TF models outperforms the AR regularized MF/TF and deep learning counterparts under most data missing scenarios, especially when data is lost for continuous hours or days. With detailed spatial information such as sensor distances, Graph Laplacian spatial regularization could benefit the prediction and imputation performance. Bidirectional LSTM could stabilize the model performance as data missing rate increases.

4.6. Comparison of the alternating method with fully back-propagation

An alternating method is proposed to solve the LSTM and Graph Laplacian regularized matrix factorization problem (4) by performing ALS and back-propagation (BP) iteratively instead of using fully BP. The two-step alternating method updates the factor matrices by ALS first and then updates LSTM network parameters by gradient descent through back-propagation in each iteration. To demonstrate that the proposed ALS + BP alternating method is superior than fully back-propagation for the LSTM and Graph Laplacian regularized MF problem, we compare the convergence speed and the local minimum acquired by these two methods on Metr-LA speed dataset. Fig. 8 shows the training loss of the LSTM-GL-ReMF model using two solving methods on Metr-LA dataset with 20% data point-wise missing. The abscissa is the training epoch, and the ordinate is the MF RMSE. It can be seen that: (1) the training process converges within a few epochs using the proposed ALS+BP alternating method, while model training by fully back-propagation converges much slower with various learning rates; (2) the local minimum acquired by the proposed alternating method is better than that obtained by fully back-propagation. The proposed ALS+BP alternating method is more effective and efficient than fully BP on the LSTM and Graph Laplacian regularized matrix factorization problem.

5. Conclusion and future directions

In this paper, we propose an online prediction and imputation framework that implements a novel LSTM and Graph Laplacian regularized matrix factorization model (LSTM-GL-ReMF). The proposed framework can make spatiotemporal prediction based on incomplete observation data and impute missing values at the same time. Our LSTM-GL-ReMF model preserves both spatial smoothness and non-linear temporal dynamics of data to enhance prediction and imputation performances.

Through numerical experiments on three spatiotemporal datasets, we show the proposed framework with the LSTM-GL-ReMF model and its variants achieve better prediction and imputation accuracy and stability under various missing patterns and missing rates, especially when data is missing for a continuous period of time, comparing to traditional matrix factorization models and state-of-the-art deep learning models. Deep learning models such as DCRNN (Li et al., 2017; Cui et al., 2020) can maintain a relatively good performance under data point-wise missing scenarios with missing rate no higher than 30%. However, all deep learning baselines fail to make proper prediction under data continuous missing scenarios which are quite general to real-world traffic datasets. Numerical experiments also reveal that the proposed ALS+BP alternating method is more effective and more efficient to solve the LSTM and Graph regularized MF problem comparing to back-propagation only.

The proposed models can be applied to any uni-variate or multi-variate spatio-temporal data with prior spatial knowledge. For spatially locally smooth data, Graph Laplacian could benefit the online prediction and imputation performance. Bidirectional LSTM further stabilizes the model when data missing rate is high (over 10%). For spatially less smooth data or more complete data, Graph Laplacian and Bidirectional LSTM could over-average the data and benefit the data imputation accuracy but also jeopardize the prediction performance. In that case, if the online prediction accuracy matters most, a light model variant LSTM-ReMF/TF is preferable.

For future research directions, we would like to explore matrix factorization approaches that can generate feature matrices with various ranks instead of a universal rank. This can prevent over-fitting or under-fitting for certain feature matrices. It is also interesting to explore other spatial regularizers such as Total Variation (He et al., 2015; Xin et al., 2016) and Directed Auto-Regression (Takeuchi et al., 2017) for spatial dependency capture. Last, we plan to test the proposed framework and LSTM-GL-ReMF model on more spatiotemporal datasets.

CRediT authorship contribution statement

Jin-Ming Yang: Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Writing - original draft, Writing - review & editing. **Zhong-Ren Peng:** Conception and design of study, Acquisition of data, Writing - original draft. **Lei Lin:** Conception and design of study, Analysis and/or interpretation of data, Writing - original draft.

Acknowledgment

This study was partially supported by the National Planning Office of Philosophy and Social Science, China (No. 16ZDA048).

All authors approved the version of the manuscript to be published.

Appendix A. Multi-variate extension: LSTM and Graph Laplacian regularized tensor factorization

The proposed LSTM and Graph Laplacian regularized matrix factorization model can be extend to a tensor factorization model using tensor CP decomposition method (Hitchcock, 1927). Take third order tensor $\mathcal{Y} \in \mathbb{R}^{M\times N\times T}$ for example, initial data tensor $\mathcal{Y}$ can be approximated by three rank R sub-matrices namely $\boldsymbol{U} \in \mathbb{R}^{M\times R}$, $\boldsymbol{V} \in \mathbb{R}^{N\times R}$ and $\boldsymbol{X} \in \mathbb{R}^{T\times R}$ following $\mathcal{Y}_{ijt} \approx \sum_{r=1}^{R} u_{ir}v_{jr}x_{tr}$. Let $\boldsymbol{U}$ denote the spatial feature matrix and $\boldsymbol{X}$ denote the temporal feature matrix, the LSTM-GL-ReTF problem can be formulated as Eq. (A.1).

$$\begin{aligned}\min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{X},\theta} \quad & \sum_{(i,j,t)\in\Omega}\left(y_{ijt}-\sum_{r=1}^{R}u_{ir}v_{jr}x_{tr}\right)^2+\frac{\lambda_u\eta}{2}\sum_{i=1}^{M}\|\boldsymbol{u}_i\|_2^2+\frac{\lambda_v\eta}{2}\sum_{j=1}^{N}\left\|\boldsymbol{v}_j\right\|_2^2+ \\ \frac{\lambda_x\eta}{2}\sum_{t=1}^{T}\|\boldsymbol{x}_t\|_2^2+\frac{\lambda_u}{2}\sum_{i,k}E^u_{i,k}\|\boldsymbol{u}_i-\boldsymbol{u}_k\|_2^2+\frac{\lambda_x}{2}&\sum_{t=l_d+1}^{T}\|\boldsymbol{x}_t-f_\theta(\boldsymbol{x}_{t-l_1},\dots,\boldsymbol{x}_{t-l_d})\|_2^2\end{aligned} \tag{A.1}$$

Similarly, the LSTM and Graph Laplacian regularized tensor factorization model can be solved by the alternating method. First, feature matrices $\boldsymbol{U}$, $\boldsymbol{V}$ and $\boldsymbol{X}$ can be updated by ALS in a vector basis as follows:

Updates for spatial feature vectors $\boldsymbol{u}_i, i = 1,2,\dots,M$:

$$\boldsymbol{u}_i^{(p+1)}=\left(\sum_{j,t:(i,j,t)\in\Omega}\left(\boldsymbol{v}_j^{(p)}\odot\boldsymbol{x}_t^{(p)}\right)\left(\boldsymbol{v}_j^{(p)}\odot\boldsymbol{x}_t^{(p)}\right)^\top+\lambda_u\eta\boldsymbol{I}+\lambda_u\sum_k E^u_{i,k}\boldsymbol{I}\right)^{-1}\left(\sum_{j,t:(i,j,t)\in\Omega}\left(\boldsymbol{v}_j^{(p)}\odot\boldsymbol{x}_t^{(p)}\right)y_{ijt}+\lambda_u\sum_k E^u_{i,k}\boldsymbol{u}_k^{(p)}\right)$$

Updates for data feature vectors $\boldsymbol{v}_j, j = 1,2,\dots,N$:

$$\boldsymbol{v}_j^{(p+1)}=\left(\sum_{i,t:(i,j,t)\in\Omega}\left(\boldsymbol{u}_i^{(p+1)}\odot\boldsymbol{x}_t^{(p)}\right)\left(\boldsymbol{u}_i^{(p+1)}\odot\boldsymbol{x}_t^{(p)}\right)^\top+\lambda_v\eta\boldsymbol{I}\right)^{-1}\sum_{i,t:(i,j,t)\in\Omega}\left(\boldsymbol{u}_i^{(p+1)}\odot\boldsymbol{x}_t^{(p)}\right)y_{ijt}$$

Updates for temporal feature vectors $\boldsymbol{x}_t, t = 1,2,\dots,l_d$:

$$\boldsymbol{x}_t^{(p+1)}=\left(\sum_{i,j:(i,j,t)\in\Omega}\left(\boldsymbol{u}_i^{(p+1)}\odot\boldsymbol{v}_j^{(p+1)}\right)\left(\boldsymbol{u}_i^{(p+1)}\odot\boldsymbol{v}_j^{(p+1)}\right)^\top+\lambda_x\eta\boldsymbol{I}\right)^{-1}\sum_{i,j:(i,j,t)\in\Omega}\left(\boldsymbol{u}_i^{(p+1)}\odot\boldsymbol{v}_j^{(p+1)}\right)y_{ijt}$$

Updates for temporal feature vectors $\boldsymbol{x}_t, t = l_d+1, l_d+2,\dots,T$:

$$\boldsymbol{x}_t^{(p+1)}=\left(\sum_{i,j:(i,j,t)\in\Omega}\left(\boldsymbol{u}_i^{(p+1)}\odot\boldsymbol{v}_j^{(p+1)}\right)\left(\boldsymbol{u}_i^{(p+1)}\odot\boldsymbol{v}_j^{(p+1)}\right)^\top+\lambda_x\eta\boldsymbol{I}+\lambda_x\boldsymbol{I}\right)^{-1}\left(\sum_{i,j:(i,j,t)\in\Omega}\left(\boldsymbol{u}_i^{(p+1)}\odot\boldsymbol{v}_j^{(p+1)}\right)y_{ijt}+\lambda_x\boldsymbol{x}_t'^{(p+1)}\right)$$

where Ω denotes the index set of observed entries, $\odot$ denotes the Hadamard product operator, $\boldsymbol{x}_t'^{(p+1)}=f_{\theta^{(p)}}(\boldsymbol{x}_{t-l_1}^{(p+1)},\boldsymbol{x}_{t-l_2}^{(p+1)},\dots,\boldsymbol{x}_{t-l_d}^{(p+1)})$. LSTM network parameters θ can then be updated through back-propagation.

Table B.4
Running time on Seattle speed dataset 20% PM.

	LSTM-GL-ReMF	TRMF-ALS	TRMF-GRMF	BTMF	LSTM
Training	2610.2 s	1092.9 s	225.0 s	2635.9 s	605.3 s
Testing	9.7 s	33.2 s	6279.8 s	1400.1 s	0.5 s
	GRU-D	DCRNN	GCGRNN	STGCN	SGMN
Training	1648.2 s	85.2 s	29.4 s	1096.2 s	113.3 s
Testing	3.5 s	2.0 s	1.1 s	0.3 s	0.2 s

Table B.5
Running time on Metr-LA speed dataset.

	LSTM-GL-ReMF	TRMF-ALS	TRMF-GRMF	BTMF	LSTM
Training	2314.9 s	992.3 s	208.0 s	2696.0 s	849.2 s
Testing	8.7 s	19.5 s	5466.8 s	1421.0 s	0.5 s
	GRU-D	DCRNN	GCGRNN	STGCN	SGMN
Training	199.5 s	105.0 s	31.9 s	755.8 s	68.4 s
Testing	2.7 s	1.2 s	0.8 s	0.2 s	0.2 s

Table B.6
Running time on PeMSD8 speed dataset.

	LSTM-GL-ReTF	BTTF	ASTGCN	DCRNN	GCGRNN
Training	1153.4 s	492.8 s	2559.3 s	23.1 s	106.4 s
Testing	10.1 s	700.2 s	6.1 s	0.3 s	0.8 s

Appendix B. Computation time cost

The experiments are carried out on a Windows 10 computer with Intel i5-8265U 3.9 GHz CPU and 8G RAM. The computation time cost for model training and testing on Metr-LA dataset are reported in the following table. Since the computation complexity of LSTM-ReMF and LSTM-GL-ReMF are almost identical, only the running time of LSTM-GL-ReMF are reported.

It can be seen from Tables B.4 B.5 B.6 that although the proposed LSTM regularized matrix factorization models requires more time for static training, but it is significantly faster than TRMF-GRMF and BTMF in the online prediction and imputation phase due to the vector based parameter online updating mechanism.

References

Anava, O., Hazan, E., Zeevi, A., 2015. Online time series prediction with missing data. In: International Conference on Machine Learning, pp. 2191–2199.

Andres, M., Nair, R., 2017. A predictive-control framework to address bus bunching. Transp. Res. B 104, 123–148.

Bokde, D., Girase, S., Mukhopadhyay, D., 2015. Matrix factorization model in collaborative filtering algorithms: A survey. Procedia Comput. Sci. 49, 136–146.

Cai, D., He, X., Han, J., Huang, T.S., 2010. Graph regularized nonnegative matrix factorization for data representation. IEEE Trans. Pattern Anal. Mach. Intell. 33 (8), 1548–1560.

Che, Z., Purushotham, S., Cho, K., Sontag, D., Liu, Y., 2018. Recurrent neural networks for multivariate time series with missing values. Sci. Rep. 8 (1), 1–12.

Chen, X., He, Z., Chen, Y., Lu, Y., Wang, J., 2019a. Missing traffic data imputation and pattern discovery with a Bayesian augmented tensor factorization model. Transp. Res. C 104, http://dx.doi.org/10.1016/j.trc.2019.03.003, http://gen.lib.rus.ec/scimag/index.php?s=10.1016/j.trc.2019.03.003.

Chen, X., He, Z., Sun, L., 2019b. A Bayesian tensor decomposition approach for spatiotemporal traffic data imputation. Transp. Res. C 98, 73–84. http://dx.doi.org/10.1016/j.trc.2018.11.003, http://www.sciencedirect.com/science/article/pii/S0968090X1830799X.

Chen, C., Kwon, J., Rice, J., Skabardonis, A., Varaiya, P., 2003. Detecting errors and imputing missing data for single-loop surveillance systems. Transp. Res. Rec. 1855 (1), 160–167.

Chen, X., Yang, J., Sun, L., 2020. A nonconvex low-rank tensor completion model for spatiotemporal traffic data imputation. Transp. Res. C 117, 102673. http://dx.doi.org/10.1016/j.trc.2020.102673, http://www.sciencedirect.com/science/article/pii/S0968090X2030588X.

Chung, J., Gulcehre, C., Cho, K., Bengio, Y., 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. ArXiv Preprint arXiv:1412.3555.

Crookston, N.L., Finley, A.O., 2008. YaImpute: an R package for kNN imputation. Journal of Statistical Software 23 (10), 16.

Cui, Z., Henrickson, K., Ke, R., Wang, Y., 2019. Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. IEEE Trans. Intell. Transp. Syst..

Cui, Z., Lin, L., Pu, Z., Wang, Y., 2020. Graph markov network for traffic forecasting with missing data. Transp. Res. C 117, 102671.

Geng, X., Li, Y., Wang, L., Zhang, L., Yang, Q., Ye, J., Liu, Y., 2019. Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, pp. 3656–3663.

Gultekin, S., Paisley, J., 2019. Online forecasting matrix factorization. IEEE Trans. Signal Process. 67 (5), 1223–1236. http://dx.doi.org/10.1109/TSP.2018.2889982.

Guo, S., Lin, Y., Feng, N., Song, C., Wan, H., 2019. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33 (01), pp. 922–929.

He, W., Zhang, H., Zhang, L., Shen, H., 2015. Total-variation-regularized low-rank matrix factorization for hyperspectral image restoration. IEEE Trans. Geosci. Remote Sens. 54 (1), 178–188.

Hitchcock, F.L., 1927. The expression of a tensor or a polyadic as a sum of products. J. Math. Phys. 6 (1–4), 164–189.

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

Hu, J., Xin, X., Guo, P., 2017. LSTM with Matrix Factorization for Road Speed Prediction. pp. 242–249. http://dx.doi.org/10.1007/978-3-319-59072-1_29.

Jagadish, H.V., Gehrke, J., Labrinidis, A., Papakonstantinou, Y., Patel, J.M., Ramakrishnan, R., Shahabi, C., 2014. Big data and its technical challenges. Commun. ACM 57 (7), 86–94.

Lee, S., Lee, Y.-I., Cho, B., 2006. Short-term travel speed prediction models in car navigation systems. J. Adv. Transp. 40 (2), 122–139.

Lee, D., Seung, H., 2001. Algorithms for non-negative matrix factorization. Adv. Neural Inform. Process. Syst. 13.

Li, Y., Yu, R., Shahabi, C., Liu, Y., 2017. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. ArXiv Preprint arXiv:1707.01926.

Lin, L., He, Z., Peeta, S., 2018. Predicting station-level hourly demand in a large-scale bike-sharing network: A graph convolutional neural network approach. Transp. Res. C 97, 258–276.

Lin, L., Wang, Q., Sadek, A.W., 2013. Short-term forecasting of traffic volume: evaluating models based on multiple data sets and data diagnosis measures. Transp. Res. Rec. 2392 (1), 40–47.

Lin, L., Wang, Q., Sadek, A.W., 2015. A novel variable selection method based on frequent pattern tree for real-time traffic accident risk prediction. Transp. Res. C 55, 444–459.

Liu, J., Musialski, P., Wonka, P., Ye, J., 2012. Tensor completion for estimating missing values in visual data. IEEE Trans. Pattern Anal. Mach. Intell. 35 (1), 208–220.

Ma, Z., Chen, G., 2018. BayesIan methods for dealing with missing data problems. J. Korean Stat. Soc. 47, http://dx.doi.org/10.1016/j.jkss.2018.03.002.

Mirchandani, P., Head, L., 2001. A real-time traffic signal control system: architecture, algorithms, and analysis. Transp. Res. C 9 (6), 415–432. http://dx.doi.org/10.1016/S0968-090X(00)00047-4, http://www.sciencedirect.com/science/article/pii/S0968090X00000474.

Paltsev, S., Reilly, J.M., Jacoby, H.D., Eckaus, R.S., McFarland, J.R., Sarofim, M.C., Asadoorian, M.O., Babiker, M.H., 2005. The MIT Emissions Prediction and Policy Analysis (EPPA) Model: Version 4. Technical Report, MIT Joint Program on the Science and Policy of Global Change.

Purushotham, S., Liu, Y., Kuo, C.-C.J., 2012. Collaborative topic regression with social matrix factorization for recommendation systems. ArXiv Preprint arXiv:1206.4684.

Purwar, A., Singh, S.K., 2015. Hybrid prediction model with missing value imputation for medical data. Expert Syst. Appl. 42 (13), 5621–5631.

Qi, Z., Wang, T., Song, G., Hu, W., Li, X., Zhang, Z., 2018. Deep air learning: Interpolation, prediction, and feature analysis of fine-grained air quality. IEEE Trans. Knowl. Data Eng. 30 (12), 2285–2297.

Rao, N., Yu, H.-F., Ravikumar, P.K., Dhillon, I.S., 2015. Collaborative filtering with graph information: Consistency and scalable methods. In: Advances in Neural Information Processing Systems. pp. 2107–2115.

Salakhutdinov, R., Mnih, A., 2008. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In: Proceedings of the 25th International Conference on Machine Learning, pp. 880–887.

Sridevi, S., Rajaram, S., Parthiban, C., SibiArasan, S., Swadhikar, C., 2011. Imputation for the analysis of missing values and prediction of time series data. In: 2011 International Conference on Recent Trends in Information Technology (ICRTIT). IEEE, pp. 1158–1163.

Stellwagen, E., Tashman, L., 2013. ARIMA: The models of box and Jenkins. Foresight: Int. J. Appl. Forecast. 28–33.

Su, X., Khoshgoftaar, T.M., Greiner, R., 2008. Using imputation techniques to help learn accurate classifiers. In: 2008 20th IEEE International Conference on Tools with Artificial Intelligence. 1, IEEE, pp. 437–444.

Sun, L., Chen, X., 2019. BayesIan temporal factorization for multidimensional time series prediction. arXiv:abs/1910.06366.

Takeuchi, K., Kashima, H., Ueda, N., 2017. Autoregressive Tensor Factorization for Spatio-Temporal Predictions. In: 2017 IEEE International Conference on Data Mining (ICDM), pp. 1105–1110.

Takeuchi, K., Kawahara, Y., Iwata, T., 2017. Structurally regularized non-negative tensor factorization for spatio-temporal pattern discoveries. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, pp. 582–598.

Tong, Y., Chen, Y., Zhou, Z., Chen, L., Wang, J., Yang, Q., Ye, J., Lv, W., 2017. The simpler the better: a unified approach to predicting original taxi demands based on large-scale online platforms, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1653–1662.

Xin, B., Kawahara, Y., Wang, Y., Hu, L., Gao, W., 2016. Efficient generalized fused lasso and its applications. ACM Trans. Intell. Syst. Technol. (TIST) 7 (4), 1–22.

Yu, H.-F., Rao, N., Dhillon, I.S., 2016. Temporal regularized matrix factorization for high-dimensional time series prediction. In: Advances in Neural Information Processing Systems 29. Curran Associates, Inc., pp. 847–855.

Yu, B., Yin, H., Zhu, Z., 2017. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. ArXiv Preprint arXiv:1709.04875.

Yule, G.U., 1926. Why do we sometimes get nonsense-correlations between Time-Series?–a study in sampling and the nature of time-series. J. Roy. Statist. Soc. 89 (1), 1–63.

Zhang, S., 2012. Nearest neighbor selection for iteratively kNN imputation. J. Syst. Softw. 85 (11), 2541–2552.

Zhang, C., Wang, K., Yu, H., Sun, J., Lim, E.-P., 2014. Latent factor transition for dynamic collaborative filtering. In: Proceedings of the 2014 SIAM International Conference on Data Mining, pp. 452–460, doi:10.1137/1.9781611973440.52, https://epubs.siam.org/doi/abs/10.1137/1.9781611973440.52.

Zhang, W., Yu, Y., Qi, Y., Shu, F., Wang, Y., 2019. Short-term traffic flow prediction based on spatio-temporal analysis and CNN deep learning. Transp. Transp. Sci. 15 (2), 1688–1711.