Missing Traffic Data Imputation Using Multi-Trajectory Parameter Transferred LSTM

Jungmin Kwon
Ewha Womans University
Seoul, Republic of Korea
jungmin.kwon@ewhain.net

Hyunggon Park*
Ewha Womans University
Seoul, Republic of Korea
hyunggon.park@ewha.ac.kr

Abstract

We propose a lightweight data imputation algorithm for spatio-temporal data based on multi-trajectory parameter transferred Long Short-Term Memory (MPT-LSTM) in a traffic environment where the roadside units (RSUs) collect traffic information. In this paper, we consider a scenario where the RSUs are accidentally broken down or have malfunctioned but cannot be immediately recovered, temporally incurring massive data loss. Unlike existing imputation algorithms based on LSTM, the proposed architecture reduces the dimensions of input data by separating the data according to trajectories in the spatio-temporal domain, thereby allowing us to train the model with irreducible LSTMs for each single input data. The designed approach can transfer parameters between irreducible LSTM phases, which trains data collected from an RSU, adopts spatial interpolation, and shows robust imputation accuracy for the trajectories. We show that the proposed MPT-LSTM improves imputation accuracy and significantly reduces the number of parameters and operations, which leads to a reduction in LSTM memory space and execution time. The proposed algorithm can also show robust and efficient performance in the experiments using real-world vehicle speed data collected from expressways and urban areas.

1 Introduction

The autonomous driving industry is growing at an unprecedented speed with technological advancements in vehicles and intelligent transport systems. To achieve a fully autonomous system, a current centralized system consists of servers or clouds that need to constantly collect data generated from a traffic environment and analyze the data for inference in real-time such that vehicles can use it. For example, vehicle behavior data (e.g., speed, geographical location, etc.), closely related to traffic environments, are becoming essential because of their high potential for use in traffic control. Therefore, it is important to analyze the data so that the appropriate traffic control signals can be seamlessly supported based on the inference. However, it is challenging to cope with the bulk of data loss incurred by unexpected network malfunction or damage to the hardware or software. For vehicular services, this becomes challenging as physical data collectors such as roadside units (RSUs) can accidentally break down or malfunction and cannot be repaired immediately. Thus, recovering the missing data is especially important in providing seamless data-driven services.

For data imputation methods, machine learning (ML) based approaches have been adopted as they can capture the uncertainty, non-linear trends, and dependencies in the data. The Long Short-Term Memory (LSTM) model is one of the representative ML models that has been widely used for

*The author is also with Smart Factory Multidisciplinary Program, Department of Electronic and Electrical Engineering at Ewha Womans University, Seoul, Republic of Korea.

Machine Learning for Autonomous Driving Workshop at the 36th Conference on Neural Information Processing Systems (NeurIPS 2022), New Orleans, USA.

data imputation of complex and non-linear datasets, including time-series data [1, 2]. However, the complexity associated with memory and time requirements for LSTM can explode exponentially depending on the size of the input data, which is directly related to the size of the model parameters. Since most of the studies have been focusing on accuracy, the memory requirement for the data imputation algorithms is often ignored. Therefore, it is important to design data imputation algorithms that are lightweight in both memory and computational complexity.

In this paper, we consider a traffic environment consisting of vehicles on roads where RSUs installed along with the roads may temporarily fail to operate, so that the spatio-temporal data collected at the location can be lost. Since the vehicles can move along the roads in various paths in the traffic environment, we consider multiple trajectories for more realistic scenarios. The proposed data imputation algorithm imputes the missing data from these multiple trajectories. This paper focuses on improving the data imputation accuracy while simultaneously reducing the complexity compared to existing LSTMs using spatio-temporal data characteristics. We propose a lightweight architecture for a low complexity spatio-temporal data imputation algorithm based on multi-trajectory parameter transferred LSTM (MPT-LSTM), where the key idea is to separate the dataset according to trajectories and the spatio-temporal domain. While parameter transferred LSTM has been used in [3] for bidirectional data imputation, it can only consider a single trajectory. Since the imputation accuracy is highly dependent on the correlations and characteristics of the data collected, the algorithm proposed in [3] may show significantly low data imputation accuracy if the selected bidirectional trajectories among the various trajectories have low correlations with the target-missing data. The proposed MPT-LSTM, however, can consider multiple trajectories by dividing a standard LSTM model into multiple irreducible LSTMs, and each irreducible LSTM is designed for the data imputation in an individual trajectory. This can lower the dimension of the data and thus makes MPT-LSTM a low complexity algorithm. More importantly, the division of LSTM into irreducible LSTMs can lead to a robust algorithm as the data imputation for an individual trajectory is processed in each irreducible LSTM, effectively limiting the potential imputation error propagation. Finally, MPT-LSTM can improve the data imputation accuracy by sharing the spatio-temporal dependencies between data in multiple trajectories via parameter transfer.

Using an extensive set of experiments with real-world datasets, we confirm that the proposed MPT-LSTM algorithm achieves better performance in both accuracy and complexity compared to state-of-the-art imputation algorithms. Moreover, MPT-LSTM shows steady and robust imputation accuracy with low complexity, even for dynamic changes in the number of trajectories.

2 Related Works

Many ML-based algorithms that ensure the data imputation performance in a single domain exist in the literature, e.g., the temporal domain. In [4], traffic flow prediction for missing data using multiscale LSTM (LSTM-M) is proposed, which outperforms several state-of-the-art methods in terms of accuracy for both short-time and long-time predictions. However, LSTM-M may generate constant values, such as zero values, if the available information is not enough to capture the patterns of missing data. An imputation algorithm for consecutively missing values for long-period time-series data is proposed based on transferred LSTM networks [5]. While these prediction approaches can impute the missing data with improved performance, it is still challenging to recover a large amount of consecutive data potentially missed by the failure of data collectors. The combination of DNN models has been studied to integrate their advantages. In [6], LSTM and Bidirectional-LSTM (Bi-LSTM) are combined to enhance the feature learning from the large-scale spatial time-series data. Similarly, stacked LSTM is used for the prediction of spatio-temporal weather data [7]. These approaches are based on a multi-layered structure where each layer must be processed sequentially, and a layer is trained in a single domain.

In [8], a precipitation nowcasting problem is formulated as spatio-temporal sequence forecasting by extending a fully connected LSTM (FC-LSTM) model to convolutional LSTM (ConvLSTM). In this approach, ConvLSTM captures not only longer temporal dependencies but also spatio-temporal correlations so that it can consider both domains simultaneously. The imputation accuracy of the ConvLSTM model is improved by combining Bi-LSTM for capturing spatial and temporal patterns [9]. However, the existing LSTM models and CNN-based LSTM models for spatio-temporal data imputation suffer from a limitation of the polynomial expansion of computational complexity in input data dimension and size. This is because the relevant features of multiple input data are trained

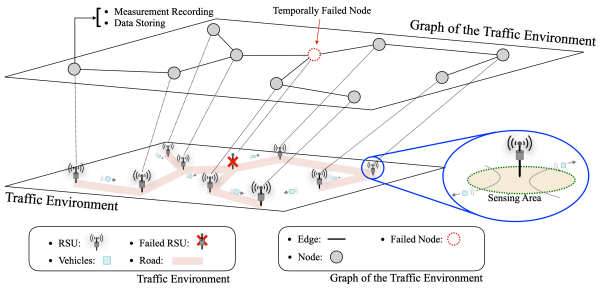

Figure 1: An illustrative example and graph of the traffic environment.

at a time, which requires a high-dimensional expansion of weight parameters [10, 11]. Therefore, a key challenge is to reduce the dimensions of LSTM model parameters for an efficient learning process over spatio-temporal data imputation with lower complexity without imputation accuracy degradation.

Sharing model parameters can be one of the approaches that consider both model efficiency and accuracy in spatio-temporal data imputation [12, 13]. These studies share model weights to capture the informative features to deal with mutual independence between separated LSTM. A memory cell-based spatio-temporal LSTM (ST-LSTM) model is proposed in [12], which allows interactions between mutually independent LSTM units. It can reduce the complexity and memory space by updating LSTM units from the previous memory state. Finally, the tensor encoding for capturing all spatial characteristics without any loss in spatial information ignores the direct causal relationships between each spatial data [12, 13].

Unlike these parameter sharing algorithms, we propose a parameter transferring architecture that updates the LSTM parameters from the preceding LSTM phase for spatial interactions without modifying the standard LSTM or adding new units. While the proposed MPT-LSTM also sequentially updates the LSTM model parameters from the preceding LSTM, the input data is divided into both spatio-temporal domains to train a single data point at a time, leading to the reduction of input data size. In our model, we adopt a graphical methodology using trajectory such that we make causal subgroup input data highly correlated. Splitting the input dataset by trajectories significantly reduces the complexity of the imputation algorithm. Furthermore, using spatial interpolation allows us to provide stable and robust imputation accuracy for dynamics in the number of trajectories.

3 System Setup and Problem Formulation

In this paper, we consider a traffic environment where the average vehicle speed is monitored over time and stored at RSUs. The traffic environment is modeled as graph $G(\mathbf{V}, \mathbf{E})$, where a set of RSUs and a set of roads are represented by node set $\mathbf{V}$ and an undirected edge set $\mathbf{E}$, respectively, as shown in Fig. 1. The vehicles move along the road, and each RSU collects spatio-temporal data, including average travel speed. We consider a case where a single RSUs temporarily fail to operate such that the spatio-temporal data collected in the failed RSUs is not available.

There are M trajectories ending at the failed RSU, and there are N RSUs along each trajectory, excluding the failed RSU. We denote the sequence of RSUs in the mth trajectory, which is a path of vehicles around the failed RSU, as $\mathbf{s}_N^m = [v_0^m, \cdots, v_n^m, \cdots, v_N^m]$, where $v_n^m \in \mathbf{V}$. The RSU sequence begins with the target failed RSU and is arranged in the order of adjacent RSUs. Hence, the first RSU of each trajectory is the same target RSU (i.e., a failed RSU $v_0^1 = \cdots = v_0^m$). For the simplicity of notation, the failed RSU is denoted by v_0, i.e., $v_0 = v_0^1 = \cdots = v_0^m$. The spatio-temporal data of an RSU is denoted as $\mathbf{x}_n^m = [x_n^m(1), \cdots, x_n^m(T)] \subseteq \mathbf{X}$, where $\mathbf{X}$ is the whole dataset of spatio-temporal data collected in the traffic environment, T is the data length, and $x_n^m(t) \in \mathbb{R}$ is the observed data at v_n^m at time t. The missing data at the target $\mathbf{v}_0$ is denoted by $\mathbf{x}_0$, which is the same as $\mathbf{x}_0^m$ because the missing data in M trajectories indicate the same data (i.e., $\mathbf{x}_0 = \mathbf{x}_0^1 = \cdots = \mathbf{x}_0^m$).

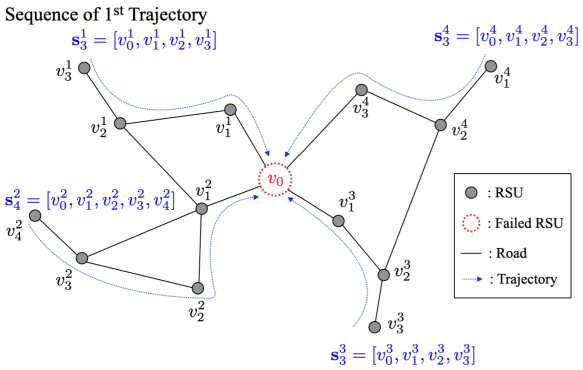

Figure 2: An illustrative examples of traffic environment and spatio-temporal data extrapolation.

Our objective is to design an imputation function f that learns $\tilde{\mathbf{x}}_0$, best approximate $\mathbf{x}_0$, based on $\mathbf{X}_{-0}$, i.e.,

$$\begin{aligned} &\text{design} \quad f \\ &\text{subject to} \quad \|\mathbf{x}_0 - f(\mathbf{X}_{-0})\| \leq \epsilon, \end{aligned} \tag{1}$$

where $f(\mathbf{X}_{-0}) = \tilde{\mathbf{x}}_0$, and $\mathbf{X}_{-0}$ denotes the neighbors' dataset collected from all RSUs except v_0, i.e., $\mathbf{X}_{-0} = \mathbf{X} - \mathbf{x}_0$. Unfortunately, solving this problem generally requires significantly high complexity as it considers the high dimensional spatio-temporal data $\mathbf{X}_{-0}$.

In order to reduce the computation complexity, we lower the dimension of the input dataset by partitioning $\mathbf{X}$ into $\mathbf{x}^1, \mathbf{x}^2, \cdots, \mathbf{x}^M$ according to M trajectories, where $\mathbf{x}^m = [\mathbf{x}_0^m, \cdots, \mathbf{x}_N^m]$ is a set of sequential data of $\mathbf{s}_N^m$. An example of the traffic environment is shown in Fig. 2 for $M = 4$ and $N = 3, 4$ with a failed RSU v_0. We further reduce the computational complexity by additional data separation in the spatial domain such that the single size of data is considered for the imputation algorithm. The successive reduction of the dimension of the spatio-temporal data makes the algorithm lightweight.

4 Proposed Designed Architecture

The proposed algorithm consists of two main stages: extrapolation, and spatial interpolation. After splitting the data according to the locations of RSUs, the duplicated target-missing data are extrapolated using the factorized LSTM. Then, the multiple extrapolated data in a spatial domain are interpolated.

4.1 Extrapolation Based on LSTM Factorization with Parameter Transfer

4.1.1 LSTM Factorization

Let $g_{\text{LSTM}} : \mathbb{R}^{M \times N \times T} \to \mathbb{R}^T$ be an extrapolation function based on the standard LSTM model. Then, the first data of the sequence $\hat{\mathbf{x}}_0^m$ can be extrapolated by g_{LSTM} such as

$$\hat{\mathbf{x}}_0^m = g_{\text{LSTM}}(\mathbf{x}_1^m, \cdots, \mathbf{x}_N^m) \tag{2}$$

for a single trajectory $\mathbf{s}_N^m$. If ideally extrapolated, the original data can be obtained, i.e., $\hat{\mathbf{x}}_0^m = \mathbf{x}_0^m$. In the case of a standard LSTM, as mentioned earlier, the entire data set of sequential input data $(\mathbf{x}_1^m, \cdots, \mathbf{x}_N^m)$ is considered at once to obtain target-missing data $\mathbf{x}_0^m$.

To reduce the input data size based on data separation, we factorize the standard LSTM into irreducible LSTMs such that each irreducible LSTM can take only one-dimensional input data $\mathbf{x}_n^m(t)$. For this, we propose a function g for an irreducible LSTM model that can only consider a single phase of LSTM and a single data collected from a single RSU as input data. This can be simply expressed as a function $g : \mathbb{R}^T \to \mathbb{R}^T$ defined as $\mathbf{x}_n^m = g(\mathbf{x}_{n+1}^m)$. Since only single input data is considered, the extrapolation process is generally an under-fitting problem. This problem can be solved by considering the irreducible LSTM phases together, expressed as

$$\mathbf{x}_0^m = g^N(\mathbf{x}_N^m), \tag{3}$$

where $g^n(\cdot) = \overbrace{g \circ g \circ \cdots \circ g(\cdot)}^{n}$ denotes the function composition. Note that the function composition in (3) can be considered as the parameter transfer between the phases of MPT-LSTM. Since dataset $\mathbf{X}_{-0}$ is separated into M subgroups, we can construct M irreducible LSTM layers in parallel. The subgroup data $\mathbf{x}_{-0}^m$ collected from each trajectory is used as the input data for each LSTM layer. Thus, we can extrapolate M duplicated $\hat{\mathbf{x}}_0^m$ data for missing data $\mathbf{x}_0$.

4.1.2 Extrapolation

A single layer of MPT-LSTM consists of $(N-1)$ phases of LSTM model training and one phase of LSTM extrapolation. In the model training phase, MPT-LSTM parameters trained from an LSTM phase are transferred to the next LSTM phase. Each LSTM phase predicts data collected from the adjacent node up to target node v_0. For the data extrapolation at the mth trajectory, $\mathbf{x}_n^m$ is used as input data to predict the data $\mathbf{x}_{n-1}^m$ at the $N-(n-1)$th LSTM phase. Note that extrapolation for model training is performed backward from the $\mathbf{x}_N^m$ to $\mathbf{x}_1^m$.

Let $f_n^m(t)$, $i_n^m(t)$, $c_n^m(t)$, $o_n^m(t)$, $h_n^m(t)$, and $y_n^m(t)$ be the variables of the forget gate, input gate, cell state, output gate, hidden state, and prediction of the nth LSTM phase of the mth trajectory at time t, expressed as

$$\begin{aligned}
f_n^m(t) =& \sigma(W_f^m(n) \cdot [h_n^m(t-1), x_{N-(n-1)}^m(t)] + b_f^m(n)),\\
i_n^m(t) =& \sigma(W_i^m(n) \cdot [h_n^m(t-1), x_{N-(n-1)}^m(t)] + b_i^m(n)),\\
c_n^m(t) =& f_n^m(t) \star c_n^m(t-1) + i_n^m(t)\\
&\star \tanh(W_c^m(n)[h_n^m(t-1), x_{N-(n-1)}^m(t)]\\
&+ b_i^m(n)),\\
o_n^m(t) =& \sigma(W_o^m(n) \cdot [h_n^m(t-1), x_{N-(N-1)}^m(t)] + b_o^m(n)),\\
h_n^m(t) =& o_n^m(t) \star \tanh(c_n^m(t)),\\
y_n^m(t) =& d_1^m(n) \cdot d_2^m(n) \cdot h_n^m(t),
\end{aligned} \tag{4}$$

where $\star$ denotes the Hadamard product, $\mathbf{W}^m(n) = [W_f^m(n), W_i^m(n), W_c^m(n), W_o^m(n)]$ denotes the weight matrix at the nth phase with forget gate weights $W_f^m(n)$, input gate weights $W_i^m(n)$, cell state weights $W_c^m(n)$, and output gate weights $W_c^m(n)$. We also consider the bias vector parameters $\mathbf{b}^m(N) = [b_f^m(n), b_i^m(n), b_o^m(n)]$ with the forget gate bias parameter $b_f^m(n)$, input gate bias parameter $b_i^m(n)$, and the output gate bias parameter $b_o^m(n)$. In MPT-LSTM, two dense layers are placed at the end of the LSTM model with parameters $\mathbf{d}^m(n) = [d_1^m(n), d_2^m(n)]$. The activation function is denoted by $\sigma(\cdot)$, which can be, e.g., Rectified Linear Unit (ReLU), linear, sigmoid, etc. In each phase, mean square error (MSE) is used as a measure of loss, expressed as

$$\mathcal{L}_n^m = \frac{1}{T}\sum_{t=1}^{T}(x_{N-(n-2)}^m(t) - y_n^m(t))^2, \tag{5}$$

where $y_n^m(t)$ denotes the prediction of the nth LSTM phase at time t. Note that the LSTM training procedures are identical to the other trajectories to predict $\mathbf{x}_{N-(n-1)}^m$ from $\mathbf{x}_{N-(n-2)}^m$ at the nth LSTM phase.

4.1.3 Parameter Transfer

The parameters of the LSTM network from the trained nth LSTM phase are transferred to the next phase. Unlike the standard LSTM extrapolation, where the parameters can only be shared at consecutive time steps in a single LSTM phase, the proposed algorithm can transfer the parameters not only over time steps but also over LSTM phases. This can be performed by passing the weight matrices, cell states, and hidden states, i.e.,

$$\begin{aligned}
\mathbf{W}^m(n+1) &\leftarrow \mathbf{W}^m(n),\\
\mathbf{b}^m(n+1) &\leftarrow \mathbf{b}^m(n),\\
c_{n+1}^m(0) &\leftarrow c_n^m(T),\\
h_{n+1}^m(0) &\leftarrow h_n^m(T).
\end{aligned} \tag{6}$$

The parameters trained from the nth LSTM phase are transferred to the $(n+1)$th LSTM phase as its initial parameters for training. This process repeats up to the $(N-1)$th LSTM phase. Hence, the interconnection information lost by the data separation in the spatio-temporal domain can be recovered by the parameter transfer between phases.

After completing $(N-1)$ LSTM phase training, the target-missing data $\mathbf{x}_0^m$ can be extrapolated by the data collected at the adjacent node $\mathbf{v}_1^m$. The functions and parameters of the extrapolation phases are denoted with an asterisk, i.e., $f^{m*}(t)$, $i^{m*}(t)$, $c^{m*}(t)$, $o^{m*}(t)$, $h^{m*}(t)$, $y^{m*}(t)$ $\mathbf{W}^{m*} = [W_f^{m*}, W_i^{m*}, W_c^{m*}, W_o^{m*}]$, $\mathbf{d}^{m*} = [d_1^{m*}, d_2^{m*}]$, and $\mathbf{b}^{m*} = [b_f^{m*}, b_i^{m*}, b_o^{m*}]$. Then, the parameters of the LSTM extrapolation phase are initialized by duplicating the parameters of the $(N-1)$th trained LSTM phase as

$$\begin{aligned} \mathbf{W}^{m*} &\leftarrow \mathbf{W}^m(N-1), \\ \mathbf{b}^{m*} &\leftarrow \mathbf{b}^m(N-1), \\ \mathbf{d}^{m*} &\leftarrow \mathbf{d}^m(N-1), \\ c^{m*}(0) &\leftarrow c_{N-1}^m(T), \\ h^{m*}(0) &\leftarrow h_{N-1}^m(T). \end{aligned} \tag{7}$$

Finally, the missing data of the mth trajectory is extrapolated by $\hat{x}_0^m(t) = y^{m*}(t)$ for all t with the duplicated weight parameters.

The MPT-LSTM architecture can enhance the extrapolation accuracy as the weight parameters of the LSTM phases are initialized from the data collected at the adjacent RSU. That is, MPT-LSTM can further reduce the $\mathcal{L}_n^m$ for a given LSTM training iteration by capturing the prevalent patterns of target spatio-temporal data in an early time step. To obtain the imputed data $\tilde{\mathbf{x}}_0$, the spatial interpolation is processed after data extrapolation using the multiple extrapolated data $\hat{\mathbf{x}}_0^m$.

4.2 Spatial Interpolation

For more robust data imputation, the extrapolated data can be interpolated in a sequel by explicitly considering the spatial correlation. The interpolation algorithm is implemented based on Angular Distance Weighting (ADW) [14], whose general form of spatial interpolation is given by

$$\tilde{\mathbf{x}}_0 = \frac{\sum_{m=1}^{M}\sum_{n=1}^{N} w_n^m \mathbf{x}_n^m}{\sum_{m=1}^{M}\sum_{n=1}^{N} w_n^m}, \tag{8}$$

where the weight w_n^m is defined as a distance weight. The weight can be designed such that it can include angular information (i.e., ADW), which has a form of

$$w_n^m = \gamma_n^m \cdot \left(1 + \frac{\sum_{m=1}^{M}\sum_{n=1}^{N} \gamma_n^m [1-\cos(\theta_{n,m}^0)]}{\sum_{m=1}^{M}\sum_{n=1}^{N} \gamma_n^m}\right), \tag{9}$$

where $\gamma_n^m = e^{-\beta(q_{(n,m)}/CDD)}$ denotes a distance weight with a correlation decay distance (CDD) for all the locations in the dataset. Here, $\theta_{n,m}^0$ denotes the angle difference between v_n^m and v_0, $q_{(n,m)}$ denotes the distance between v_n^m and v_0, and β denotes the user-dependent exponent.

Since only M extrapolated data need to be considered, the interpolation can be simply expressed as

$$\tilde{\mathbf{x}}_0 = \frac{\sum_{m=1}^{M} w_m^* \hat{\mathbf{x}}_0^m}{\sum_{m=1}^{M} w_m^*}, \tag{10}$$

which is the case of $n=0$ in (8), and w_m^* is defined as the distance weight. Since M extrapolated data are generated from each trajectory, the data correlation of each trajectory should be included in weight w_m^*. Therefore, the weight w_m^* is expressed as

$$w_m^* = \rho_s \cdot \gamma_0^m \cdot \left(1 + \frac{\sum_{m=1}^{M} \gamma_0^m [1-\cos(\theta_{0,m}^0)]}{\sum_{m=1}^{M} \gamma_0^m}\right), \tag{11}$$

with the average spatio-temporal correlation ρ_s at the mth trajectory in (12). Here, $\overline{\mathbf{x}}_n^m = \frac{1}{T}\sum_{t=1}^{T} x_n^m(t)$. Since all duplicated extrapolated data $\hat{\mathbf{x}}_0^m$ are considered to be collected from the same target RSU v_0, however, the angular difference and distance difference between the duplicated RSUs and v_0 becomes zero (i.e., $\theta_{0,m}^0 = q_{(0,m)} = 0$ for all m). Therefore, the imputed data $\tilde{\mathbf{x}}_0$ is determined as a weighted linear regression of $\hat{\mathbf{x}}_0^m$ for all trajectories with simplified weight $w_m^* = \rho_s$.

$$\rho_s = \frac{1}{N-1} \cdot \sum_{n=1}^{N-1} \frac{\sum_{t=1}^{T}(x_n^m(t) - \overline{\mathbf{x}}_n^m)(x_{n+1}^m(T) - \overline{\mathbf{x}}_{n+1}^m)}{\sqrt{\sum_{t=1}^{T}(x_n^m(t) - \overline{x}_n^m)^2 \sum_{t=1}^{T}(x_{n+1}^m(t) - \overline{x}_{n+1}^m)^2}} \tag{12}$$

5 Experiments

In this section, we evaluate the performance of the proposed algorithm in terms of complexity reduction and accuracy improvement using a real-world dataset. We quantitatively evaluate the space and time complexity of MPT-LSTM compared to other state-of-art algorithms. Finally, we evaluate the imputation accuracy and robustness of the proposed algorithm against other algorithms with the goal of imputing the missing data in the imperfect dataset.

5.1 Experiment Setup

In this section, we use real-world datasets to perform data imputation experiments. To evaluate the proposed data imputation algorithm, we use two different types of vehicle speed datasets.

- Expressway dataset: The speed dataset measured on the expressway was collected from Korea Expressway Corporation [15]. It was collected from a Vehicle Detection System (VDS) every hour of the day. We use the data collected in June of 2022 and consider four days of data as a single data length. We consider that there are four trajectories around the target RSUs and six RSUs in a single trajectory.
- Urban dataset: The speed dataset measured in an urban location in downtown Seoul was collected from the Transport Operation & Information Service in Korea [16]. The speed data are collected every hour in a day from more than 70,000 card taxis with location information (GPS) and average travel speed. We consider four days of data as a single data length with four trajectories and six RSUs in a single trajectory collected in June of 2022.

We evaluate the imputation performance against seven algorithms and conducted 500 independent experiments. We report the averaged imputation performance of the target RSU and report the mean and standard deviation of the performance in four datasets (two for the expressway dataset and two for the urban speed dataset). The spatio-temporal correlation of all trajectories satisfies $0.58 \leq \rho_s \leq 0.98$.

In this paper, we use the root mean square error (RMSE) as an accuracy measure $e(\cdot, \cdot)$, expressed as

$$e(\hat{\mathbf{x}}_0, \mathbf{x}_0) = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(\hat{x}_0(t) - x_0(t))^2}. \tag{13}$$

As a measure of complexity, we consider two types of computational complexity: time complexity and space complexity. The time complexity of algorithms is measured by the number of flops and actual execution time. The space complexity is measured by the number of parameters and memory size required for MPT-LSTM.

We use sigmoid as the activation function of each neural network layer. We train all the models with an Adam optimizer [17] with a learning rate of 0.001, an epoch of $300 \times (N-2)$ with input dimension $N-2$ for a single trajectory, and an epoch of $300 \times (N-2) \times M$ for multiple trajectories. The depth of LSTM networks is set to the data length, i.e., T. The number of batches in the temporal domain is set to 32, and the batch size is set to M for both MPT-LSTM and state-of-the-art algorithms. The accuracy and imputation time are averaged over 500 independent simulations.

For performance comparison with the proposed algorithm, we consider existing data imputation algorithms, which are LSTM, Bi-LSTM, ConvLSTM, GRU, Stacked LSTM [7], and BRITS [6]. In the case of ConvLSTM, the filter and kernel size are set as $N-2$, which is the number of the training dataset. All simulations with models described above are performed on a single GPU (NVIDIA GeForce RTX 3080 Ti).

Table 1: Computational complexity of Learning models on the Tensorflow framework where $T = 100$, $N = 6$, and $M = 4$.

Algorithm	Space Complexity		Time Complexity	
	Parameters	Memory Size (Byte)	Flops	Execution Time (sec)
CNN-LSTM	237	21,037	192,416	24.08
GRU	145	14,545	117,208	24.50
LSTM	169	14,569	139,616	22.23
Bi-LSTM	329	21,129	274,032	28.42
Stacked LSTM	225	27,425	189,280	28.07
Brits	161,018	187,386	516,520	40.11
MPT-LSTM	**64**	**4,816**	**13,616**	**21.80**

5.2 Evaluation of Time Complexity and Space Complexity

For the analysis of the space complexity, we confirm the computational complexity in the Tensorflow framework. Table 1 shows the space complexity based on the Tensorflow framework [18], where $T = 100$, $N = 6$, and $M = 4$. It is clearly observed that MPT-LSTM significantly reduces space complexity compared to existing algorithms on the basis of both the number of parameters and the memory size. For instance, MPT-LSTM requires only 16 parameters which are only 44.13% of the next-best algorithm (GRU with 145 parameters). This is because the parameter transfer of MPT-LSTM does not need to set new parameters for every layer. Similarly, the time complexity for the algorithms shows that the proposed MPT-LSTM outperforms other algorithms. This is due to the fact that MPT-LSTM splits data into lower dimensions based on the spatio-temporal domain resulting in lower numbers of parameters and flops.

Although MPT-LSTM shows the lowest execution time with a narrow margin compared to standard LSTM, as discussed in Table 1, the performance of space complexity would suggest that MPT-LSTM extrapolates the missing data with the lowest amount of memory.

5.3 Evaluation of Imputation Accuracy

Table 2 shows the imputation results at two locations. It can be observed that any particular algorithm cannot always show the best imputation accuracy. Rather, the performance of imputation accuracy depends on the traffic data set. However, we can observe that the imputation accuracy of the proposed MPT-LSTM using real-world datasets is generally better than the other algorithms. For example, MPT-LSTM imputation accuracy outperforms other algorithms in most expressway datasets except at two trajectories in Expressway #1 dataset. In particular, it can be observed that the proposed MPT-LSTM consistently outperforms the standard LSTMs at multi-trajectories ($M > 1$). Although the imputation accuracy depends on the data characteristics (such as spatio-temporal correlation, stationarity, or mean and variance), MTP-LSTM outperforms the other algorithms in terms of accuracy and robustness when multi-trajectory is considered for data imputation.

These results are similar to the urban speed dataset, as shown in Table 3. It is observed that the imputation accuracy using MPT-LSTM performs better compared to other algorithms, but there are larger variations on the RMSEs. This can be implied by the average standard deviations, which are 0.0414 for the expressway dataset and 0.1238 for the urban dataset. The imputation accuracy of MPT-LSTM is the best in most urban datasets, except for a single trajectory. Note that the spatio-temporal correlation ρ_s for each trajectory is widely distributed, as there are various complex urban roads. While the improvement of imputation accuracy varies depending on road characteristics, we can conclude that the proposed MPT-LSTM outperforms the other algorithms in terms of imputation accuracy and robustness.

6 Conclusion

In this paper, we proposed an imputation algorithm for the missing spatio-temporal data based on MPT-LSTM in a traffic environment where multi-trajectories are placed. We propose a lightweight MPT-LSTM algorithm that successively separates the data for trajectories and the spatio-temporal

Table 2: Imputation performance comparison with expressway speed dataset (Mean ± Std of RMSE)

Algorithm	Expressway #1 ($0.93 < \rho_s < 0.98$)			
	1	2	3	4
CNN-LSTM	6.86±0.38	5.86±1.79	9.27±3.80	4.33±0.73
GRU	7.08±0.23	7.67±2.78	9.60±1.10	5.38±2.14
LSTM	7.11±0.43	7.38±3.46	8.53±1.24	5.02±1.72
Bi-LSTM	7.25±0.27	7.68±3.01	9.99±1.50	4.89±1.24
Stacked	6.47±0.31	3.53±0.19	11.54±2.47	4.66±0.63
Brits	6.39±0.42	7.61±1.61	8.77±1.09	4.96±1.01
MPT-LSTM	**4.41±0.48**	4.71±0.40	**3.86±0.18**	**3.97±0.18**
Algorithm	Expressway #2 ($0.93 < \rho_s < 0.98$)			
	1	2	3	4
CNN-LSTM	6.10±1.50	10.79±3.26	9.32±1.29	7.85±1.51
GRU	5.43±0.11	9.61±4.67	9.57±0.66	7.46±3.54
LSTM	7.62±2.57	9.51±4.81	9.70±1.13	7.30±2.83
Bi-LSTM	5.40±0.20	10.18±4.59	10.82±1.02	7.66±4.05
Stacked	5.87±0.12	11.37±3.78	9.19±1.36	7.69±1.16
Brits	13.70±0.51	12.72±2.71	15.19±2.51	14.24±0.77
MPT-LSTM	**5.34±0.21**	**8.27±0.12**	**7.28±0.18**	**6.84±0.14**

Table 3: Imputation performance comparison with urban speed dataset (Mean ± Std of RMSE)

Algorithm	Urban #1 ($0.66 < \rho_s < 0.86$)			
	1	2	3	4
CNN-LSTM	10.11±7.73	7.25±11.52	10.00±17.02	9.00±8.03
GRU	9.84±2.39	7.86±16.08	10.94±15.28	10.84±14.62
LSTM	9.54±9.38	8.16±16.67	10.73±13.55	10.53±11.57
Bi-LSTM	10.45±4.01	8.20±13.98	11.15±14.54	9.81±11.39
Stacked	8.23±2.34	9.24±7.80	6.71±4.93	7.66±9.93
Brits	5.67±2.81	5.34±5.71	8.22±5.42	6.26±11.05
MPT-LSTM	9.37±0.66	**3.32±2.78**	**3.61±2.08**	**5.71±1.36**
Algorithm	Urban #2 ($0.58 < \rho_s < 0.92$)			
	1	2	3	4
CNN-LSTM	5.47±0.83	12.40±3.29	8.13±3.08	10.90±3.71
GRU	5.79±0.52	12.24±2.89	8.67±3.64	10.36±3.86
LSTM	5.54±1.14	12.30±3.30	8.18±2.98	10.19±3.30
Bi-LSTM	5.60±0.61	11.84±2.59	8.22±3.24	10.21±4.01
Stacked	4.90±0.17	10.42±0.67	8.19±3.56	9.89±2.70
Brits	7.69±0.82	11.88±3.30	10.62±2.98	10.72±3.30
MPT-LSTM	6.83±0.19	**5.08±0.32**	**4.25±0.20**	**5.14±0.16**

domain by factorizing the standard LSTM architecture into irreducible LSTMs. The complexity of the proposed MPT-LSTM can thus be prominently reduced as the dimension of parameters is reduced to a single size, and the imputation accuracy can be improved by the parameter transfer to the next phase of LSTM. Moreover, spatial interpolation merges the extrapolated data to capture the variabilities of the trajectories leading to robust imputation accuracy. Our experiments with the real-world datasets show that the MPT-LSTM outperforms the state-of-the-art imputation algorithms in both imputation accuracy and complexity. Therefore, leveraging the contributions of spatio-temporal correlation in the traffic environment, MPT-LSTM demonstrates consistent and significant improvements over state-of-the-art algorithms in imputing realistic spatio-temporal data.

Acknowledgments and Disclosure of Funding

This work was supported by an Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2021-0-00739, Development of Distributed/Cooperative AI based 5G+ Network Data Analytics Functions and Control Technology) and in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2020R1A2B5B01002528)

References

[1] Y. Zhang, M. Pezeshki, P. Brakel, S. Zhang, C. L. Y. Bengio, and A. Courville, "Towards end-to-end speech recognition with deep convolutional neural networks," in *17th Annual Conference of the International Speech Communication Association (INTERSPEECH 2016)*, 2016, pp. 410–414.

[2] X. Qing and Y. Niu, "Hourly day-ahead solar irradiance prediction using weather forecasts by LSTM," *Energy*, vol. 148, pp. 461–468, 2018.

[3] J. Kwon, C. Cha, and H. Park, "Bidirectional imputation of spatio-temporal data based on lstm with parameter transfer," in *IEEE Wireless Communications and Networking Conference (WCNC 2021)*, 2021, pp. 1–6.

[4] Y. Tian, K. Zhang, J. Li, X. Lin, and B. Yang, "LSTM-based traffic flow prediction with missing data," *Neurocomputing*, vol. 318, pp. 297–305, 2018.

[5] J. Ma, J. C. Cheng, Y. Ding, C. Lin, F. Jiang, M. Wang, and C. Zhai, "Transfer learning for long-interval consecutive missing values imputation without external features in air pollution time series," *Advanced Engineering Informatics*, vol. 44, no. 101092, 2020.

[6] W. Cao, D. Wang, J. Li, H. Zhou, L. Li, and Y. Li, "Brits: Bidirectional recurrent imputation for time series," in *Conference on Neural Information Processing Systems (NeurIPS 2018)*, 2018, pp. 6775–6785.

[7] Z. Karevan and J. A. Suykens, "Spatio-temporal stacked LSTM for temperature prediction in weather forecasting," *arXiv preprint arXiv:1811.06341*, 2018.

[8] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Conference on Neural Information Processing Systems (NeurIPS 2015)*, vol. 28, 2015, pp. 802–810.

[9] R. Asadi and A. Regan, "A convolution recurrent autoencoder for spatio-temporal missing data imputation," in *International Conference on Artificial Intelligence (ICAI 2019)*, 2019, pp. 206–213.

[10] Q. Zhu, J. Chen, L. Zhu, X. Duan, and Y. Liu, "Wind speed prediction with spatio–temporal correlation: A deep learning approach," *Energies*, vol. 11, no. 4, p. 705, 2018.

[11] C. Qiu, Y. Zhang, Z. Feng, P. Zhang, and S. Cui, "Spatio-temporal wireless traffic prediction with recurrent neural network," *IEEE Wireless Communications Letters*, vol. 7, no. 4, pp. 554–557, 2018.

[12] Y. Wang, M. Long, J. Wang, Z. Gao, and S. Y. Philip, "PredRNN: Recurrent neural networks for predictive learning using spatiotemporal LSTMs," in *Conference on Neural Information Processing Systems (NeurIPS 2017)*, 2017, pp. 879–888.

[13] D. Li, L. Li, X. Li, Z. Ke, and Q. Hu, "Smoothed lstm-ae: A spatio-temporal deep model for multiple time-series missing imputation," *Neurocomputing*, vol. 411, pp. 351–363, 2020.

[14] M. New, M. Hulme, and P. Jones, "Representing twentieth-century space–time climate variability. part II: Development of 1901–96 monthly grids of terrestrial surface climate," *Journal of Climate*, vol. 13, no. 13, pp. 2217–2238, 2000.

[15] Korea Expressway Corporation (KMA), "Vehicle Detection System (VDS) Dataset," Last accessed 1 August 2022. [Online]. Available: https://data.ex.co.kr/portal/speed/speedVDS

[16] Seoul Transport Operation & Information Service (TOPIS), "Monthly traffic speed report dataset," Last accessed 1 August 2022. [Online]. Available: https://topis.seoul.go.kr/refRoom/openRefRoom_1.do

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

[18] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 2016)*, 2016, pp. 265–283.