Vehicle Speed Data Imputation based on Parameter Transferred LSTM

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

Chaeyeon Cha
Ewha Womans University
Seoul, Republic of Korea
soccer970405@ewhain.net

Hyunggon Park
Department of Electronic and Engineering and Smart Factory Multidisciplinary Program
Ewha Womans University, Seoul, Republic of Korea
The Alan Turing Institute, London, The United Kingdom
hyunggon.park@ewha.ac.kr

Abstract

In this paper, we propose a traffic speed data imputation algorithm based on a parameter transferred Long Short-Term Memory (LSTM) architecture for vehicle to infrastructure (V2I) networks. We consider a scenario in V2I networks where an Road Side Units (RSU) on the road does not operate temporarily and thus the traffic speed data cannot be collected. This makes any services that rely on the traffic speed data at the RSU be unavailable. For uninterrupted and seamless V2I services, an efficient and low complexity data imputation algorithm is imperative. The proposed algorithm is based on the architecture that includes multiple LSTM layers with parameter transfers, which can explicitly take into account the spatio-temporal characteristics of the traffic speed data. By transferring parameters from one LSTM layer to its adjacent LSTM layer, the complexity associated with the algorithm can be significantly reduced. The proposed algorithm includes bidirectional imputation, which can further improve imputation accuracy. Our simulation and experiment results confirm that the time for training and data imputation of the proposed algorithm can be significantly reduced while maintaining imputation accuracy.

1 Introduction

Intelligent Transportation Systems (ITS) has been attracting considerable attention recently since it enables the innovative development of smart cities as well as effective management of transport systems [1–3]. The ITS technology can actively collect traffic information such as volume, speed, direction, crash rate from various types of sources and is able to process such information. Moreover, these advances have become the fundamental basis for service providers to provide data-driven services such as vehicle and road maintenance, long and short term traffic management, road safety, convenience services, etc. [4, 5]. For service providers, it is one of the most important goals to support seamless services to the vehicles even when unexpected network malfunctions occur. For this, the imputation of the data missed by such network malfunctions or network failure can play an essential role in providing seamless data-driven services.

It is often challenging, however, to recover missing data in the vehicle to infrastructure (V2I) networks because of the dynamic, complex, and stochastic characteristics of the traffic. Since the traffic data collected at Road Side Units (RSU) changes over time and locations (i.e., spatio-temporal domain), the

Machine Learning for Autonomous Driving Workshop at the 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

performance of data imputation significantly depends on environmental changes, making it difficult to simply use predetermined models or algorithms. The problem becomes even more challenging if the efficiency of the data imputation is taken into account. The time required for imputing the missing data may be prohibitively high if a large size of data needs to be processed, so that the imputed data may not be available when needed. Therefore, it is important to design efficient algorithms that can reduce the time required for data imputation without degrading the data imputation performance.

Since the entities RSU and On-Board Units (OBU) that compose V2I networks generally have only limited computing powers, the collected traffic data can be processed by cloud servers. However, if the low end-to-end latency requirement restricts the use of cloud servers, the cloud computing resources and storages can be placed at the edge of the vehicular network, even more decreasing the round-trip time of data processing [6]. In this paper, we refer to the edge server as a local server that can offload traffic data processing load from the cloud server and assume that the proposed data imputation algorithm is performed in the local server. The proposed data imputation algorithm uses the spatio-temporal characteristics of traffic data based on Long Short-Term Memory (LSTM) networks [7, 8], which have been adopted with success to solve the long-distance dependence in vehicle networks.

Contributions. In order to support uninterrupted and seamless V2I services even with RSU failures and the corresponding missing data, we propose an efficient traffic data imputation algorithm that can be deployed in the local server with the cached traffic dataset. This makes RSUs only perform simple operations, such as traffic data collection and store-and-forward delivery between the local server and OBUs. We adopt an LSTM network to learn the representation of time series traffic data and propose parameter transferred LSTM network for efficient spatial data imputation. The proposed algorithm can reduce training time that is critical for several V2I services, while maintaining the data imputation performance. For efficient algorithm design, we explicitly take into account the spatio-temporal characteristics of traffic data. Specifically, the temporal dependency of the traffic data can be efficiently captured by using LSTM based learning process. The learning process can be significantly expedited by redesigning the typical LSTM network as multiple LSTM layers with transferred parameters, because the spatial correlation of the traffic data enables the parameters trained in the learning process to be transferable. This eventually reduces the training and imputing time. Moreover, the spatial correlation permits the use of a simple linear interpolation for efficient data imputation, so that data imputation can be processed in multiple directions, i.e., both forward and backward imputations. This can significantly improve data imputation performance. The simulation and experiment results confirm that the proposed algorithm can significantly reduce the training and data imputation time while maintaining the imputation accuracy.

2 Related Works

The data imputation methods for missing traffic information can be classified into two categories: statical methods and deep learning (DL) based methods. The conventional statistical methods that include time-series approaches (e.g., Autoregressive Integrated Moving Average (ARIMA) [9, 10]) and probabilistic graphical models (e.g., Bayesian Network [11], Markov Chain [12]) are easy to adopt and show acceptable performance if the proportion of missing data is not significant. However, these methods can only show limited performance as more data is missing or is more complicated (e.g., dynamic patterns of data, non-linear data, etc.).

Alternatively, deep learning based methods have been proposed to impute the missing data, in particular, with non-linearity. LSTM is one of the general deep learning models that has been widely used for applications including time-series data such as speech recognition, weather forecasting, and traffic prediction [13–15]. Traffic data imputation can also be well-captured by LSTM networks for V2I applications [16, 17]. In [16], traffic flow prediction for missing data using multiscale LSTM (LSTM-M) is proposed, which shows satisfactory results for both short-time and long-time predictions. However, LSTM-M may generate constant values when 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 partially spatial is proposed based on transferred LSTM networks [17]. While these traffic 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 the data collector (e.g., RSUs).

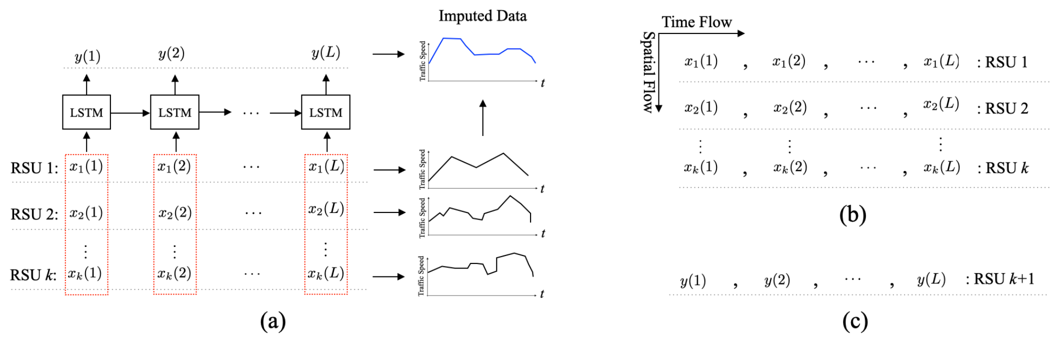


Figure 1: The architecture of general LSTM network (a) with spatio-temporal input data (b) and output data (c).

Table 1: Summary of data imputation techniques.

Ref.	Input Data Application	Methodology	Training Domain
[9]	Traffic flow data	(STAT) ARIMA	Temporal
[10]	Traffic flow data	(STAT) ARIMA	Spatio-temporal
[11]	Traffic flow data	(STAT) Markov chain	Temporal
[12]	Traffic speed data	(STAT) Bayesian Network	Spatio-temporal
[16]	Traffic flow data	(DL) LSTM-M	Temporal
[17]	Air pollution data	(DL) Transferred LSTM	Temporal & partially spatial
[18]	Traffic speed data	(DL) LSTM & Bi-LSTM	Spatio-temporal
[19]	Temperature data	(DL) Stacked LSTM	Spatio-temporal

In order to efficiently extract the representation of a large amount of missing data, both spatial and temporal relationship of the data needs to be considered. However, only a few studies have deployed LSTM based imputation for spatio-temporal missing data. In [18], 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 [19]. 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. For example, as shown in Figure 1, traffic data collected from different locations can be used to impute missing data based on LSTM with multiple layers. An LSTM layer is trained in the temporal domain after spatial patterns of each dataset are trained. This type of structure is considered as a general LSTM network structure for the data with spatio-temporal characteristics. Table 1 summarizes the approaches discussed above.

3 Traffic Management System in V2I Network

3.1 System Description

We consider V2I networks that consist of a cloud server, local servers, RSUs, and OBUs. The considered network has a hierarchical structure described as follows. RSUs upload the traffic data collected by OBUs to the cloud server through the local server. Similarly, the cloud server provides OBUs with V2I services using the traffic data uploaded from the local server and RSU. In the proposed system, we assume that the local server has limited storage so that it can store partial traffic data recently observed from OBUs. The overall description of the proposed V2I network is shown in Figure 2.

In this paper, each OBU is dynamically associated with RSUs as it moves, while each RSU is connected to the local server. An RSU denoted as r_k ($k \in K$), where K is an index set of RSUs, has its adjacent neighbor RSUs $r_{k-N}, \cdots, r_{k-1}, r_{k+1}, \cdots, r_{k+N}$. An RSU r_k collects traffic speed data of OBUs moving in a specific direction. The traffic speed data collected from r_k is denoted as $\mathbf{x}_k = [x_k(1), \cdots, x_k(L)]$, where $x_k(t) \in \mathbb{R}$ is the speed data at time t and L is the data length. Since the adjacent RSUs are placed at a fixed distance, $\mathbf{x}_{k-N}, \cdots, \mathbf{x}_{k+N}$ are correlated temporally and spatially.

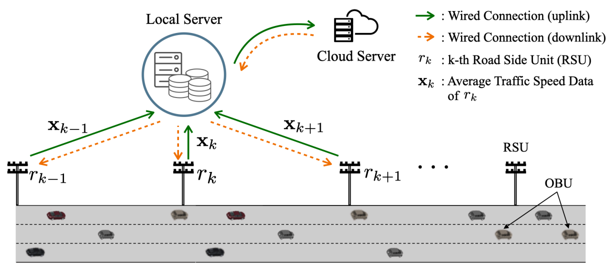


Figure 2: Overall description of the proposed V2I system.

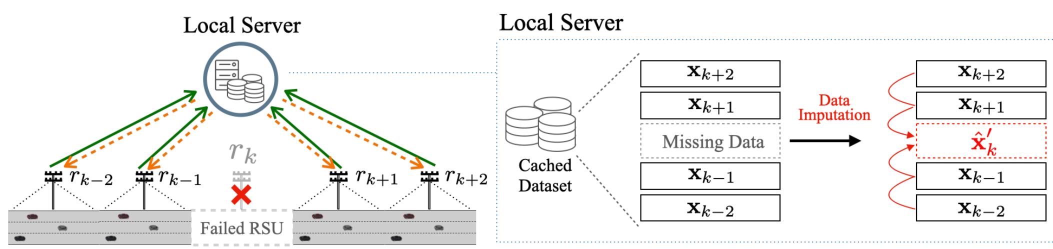


Figure 3: Illustrative examples of RSU failure scenario and data imputation.

3.2 Problem Formulation

We consider a scenario where the operation of an RSU is temporarily failed so that the local server cannot provide the traffic speed data collected at the failed RSU to OBUs. The missing data from the RSU can be recovered by traffic speed data imputation in a short period of time.

Let $\mathbf{x}_k$ be the lost traffic data of the failed RSU r_k. Since the traffic speed data collected at each RSU are temporally and spatially correlated, $\mathbf{x}_k$ lost in failed r_k can be imputed by using the data from adjacent RSUs. The proposed local server observes the traffic data collected from $2N$ adjacent RSUs (i,e., $\mathbf{x}_{k-N}, \cdots, \mathbf{x}_{k-1}, \mathbf{x}_{k+1}, \cdots, \mathbf{x}_{k+N}$) and recover the lost traffic data $\mathbf{x}_k$ based on the data imputation, as shown in Figure 3. Function $f : \mathbb{R}^{2N\times L} \to \mathbb{R}^L$ and $f' : \mathbb{R}^{2N\times L} \to \mathbb{R}^L$ denote the general LSTM network and the proposed data imputation algorithm discussed in Section 4, respectively. Then, the data $\hat{\mathbf{x}}_k = [\hat{x}_k(1), \cdots, \hat{x}_k(L)]$ and $\hat{\mathbf{x}}'_k = [\hat{x}'_k(1), \cdots, \hat{x}'_k(L)]$ imputed by them can be expressed as

$$\begin{aligned} \hat{\mathbf{x}}_k &= f(\mathbf{x}_{k-N}, \cdots, \mathbf{x}_{k-1}, \mathbf{x}_{k+1}, \cdots, \mathbf{x}_{k+N}), \\ \hat{\mathbf{x}}'_k &= f'(\mathbf{x}_{k-N}, \cdots, \mathbf{x}_{k-1}, \mathbf{x}_{k+1}, \cdots, \mathbf{x}_{k+N}). \end{aligned} \tag{1}$$

The goal is to reduce training and imputing time for data imputation while the performance deviation compared to the general LSTM network is limited. Given a performance measure $e : \mathbb{R}^{2\times L} \to \mathbb{R}$ and a maximum performance deviation ϵ, the proposed algorithm can reduce the training and imputing time T required to impute $\hat{\mathbf{x}}'_k$ with the constraint

$$|e(\hat{\mathbf{x}}'_k, \mathbf{x}_k) - e(\hat{\mathbf{x}}_k, \mathbf{x}_k)| \leq \epsilon$$

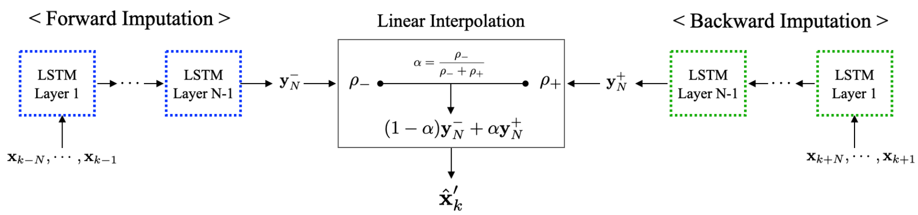
for dataset $\mathbf{x}_{k-N}, \cdots, \mathbf{x}_{k-1}, \mathbf{x}_{k+1}, \cdots, \mathbf{x}_{k+N}$. In this paper, we use the root mean square error (RMSE) as the performance measure $e(\cdot, \cdot)$, expressed as

$$e(\hat{\mathbf{x}}'_k, \mathbf{x}_k) = \sqrt{\frac{1}{L}\sum_{j=1}^{L}(\hat{x}'_k(j) - x_k(j))^2}. \tag{2}$$

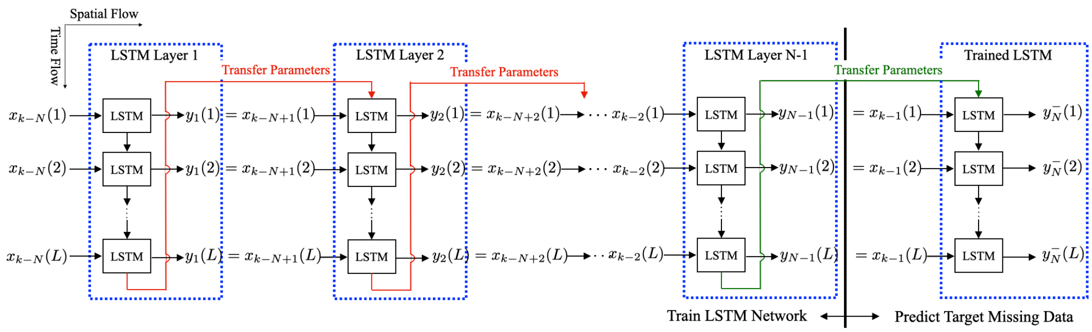
Thus, our proposed algorithm can provide OBUs with traffic speed data of failed RSU section using stable and efficient parameter transferred LSTM network.

4 Parameter Transferred LSTM

In this paper, we propose a data imputation architecture, referred to as parameter transferred LSTM, which aims at imputing dataset $\mathbf{x}_k$ missed by an RSU r_k using $2N$ datasets from its adjacent RSUs,



(a) Overall procedure of the proposed algorithm



(b) The architecture of the proposed parameter transferred LSTM network at forward imputation

Figure 4: The overall architecture of the proposed algorithm.

$\mathbf{x}_{k-N}, \cdots, \mathbf{x}_{k-1}, \mathbf{x}_{k+1}, \cdots, \mathbf{x}_{k+N}$. The overall procedure of the proposed parameter transferred LSTM is shown in Figure 4-(a), which consists of bidirectional imputations (i.e., forward and backward imputations) with parameter transfer. Each directional imputation is based on consecutive $(N-1)$ LSTM layers and the parameters trained from an LSTM layer are transferred to its next LSTM layer. Each LSTM layer predicts the traffic speed data for its next RSU in the spatial domain. For example, in forwarding imputation, $\mathbf{x}_{k-(N-l+1)}$ is used as input data to predict the traffic speed data $\mathbf{x}_{k-(N-l)}$ at the l-th LSTM layer.

Let $f_t^{[l]}$, $i_t^{[l]}$, $c_t^{[l]}$, $o_t^{[l]}$, and $h_t^{[l]}$ be the variables of the forget gate, input gate, cell state, output gate, and hidden state of l-th forwarding imputation LSTM layer at time t, which are expressed as [20]

$$
\begin{aligned}
f_t^{[l]} &= \sigma(W_f^{[l]} \cdot [h_{t-1}, x_{k-(N-l+1)}(t)] + b_f), \\
i_t^{[l]} &= \sigma(W_i^{[l]} \cdot [h_{t-1}, x_{k-(N-l+1)}(t)] + b_i), \\
c_t^{[l]} &= f_t^{[l]} * c_{t-1}^{[k]} + i_t^{[l]} * \tanh(W_c^{[l]}[h_{t-1}, x_{k-(N-l+1)}(t)] + b_i), \\
o_t^{[l]} &= \sigma(W_o^{[l]} \cdot [h_{t-1}, x_{k-(N-l+1)}(t)] + b_o), \\
h_t^{[l]} &= o_t^{[l]} * \tanh(c_t^{[l]}),
\end{aligned}
\tag{3}
$$

where $*$ denotes the Hadamard product, $W_f^{[l]}$, $W_i^{[l]}$, $W_c^{[l]}$, and $W_o^{[l]}$ denote the weight matrices of gates, b_f, b_i, and b_o denote bias vector parameters, and $\sigma(\cdot)$ is the sigmoid function. At each layer, mean square error (MSE) is used as a loss function, expressed as

$$
J(\mathbf{x}_{k-(N-l)}, \mathbf{y}_l) = \frac{1}{L}\sum_{t=1}^{L}(x_{k+l}(t) - y_l(t))^2 \tag{4}
$$

where $\mathbf{y}_l = [y_l(1), \cdots, y_l(L)]$ denotes the predicted traffic speed value in the l-th LSTM layer at time t. Note that the LSTM training procedures are the same for the backward imputation for predicting $\mathbf{x}_{k+(N-l)}$ from data $\mathbf{x}_{k+(N-l+1)}$ at the l-th backward LSTM layer.

The parameters of the LSTM network from the trained l-th LSTM layer are transferred to the next layer by passing the weight matrices, cell states, and hidden states, i.e.,

$$\begin{array}{l} W_f^{[l+1]} \leftarrow W_f^{[l]} \\ W_i^{[l+1]} \leftarrow W_i^{[l]} \\ W_c^{[l+1]} \leftarrow W_c^{[l]} \\ W_o^{[l+1]} \leftarrow W_o^{[l]} \end{array}, \ c_0^{[l+1]} \leftarrow c_L^{[l]}, \text{ and } h_0^{[l+1]} \leftarrow h_L^{[l]}. \tag{5}$$

The parameters trained from the l-th LSTM layer are transferred to the $(l+1)$-th LSTM layer as its initial parameters for training. This process repeats up to the $(N-1)$-th LSTM layer. This is shown in Figure 4-(b). The proposed parameter transferred LSTM architecture can significantly reduce the computation complexity by considerably decreasing the dimension of the LSTM parameters. Moreover, the parameters of the proposed LSTM can be efficiently initialized, thereby expediting the characterization of prevalent spatio-temporal traffic data patterns.

Given the two predicted outputs $\mathbf{y}_N^-$ and $\mathbf{y}_N^+$ from $\mathbf{x}_{k-1}$ with the forward imputation and $\mathbf{x}_{k+1}$ with backward imputation, respectively, $\mathbf{x}_k$ is determined by the linear interpolation using spatial correlation for efficient data imputation. Let ρ_- and ρ_+ be the average spatial correlation coefficients of the forward and backward input datasets, which are expressed as

$$\rho_- = \frac{1}{N-1}\sum_{l=1}^{N-1} \rho(\mathbf{x}_{k-l}, \mathbf{x}_{k-l-1}) \quad \text{and} \quad \rho_+ = \frac{1}{N-1}\sum_{l=1}^{N-1} \rho(\mathbf{x}_{k+l}, \mathbf{x}_{k+l+1}), \tag{6}$$

where

$$\rho(\mathbf{x}_i, \mathbf{x}_{i+1}) = \frac{\sum_{j=1}^{L}(x_i(j) - \overline{\mathbf{x}}_i)(x_{i+1}(j) - \overline{\mathbf{x}}_{i+1})}{\sqrt{\sum_{j=1}^{L}(x_i(j) - \overline{\mathbf{x}}_i)^2 \sum_{j=1}^{L}(x_{i+1}(j) - \overline{\mathbf{x}}_{i+1})^2}} \tag{7}$$

with $\overline{\mathbf{x}}_i = \frac{1}{L}\sum_{j=1}^{L} x_i(j)$. Then, the data imputation for missing data $\mathbf{x}_k$ is determined as a linear interpolation of $\mathbf{y}_N^-$ and $\mathbf{y}_N^+$, i.e.,

$$\hat{\mathbf{x}}_k' = (1-\alpha)\mathbf{y}_N^- + \alpha\mathbf{y}_N^+ \tag{8}$$

with a weight $\alpha = \frac{\rho_-}{\rho_- + \rho_+}$ computed by the average spatial correlation coefficients ρ_- and ρ_+.

5 Results and Discussions

In this section, we evaluate the performance of the proposed data imputation algorithm based on synthetic traffic data and actual traffic data collected from highways. The synthetic traffic data is randomly generated with spatial and temporal correlations. We also use the actual traffic data that is collected by Korea Expressway Corporation from January to April 2020 [21].

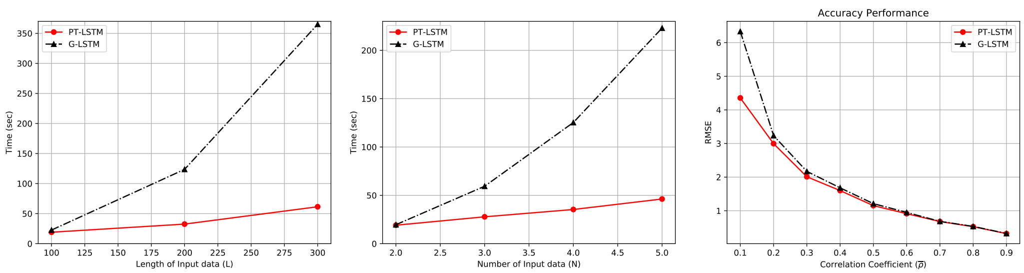
5.1 Simulation Results

Our focus of this section is on studying the impact of traffic data correlation, short-time, long-time dependency, and the number of traffic speed datasets on the imputation performance of the proposed algorithm. For simulations, we evaluate the performances of two data imputation algorithms, which are based on a general LSTM (G-LSTM) network discussed in Section 2, and the proposed parameter transferred LSTM (PT-LSTM) network.

The synthetic traffic data $\mathbf{x}_k = [x_k(1), \cdots, x_k(L)] (k \in K)$ is randomly generated from the Gaussian distribution $N(0, 1^2)$. The adjacent data $\mathbf{x}_{k+1}$ is also generated based on the Gaussian distribution but is temporally and spatially correlated with $\mathbf{x}_k$ such that $\rho(\mathbf{x}_k, \mathbf{x}_{k+1})$. We measure the accuracy in terms of RMSE and training time on average over 100 independent simulations. For both algorithms, the epoch during the training is set as 50. For simulations, the average correlation of data between RSUs is set as $\rho(\mathbf{x}_k, \mathbf{x}_{k+1}) \in \{0.1, 0.2, \cdots, 0.8, 0.9\}$ where data length and the number of adjacent RSU are set as $L = 200$ and $N = 4$ by default. For simplicity, the average spatial correlation coefficients of the forward and backward imputation are equally set, i.e., $\rho_+ = \rho_-$, so that the linear interpolation weight is $\alpha = 0.5$. For convenience, we simply denote the average spatial correlation coefficients by $\overline{\rho}$.

Table 2: Imputation performance (RMSE) of PT-LSTM and G-LSTM

	Length of input data (L), where $N = 4$				Number of input data (N), where $L = 200$				
$\overline{\rho}$	RMSE	100	200	300	RMSE	2	3	4	5
0.3	PT-LSTM	**1.7165**	**2.0112**	**2.5137**	PT-LSTM	**2.2152**	**2.1346**	**2.0112**	**1.9511**
	G-LSTM	2.2759	2.1715	2.8253	G-LSTM	2.4188	2.2321	2.1715	2.1638
0.6	PT-LSTM	**0.9123**	**0.9139**	**0.8928**	PT-LSTM	0.9857	0.9166	**0.9139**	**0.8810**
	G-LSTM	1.0035	0.9513	0.9506	G-LSTM	**0.9851**	**0.8473**	0.9513	0.9550



(a) $\overline{\rho} = 0.5, N = 4$ (b) $\overline{\rho} = 0.5, L = 200$ (c) $L = 200, N = 4$

Figure 5: Performance of PT-LSTM and G-LSTM

The simulation results for performance evaluation of the data imputation algorithms over the length of input data (L) and the number of spatially correlated input data (N) are shown in Figure 5 and Table 2. It is observed that most of the imputation accuracy measured by RMSE improves both for PT-LSTM and G-LSTM as the length of input data, the number of input data, and the average correlation coefficients increase. This is because LSTM networks can be better trained by such data sets, i.e., more data with higher correlation and less overfitting problem.

For the comparison between PT-LSTM and G-LSTM, the proposed PT-LSTM significantly reduces the training and imputation time, as shown in Figure 5-(a) and Figure 5-(b). It is clearly observed that PT-LSTM requires significantly lower training and imputation time than G-LSTM for all the ranges of input data lengths and the numbers of input data. This is because, unlike G-LSTM, PT-LSTM allows the parameters trained from one LSTM to be transferred to adjacent LSTMs, reducing the dimension of the LSTM parameters for the characterization of representative patterns for missing traffic data.

While the proposed PT-LSTM significantly reduces the training and imputation time, PT-LSTM still shows better or similar imputation accuracy than G-LSTM for all the range of correlations, as shown in Figure 5-(c). For example, PT-LSTM outperforms G-LSTM up to 45.42% at $\overline{\rho} = 0.1$. This implies that PT-LSTM can efficiently take into account the correlation in the traffic data for imputation, even for the data with relatively lower correlations.

Therefore, it can be concluded from the simulation results that the proposed PT-LSTM can impute data with lower complexity while maintaining data imputation accuracy.

5.2 Experiment Results

In this section, we examine the performance of the proposed PT-LSTM using the actual traffic speed dataset. The data of the traffic speed were collected every 5 minutes from February to April 2020. We use six regions of the highway and each region is covered by RSUs. The distance between RSUs is 8km. We evaluate the average performance of the algorithms on the same day of the week for three months. In the experiments, missing daily traffic data $\mathbf{x}_k = [x_k(1), \cdots, x_k(L)]$ with $L = 288$ is intentionally set in order to evaluate the performance of data imputation algorithms. The number of adjacent nodes is $N = 4$.

For performance comparison, we evaluate the performance based on four existing data imputation algorithms: CNN-LSTM [22], Bi-LSTM [23], GRU [15] and G-LSTM. CNN-LSTM [22] for traffic data imputation is an algorithm that captures spatio-temporal with convolutional layers while explicitly

Table 3: Imputation accuracy (RMSE)

RMSE	GRU [15]	Bi-LSTM [23]	CNN-LSTM [22]	G-LSTM	PT-LSTM
Region 1	5.7214	5.7673	5.7592	**5.7207**	7.2434
Region 2	7.9827	8.2233	6.7072	8.0161	**5.5162**
Region 3	14.0222	14.7628	11.6910	14.0843	**6.7772**
Region 4	7.1805	7.2891	**6.6919**	7.1950	6.8687
Region 5	7.7931	8.5033	8.2154	**7.7832**	8.9219
Region 6	7.4122	7.4037	**7.1821**	7.4197	13.4166

Table 4: Training and imputation time (seconds)

Time (sec)	GRU [15]	Bi-LSTM [23]	CNN-LSTM [22]	G-LSTM	PT-LSTM
Region 1	1002.35	1580.00	1398.78	1315.59	**225.84**
Region 2	1005.16	1575.63	1404.20	1307.77	**230.82**
Region 3	997.91	1575.74	1396.53	1309.96	**233.56**
Region 4	1003.08	1605.47	1407.67	1318.48	**218.56**
Region 5	1005.30	1580.73	1409.51	1322.49	**191.67**
Region 6	1009.90	1580.28	1410.73	1329.32	**223.03**
Average (Ratio [%])	1004.96 (455.60%)	1582.80 (717.56%)	1404.57 (636.75%)	1317.27 (597.18%)	**220.58** (100%)

considering various types of short-term and long-term patterns. The number of filters is set as 288 and the kernel size is set as 4. Bi-LSTM [23] is an algorithm that captures the relation of past and future data simultaneously and has a similar architecture with the proposed algorithm. GRU [15] can also be used as it is a modified LSTM for a simpler structure and convenience to solve [24]. G-LSTM is again used for performance comparison in the same way as shown in Section 5. All experiments with models discussed above are performed on the platform with NVIDIA GeForce RTX 2080 Ti GPUs.

Table 3 shows the imputation accuracy of the algorithms for the traffic data sets collected from different regions. 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.

In terms of time complexity for data imputation, the proposed PT-LSTM clearly shows the best performance as shown in Table 4. In these experiments, PT-LSTM requires approximately 220 seconds on average for all traffic data sets from all regions. However, other algorithms take significantly more time (i.e., approximately 4.5 times more for GRU, 6 times more for CNN-LSTM, G-LSTM, and 7 times more for Bi-LSTM) than PT-LSTM.

Hence, we can conclude that the proposed PT-LSTM is a very efficient traffic data imputation algorithm.

6 Conclusion

In this paper, we have proposed a data imputation algorithm based on parameter transferred LSTM architecture in V2I networks. In order to support seamless V2I services, the training and data imputation time is critical for the services while maintaining the data imputation performance. Therefore, we take into account the spatio-temporal characteristics of traffic data for efficient algorithm design. The learning process of the proposed PT-LSTM can be significantly reduced as the dimension of the LSTM parameters decreases by transferring the trained LSTM parameters to adjacent LSTMs. This enables every LSTM layers to skip redundant parameter training steps, thereby expediting the characterization of the representative traffic speed patterns. Moreover, the traffic speed data were imputed using linear interpolation that can improve imputation performance. Our simulation and experiment results confirm that the proposed algorithm can impute data with lower time complexity while maintaining data imputation performance by comparing with existing LSTM algorithms in terms of imputation accuracy and training and imputing time.

Acknowledgments and Disclosure of Funding

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2019-0-00024, Supervised Agile Machine Learning Techniques for Network Automation based on Network Data Analytics Function) and supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF2020R1A2B5B01002528).

References

[1] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big data analytics in intelligent transportation systems: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 383–398, 2018.

[2] J. Guerrero-Ibáñez, S. Zeadally, and J. Contreras-Castillo, "Sensor technologies for intelligent transportation systems," *Sensors*, vol. 18, no. 4, p. 1212, 2018.

[3] H. Menouar, I. Guvenc, K. Akkaya, A. S. Uluagac, A. Kadri, and A. Tuncer, "UAV-enabled intelligent transportation systems for the smart city: Applications and challenges," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 22–28, 2017.

[4] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1624–1639, 2011.

[5] N. Cheng, F. Lyu, J. Chen, W. Xu, H. Zhou, S. Zhang, and X. Shen, "Big data driven vehicular networks," *IEEE Network*, vol. 32, no. 6, pp. 160–167, 2018.

[6] J. Liu, J. Wan, B. Zeng, Q. Wang, H. Song, and M. Qiu, "A scalable and quick-response software defined vehicular network assisted by mobile edge computing," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 94–100, 2017.

[7] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[8] Z. Zhao, W. Chen, X. Wu, P. C. Chen, and J. Liu, "LSTM network: a deep learning approach for short-term traffic forecast," *IET Intelligent Transport Systems*, vol. 11, no. 2, pp. 68–75, 2017.

[9] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results," *Journal of Transportation Engineering*, vol. 129, no. 6, pp. 664–672, 2003.

[10] M. Van Der Voort, M. Dougherty, and S. Watson, "Combining Kohonen maps with ARIMA time series models to forecast traffic flow," *Transportation Research Part C: Emerging Technologies*, vol. 4, no. 5, pp. 307–318, 1996.

[11] X. Chen, Z. He, Y. Chen, Y. Lu, and J. Wang, "Missing traffic data imputation and pattern discovery with a Bayesian augmented tensor factorization model," *Transportation Research Part C: Emerging Technologies*, vol. 104, pp. 66–77, 2019.

[12] D. Huang, Z. Deng, L. Zhao, and B. Mi, "A short-term traffic flow forecasting method based on Markov Chain and Grey Verhulst model," in *IEEE Data Driven Control and Learning Systems (DDCLS)*, 2017, pp. 606–610.

[13] 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," *arXiv preprint arXiv:1701.02720*, 2017.

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

[15] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," in *in IEEE Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, 2016, pp. 324–328.

[16] 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.

[17] 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, p. 101092, 2020.

[18] Z. Cui, R. Ke, Z. Pu, and Y. Wang, "Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction," *arXiv preprint arXiv:1801.02143*, 2018.

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

[20] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," in *IET International Conference on Artificial Neural Networks (ICANN)*, 1999, pp. 850–855.

[21] Korea highway system, 2020, Korea Expressway Corporation. [Online] Available from: http://data.ex.co.kr/portal/fdwn/view?type=VDS&num=38&requestfromd%CC%84ataset [Accessed 15 July 2020].

[22] R. Asadi and A. C. Regan, "A spatio-temporal decomposition based deep neural network for time series forecasting," *Applied Soft Computing*, vol. 87, p. 105963, 2020.

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

[24] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.