Data Science for Transportation (2024) 6:22
https://doi.org/10.1007/s42421-024-00104-2

RESEARCH

An Imputation-Enhanced Hybrid Deep Learning Approach for Traffic Volume Prediction in Urban Networks: A Case Study in Dresden

Peng Yan[1] · Zirui Li[1] · Jyotirmaya Ijaradar[1] · Sebastian Pape[1] · Matthias Körner[1] · Meng Wang[1]

Received: 11 June 2024 / Revised: 26 July 2024 / Accepted: 1 August 2024 / Published online: 13 September 2024

Abstract
Advanced traffic management systems rely heavily on accurate traffic state estimation and prediction. Traffic prediction based on conventional road-based sensors faces considerable challenges due to spatiotemporal correlations of traffic flow propagation, and heterogeneous, error-prone, and missing data. This paper proposes a hybrid deep learning approach for online traffic volume prediction in an urban network. The approach ensembles the long short-term memory (LSTM) neural network and the convolutional neural networks (CNN) in a parallel way. In order to deal with missing data, a state-of-the-art Bayesian probabilistic imputation method is employed in the overall prediction pipeline. The hybrid traffic prediction structure can capture the spatiotemporal characteristics of traffic volume. The proposed prediction model is verified by the loop and infrared sensor data in the inner city network of the City of Dresden. Experimental results show that it can achieve superior volume prediction compared with baseline methods.

Keywords Spatiotemporal traffic data · Traffic prediction · Time series prediction · Data imputation · Traffic management

Introduction

Accurate traffic forecasting constitutes a fundamental component of intelligent transportation systems (Vlahogianni et al. 2014; Yin et al. 2021; Azfar et al. 2024). The goal of traffic forecasting is to predict future traffic characteristics, such as volume or speed, based on historical or observed traffic data (Guo et al. 2019). A reliable and accurate traffic prediction model is an indispensable component for advanced traffic management systems (ATMS) and advanced traveler information systems (ATIS), where traffic managers and travelers highly desire accurate and reliable traffic information (Wang et al. 2016; Ma et al. 2019; Furno et al. 2024).

In recent years, numerous works have been conducted, exploring diverse approaches to address the traffic prediction problem (Jiang and Luo 2022; Liu et al. 2024). Despite the numerous publications that recently appeared on this topic, a robust solution in practice is still missing due to several challenges (Ramana et al. 2023; Chen et al. 2021; Wang et al. 2023). Firstly, the underlying traffic flow dynamics are highly nonlinear, especially in congested states. The propagation of traffic flow in a road network exhibits strong spatio-temporal correlations (Ramana et al. 2023). Secondly, loop detectors are the most widely adopted traffic sensors in real-world systems, but they have reliability problems and are prone to erroneous and missing measurements (Chen et al. 2021). Thirdly, road operators have implemented different sensor types that have different mechanisms in traffic measurements. The measured quantities, the sptio-temporal resolutions, and the characteristics of error and missing values are of substantial difference between the sensors (Wang et al. 2023). Reliable and robust prediction models that address the aforementioned challenges are highly desirable

Peng Yan, Zirui Li have contributed equally to this work.

✉ Meng Wang
meng.wang@tu-dresden.de

Peng Yan
yanpengookk@163.com

Zirui Li
zirui.li@tu-dresden.de

Jyotirmaya Ijaradar
jyotirmaya.ijaradar@tu-dresden.de

Sebastian Pape
sebastian.pape@tu-dresden.de

Matthias Körner
matthias.koerner@tu-dresden.de

[1] Chair of Traffic Process Automation, Technische Universität Dresden, Hettnerstraße 3, 01069 Dresden, Saxony, Germany

for real-time ATMS and ATIS, and are the scope of this work.

Related Works

We first review the relevant work on traffic prediction, with a focus on data-driven methods (Vlahogianni et al. 2014; Lv et al. 2015). Data-driven approaches to traffic state forecasting are typically classified into parametric and nonparametric techniques. Among parametric methods, Bayesian networks (Ghosh et al. 2007), autoregressive integrated moving average models (ARIMA) (Lee and Fambro 1999; Zhong et al. 2004a, b), and support vector regression (SVR) (Castro-Neto et al. 2009) are frequently used for traffic prediction. Bayesian networks are renowned for their probabilistic graphical modelling capabilities, whereas ARIMA models are well-regarded for their robustness in univariate time series prediction, and SVR has been recognized for its effectiveness in regression problems. However, parametric models have several limitations. The primary limitation of parametric models lies in their assumption of a specific data distribution or structure, which can lead to model misspecification if the actual data do not conform to these assumptions. Additionally, parametric models may struggle with flexibility, as they are not designed to adapt to the complexities and nonlinearities present in traffic data. Nonparametric models have gained popularity in response to the constraints of parametric methods. Nonparametric approaches do not assume a particular model structure, offering greater flexibility in capturing the patterns and trends inherent in traffic data (Gershman and Blei 2012; Li et al. 2024).

Through the latest research in this field, the NNs are particularly promising for transportation research within the context of big data and the high dimensions of features (Do et al. 2019; Ye et al. 2020; Rahmani et al. 2023). Recurrent neural networks (RNNs) are a subclass of neural networks adept at handling sequential data. Unlike feed-forward neural networks, RNNs incorporate feedback connections, allowing information from previous steps to influence the network's future states. This characteristic is exemplified by the Elman and Jordan networks, where feedback originates from the hidden and output layers, respectively (Medsker and Jain 2001). For instance, Jordan networks have been applied to traffic prediction with notable success (Zhang 2000).

However, traditional RNNs face the vanishing gradient problem, which impairs their ability to learn long-term dependencies. Long Short-Term Memory (LSTM) networks address this issue with a gating mechanism that regulates memory updates, consisting of a cell and input, forget, and output gates (Ma et al. 2015). While LSTMs excel at capturing temporal patterns, as demonstrated by a three-layer LSTM traffic predictor (Ma et al. 2015), their application has often been confined to unidimensional time series, lacking in spatial correlation representation. In contrast, convolutional neural networks (CNNs) have revolutionized computer vision through their ability to process grid-like topologies, a feature increasingly exploited in short-term traffic prediction. Ma et al. (2017) proposed a two-dimensional CNN approach to predict the traffic network speed. CNNs automatically extract features from tensor-represented inputs with local correlations, achieved via convolution operations. With the development of deep learning, the attention mechanism was successfully applied in the traffic prediction. Wang et al. (2022) proposed to predict traffic flow using the spatial-temporal gated graph neural network. It demonstrated improved accuracy by capturing complex dependencies in traffic data. In Chen et al. (2023), a dynamic spatiotemporal graph attention network was developed based on the attention mechanism, which considered the macroscopic periodic characteristics of traffic flow. Similarly, Ye et al. (2023) presented a traffic flow predictor based on dynamic multi-graph neural network. It incorporated sudden traffic accidents into multi-step traffic flow prediction by constructing an accident-related adjacency matrix.

Moreover, several works have reported that hybrid models offer multiple benefits for time series forecasting, including improved prediction accuracy, better interpretability, and more effective handling of data limitations (Shah et al. 2022; Salman et al. 2024). In Zheng et al. (2006), a Bayesian combination neural network approach was used to combine two different neural network predictors: the back propagation neural network model and the radial basis function neural network model to predict short-term freeway traffic flow. As traffic data involves strong spatial-temporal correlations, the CNN-LSTM combination is a popular choice for traffic state estimation (Zhao et al. 2021; Wu and Tan 2016). Specifically, Wu and Tan (2016) proposed a deep architecture that inherits the advantages of both CNN and LSTM for short-term traffic flow prediction. Similarly, Cao et al. (2020); Islam et al. (2022); Rajalakshmi and Ganesh Vaidyanathan (2022) utilized a CNN-LSTM-based hybrid architecture to forecast traffic speed and flow.

Lastly, in practical scenarios, traffic data frequently contain missing values due to issues such as detector malfunction or communication loss, leading to gaps in traffic information that can undermine the effectiveness of traffic flow predictions (Asif et al. 2016; Liu et al. 2018). Imputing missing data before prediction has been shown to improve traffic forecasting performance (Boquet et al. 2019). Over the years, several methods have been proposed to impute missing data, ranging from basic statistical techniques to advanced deep learning models (Smith et al. 2003; Chan et al. 2023). Early methods include historical average (HA) (Smith et al. 2003), Autoregressive Integrated Moving Average (ARIMA) (Sharma et al. 2004), and principal component analysis (PCA) (Qu et al. 2009). However,

these methods primarily focus on temporal correlations and neglect spatial relationships. To address this issue, Chang et al. (2012) used an improved K-Nearest Neighbor (KNN) method incorporating spatiotemporal information to impute traffic volume data in Beijing. Additionally, Li et al. (2020) presented a hybrid model combining a prophet model for temporal properties and an iterative random forest model for spatial dependencies. Moreover, tensor factorization methods have been widely used for traffic data imputation by leveraging low-rank properties of both temporal and spatial data (Ran et al. 2015; Chen et al. 2019a, b; Xue et al. 2024). Where, Ran et al. (2015) utilized a low-rank tensor completion method to impute missing traffic data, effectively capturing spatio-temporal correlations. Chen et al. (2019a, 2019b) incorporated a Bayesian approach into the tensor decomposition model for traffic data imputation. Specifically, Chen et al. (2019a) proposed a Bayesian probabilistic matrix factorization method that utilizes a third-order tensor structure to capture the underlying dependencies among different dimensions for incomplete spatiotemporal traffic data. Furthermore, Lyu et al. (2024) introduced novel Tucker Imputation for addressing missing values in spatio-temporal traffic data that combines tensor factorization with rank minimization, effectively capturing key traffic dynamics without exhaustive rank tuning. The model incorporates time series decomposition to handle trends, spatio-temporal correlations, and outliers, enhancing the robustness of imputation results. While tensor factorization methods have shown promising performance in handling complex traffic data imputation tasks, they require low-rank traffic data and retraining for new missing data. Deep learning methods, on the other hand, have emerged as a popular alternative due to their ability to effectively capture spatiotemporal correlations in traffic data.

In recent years, there have been significant advancements in deep learning-based methods for traffic data imputation, with researchers employing various architectures such as Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), Generative Adversarial Networks (GAN), Graph Neural Networks (GNN), and attention mechanisms (Chan et al. 2023). Huang et al. (2023) proposed a GAN-based framework for traffic sensor data imputation, incorporating a novel time-dependent encoding method called GASF. This method converts time-series data into images, enabling improved GAN-based generation and achieving high imputation accuracy even with significant data loss. Tian et al. (2023) combined the stacked autoencoder (SAE) with GAN for traffic data imputation. The SAE extracts spatiotemporal features from incomplete traffic data, while the GAN generates complete traffic data based on these extracted features. Another popular approach is the use of GNN, as road networks can be treated as graphs, capturing strong spatiotemporal correlations. Liang et al. (2022) developed a Memory-augmented Dynamic Graph Convolution Networks (MDGCN) that combines a memory network, LSTM, and GCN to capture dynamic spatial correlations for traffic data imputation. Zhang et al. (2021) introduced an imputation model combining Graph Convolutional Networks (GCN) and GRU to handle missing traffic data and forecast traffic states. Ye et al. (2021) employed an encoder-decoder structure with Graph Attention Convolutional Networks (GACN) to fill in missing traffic data, leveraging graph attention mechanisms for spatial features and temporal convolutional layers for temporal features. Xue et al. (2024) introduced NMFD-GNN, a physics-informed machine learning method integrating the network macroscopic fundamental diagram (NMFD) with graph neural networks (GNN) for traffic state imputation. This approach captures spatio-temporal dependencies while adhering to traffic flow theories and outperformed six baseline models in real-world experiments on Zurich and London traffic networks using the UTD19 dataset.

Objectives and Contributions

From the existing literature on traffic state prediction, it is evident that traffic data exhibits both temporal and spatial correlations. While LSTM models excel at capturing temporal correlations, CNNs are effective at deciphering spatial patterns. Combining the strengths of LSTM and CNN has shown promise in enhancing traffic prediction performance. The development of such hybrid models is increasingly gaining researchers' interest. However, in previous studies, researchers often paired LSTM layers with one-dimensional CNN layers. This configuration causes the neural networks to process the two types of correlations separately, which may limit their effectiveness.

This paper proposes an imputation-enhanced hybrid traffic volume prediction model. In particular, we put forward a data imputation-enhanced traffic volume prediction method, where a data imputation method based on Bayesian Gaussian tensor decomposition is integrated into the prediction model (Chen et al. 2019a). The imputation method has the advantage of dealing with the original incomplete dataset and the sparsity issue. To generate robust prediction performance, we proposed a hybrid deep learning approach, which combines a long short-term memory (LSTM) neural network and a convolutional neural network (CNN). The parallel architecture ensures simultaneous processing of the input data matrix by both LSTM and two-dimensional CNN layers, which explores the superiority of LSTM and CNN in temporal and spatial modeling, respectively. After feature extraction by the LSTM layers and CNN layers, the cells are merged with a concatenate layer, where the data from LSTM and CNN are stitched together. The proposed architecture guarantees that the two-dimensional CNN extracts the

spatiotemporal features in the input data, while the LSTM enhances the understanding of temporal correlations.

Systematic experiments with real-world traffic detector data in Hangzhou, China and Dresden, Germany demonstrate that BGCP can impute missing data accurately. Raw road sensor data from the Traffic Analysis, Management and Optimisation System (Verkehrs Analyse, Management und Optimierungs System—VAMOS) of the city of Dresden with heterogeneous sampling time, missing values and unreliable measurements are used for verifying the prediction model performance. By comparing with the baseline prediction methods, our proposed hybrid prediction architecture can generate robust performance in terms of different history input lengths and prediction lengths for the Dresden city network.

In the sequel, we first revisit the imputation method and then present the neural network-based hybrid prediction model, followed by the experimental design and verification results of the imputation method and the prediction model performance.

The Proposed Prediction Model

This paper proposes an imputation-enhanced hybrid traffic volume prediction model. As shown in Fig. 1, raw traffic data from VAMOS is fed into the BGCP model to impute missing data (Chen et al. 2019a). The VAMOS data we were dealing with has diverse missing patterns. BGCP was chosen over other imputation techniques due to its superior performance in handling non-linear relationships and its reliability and transferability for spatiotemporal traffic data imputation tasks in a wide range of datasets (Chen et al. 2019a). The imputed data is then fed into the hybrid parallel prediction model consisting of LSTM and CNN networks. The newly proposed parallel hybrid model enables online traffic volume prediction. We discuss the details of the different modules of the prediction model in the remainder of the section.

Revisit of the BGCP Method

BGCP is an extension of Bayesian matrix factorization model in high order tensor, which applies Bayesian Gaussian Process in the data generation of CANDECOMP/PARAFAC (CP) tensor decomposition (Salakhutdinov and Mnih 2008; Chen et al. 2019a). Traffic matrix factorization involves decomposing a traffic data matrix into multiple lower-dimensional factors. This decomposition captures the underlying patterns and structures in the traffic data. In BGCP, this is achieved using tensor decomposition techniques, specifically the CP model. The CP model represents the multi-dimensional traffic data as a sum of component rank-one tensors, making it easier to analyze and manipulate the data. The underlying assumption

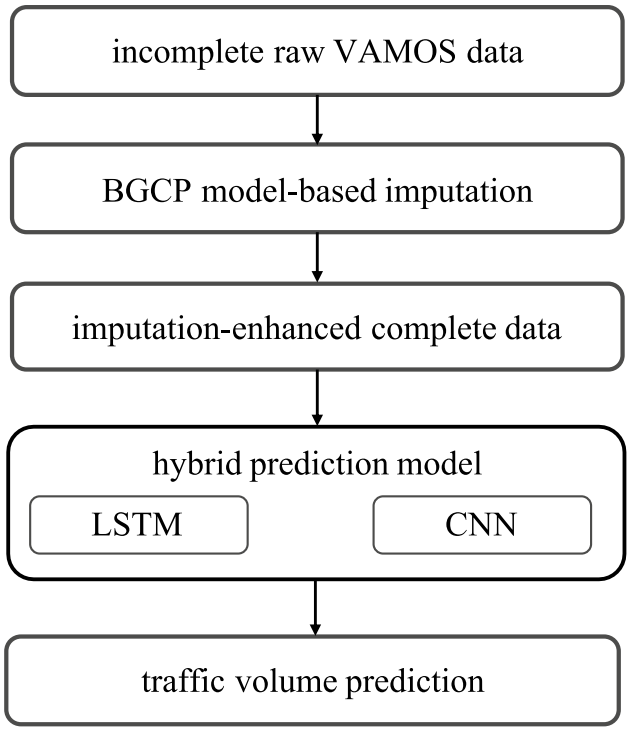


Fig. 1 The overall illustration of the proposed framework

for incomplete traffic data in the BGCP algorithm is that the observed traffic data can be modeled as a low-rank tensor with missing entries. This assumption is based on the idea that the traffic data matrix or tensor has an inherent low-dimensional structure, despite being high-dimensional and sparse due to missing values.

Let $\mathbf{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ donates a *dth* tensor, where n_k is the dimension along the *kth* way. In tensor $\mathbf{X}$, $x_{\mathbf{i}}$ is applied to represent the entry with index $\mathbf{i} = (i_1, i_2, ..., i_d)(1 \leqslant i_k \leqslant n_k, \forall k \in \{1, 2, ..., d\})$. The motivation of CP decomposition is to estimate $\mathbf{X}$ by the low-rank structure,

$$\hat{\mathbf{X}} = \sum_{r=1}^{R} \mathbf{u}_r^{(1)} \otimes \mathbf{u}_r^{(2)} \otimes \cdots \otimes \mathbf{u}_r^{(d)} \tag{1}$$

with factor matrix:

$$\boldsymbol{U}^{(k)} = \begin{bmatrix} | & | & & | \\ \mathbf{u}_1^{(k)} & \mathbf{u}_2^{(k)} & \cdots & \mathbf{u}_R^{(k)} \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{n_k \times R}, k = 1, 2, \ldots, d \tag{2}$$

where $\otimes$ is the outer product and $\mathbf{u}_r^{(1)} \otimes \mathbf{u}_r^{(2)} \otimes \cdots \otimes \mathbf{u}_r^{(d)}$ is the rank-one matrix. R is the CP rank of tensor $\hat{\mathbf{X}}$. The entry $x_{\mathbf{i}}$ can be formulated as follow:

$$x_{i_1,i_2,\ldots,i_d} = \sum_{r=1}^{R} u_{i_1,r}^{(1)} \times u_{i_2,r}^{(2)} \times \cdots \times u_{i_d,r}^{(d)} = \sum_{r=1}^{R} \prod_{k=1}^{d} u_{i_k,r}^{(k)} \tag{3}$$

where $u_{i_k,r}^{(k)}$ is the value at $i_k th$ row and rth column in the kth factor matrix. In the CP decomposition, tensor $\mathbf{X}$ is incomplete with missing elements and the indexes of observed entries is denoted as Ω. In BGCP, a fully Bayesian model is introduced for the missing data generation, which assumes observed entries in $\hat{\mathbf{X}}$ follows the Gaussian distribution,

$$x_{\mathbf{i}} \sim \mathcal{N}(\hat{x}_{\mathbf{i}}, \tau_\epsilon^{-1}) \tag{4}$$

where $\mathcal{N}(\cdot)$ is the Gaussian distribution and τ_ϵ is the precision. Similarly, the factor matrices also follow the multivariate Gaussian distributions:

$$\mathbf{u}_{i_k}^{(k)} \sim \mathcal{N}\left(\boldsymbol{\mu}^{(k)}, \left(\Lambda^{(k)}\right)^{-1}\right) \tag{5}$$

where $\boldsymbol{\mu}^{(k)} \in \mathbb{R}^r$ and $\Lambda^{(k)} \in \mathbb{R}^{r\times r}$ are hyper-parameters, which is modeled by conjugate Gaussian–Wishart priors:

$$\begin{aligned}&\left(\boldsymbol{\mu}^{(k)}, \Lambda^{(k)}\right) \sim \text{Gaussian–Wishart}\left(\boldsymbol{\mu}_0, \beta_0, W_0, \nu_0\right),\\ &p\left(\boldsymbol{\mu}^{(k)}, \Lambda^{(k)}|-\right) = \mathcal{N}\left(\boldsymbol{\mu}^{(k)}|\boldsymbol{\mu}_0, \left(\mathcal{B}_0\Lambda^{(k)}\right)^{-1}\right) \times \text{Wishart}\left(\Lambda^{(k)}|W_0, \nu_0\right)\end{aligned} \tag{6}$$

where $\beta_0 > 0$ is the coefficient of $\Lambda^{(k)}$. A Wishart distribution with ν_0 degrees of freedom and scale matrix W_0 is given by

$$\text{Wishart}\left(\Lambda^{(k)}|W_0, \nu_0\right) = \frac{1}{C}|\Lambda^{(k)}|^{\frac{\nu_0-r-1}{2}} \exp\left\{-\frac{1}{2}\text{tr}(W_0^{-1}\Lambda^{(k)})\right\} \tag{7}$$

where $\text{tr}(\cdot)$ is the trace function. For the precision τ_ϵ, a flexible conjugate Gamma prior is applied over τ_ϵ to improve the robustness of the model:

$$\tau_\epsilon \sim \text{Gamma}\left(a_0, b_0\right) \tag{8}$$

where a_0 and b_0 are the shape parameter and rate parameter, respectively. In the inferring process, the Gibbs sampling algorithm is introduced to estimate the hyper-parameters and generate missing values. For more details of the Gibbs sampling and the complete BGCP method, we refer readers to Chen et al. (2019a), Salakhutdinov and Mnih (2008).

Long Short-Term Memory (LSTM) Neural Network

LSTM is a network with a long-term memory function consisting of a forgetting gate, input data, and output date (Hochreiter and Schmidhuber 1997), as shown in Fig. 2. The forward process of LSTM could be represented as (9)–(13). The forget gate decides what information to discard in the cell state and updates the cell state (9–10). The input gate decides what new information to store in the cell state (11). The output gate controls the output information based on the cell state (12)–(13).

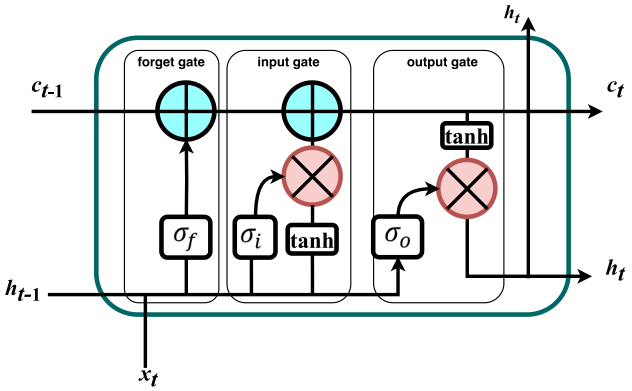


Fig. 2 The illustration of LSTM

At time step t, the input and output of the LSTM hidden layer are x_t and h_t, and the memory unit is c_t. The forget gate is formulated as follow:

$$f_t = \sigma_f\left(\mathbf{W}_{xf}\mathbf{x}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1} + \mathbf{b}_f\right) \tag{9}$$

where $\mathbf{b}_f$ is the bias, $\mathbf{W}_{xf}$ and $\mathbf{W}_{hf}$ are the weight matrices in forget gate, $\mathbf{h}_{t-1}$ is the output from the last timestep and $\mathbf{x}_t$ is the input at the current timestep. The forget gate uses a Sigmoid activation function σ_f, which outputs values between 0 and 1. These values are multiplied element-wise to the previous cell state. A value close to 0 means forgetting this completely, while a value close to 1 means retaining this completely. Then, the cell state is updated by:

$$\mathbf{c}_t = \mathbf{f}_t * \mathbf{c}_{t-1} + \mathbf{i}_t * tanh(\mathbf{W}_{xc}\mathbf{x}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1} + \mathbf{b}_c) \tag{10}$$

where $\mathbf{W}_{xc}$ and $\mathbf{W}_{hc}$ are weights. $\mathbf{b}_c$ is the bias in the update of cell state. The old cell state $\mathbf{c}_{t-1}$ is multiplied by f_t from the forget gate, deciding which parts of the old state to keep. New candidate values, generated by a tanh layer, are scaled by the output of the input gate i_t. These values are then added to the state. The input gate controls which values of the candidate state are allowed to be added to the cell state. The input gate is detailed as follow:

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

where $\mathbf{W}_{xi}$ and $\mathbf{W}_{hi}$ are weights. $\mathbf{b}_c$ is the bias in the update of input gate. It uses a Sigmoid function to output values between 0 and 1, effectively acting as a filter for the candidate values. The output gate is formulated in the similar way with weight matrices $\mathbf{W}_{xo}$, $\mathbf{W}_{ho}$ and $\mathbf{b}_o$.

$$\mathbf{o}_t = \sigma\left(\mathbf{W}_{xo}\mathbf{x}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{b}_o\right) \tag{12}$$

$$\mathbf{h}_t = \mathbf{o}_t * tanh(\mathbf{c}_t) \tag{13}$$

Finally, the hidden state $\mathbf{h}_t$ is calculated by multiplying the output gate's activation $\mathbf{o}_t$ with the tanh of the updated cell state ($\mathbf{c}_t$. This step filters the cell state through the output gate, allowing the LSTM to control what information is passed along to the output layer or the next time step in the sequence. Each component (gate) of the LSTM has a specific role in regulating the flow of information, making the network capable of learning which data is relevant to retain or discard over long sequences. This design helps mitigate the vanishing gradient problem common in traditional recurrent neural networks.

Convolutional Neural Networks (CNN)

CNN, as multilayer perceptrons, are commonly used to analyze visual images LeCun et al. (1989, 1998). Figure 3 shows the structure of a two-dimensional CNN in the context of traffic state prediction with four main parts: model input, traffic feature extraction, prediction and model output (Cui et al. 2020). The model input is the image-like feature generated from a traffic network with spatiotemporal characteristics. Let $\mathbf{X} \in \mathbb{R}^{T \times P}$ donate the input tensor, where T is the number of historical timestamps and P is the number of road sensors.

The extraction of the spatiotemporal features is the combination of convolutional and pooling layers and is also the core part of the CNN model. In this step, the processes in the convolutional layer and pooling layer are worth to be described, and before discussing the explicit layers, it should be noted that each layer is activated by an activation function. The benefits of employing the activation function are as follows: (1) the activation function transforms the output to a manageable and scaled data range, which is beneficial to model training; (2) the combination of the activation function through layers can mimic very complex nonlinear functions, making the CNN powerful enough to handle the complexity of a traffic network. Similar to Cui et al. (2020), the Relu function is applied. It is defined as follows:

$$\sigma(x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

where x is the element in $\mathbf{X}$. Convolutional layers differ from traditional feedforward neural networks where each input cell is connected to each output cell and the networks are fully connected. The CNN uses convolutional filters over its input layer and obtains local connections where only local input cells are connected to the local output cells. Hundreds of filters are sometimes applied to the input and results are merged in each layer. One filter can extract one traffic feature from the local input layer. Those extracted traffic features are combined further to extract a higher level and more abstract traffic features. This process confirms the compositionality of the CNN, meaning each filter composes a local path from lower-level into higher-level features. When one convolutional filter $\mathbf{W}_l^r$, where r is the index of the multiple convolutional filters and l is the index of the multiple convolutional layers in the CNN, is utilized to the local input, the output can be calculated as:

$$y_{\text{conv}} = \Sigma_{e=1}^{m} \Sigma_{f=1}^{n} \left(\mathbf{W}_{l,(e,f)}^{r} \mathbf{d}_{(e,f)} \right) \tag{15}$$

where m and n are two dimensions of the filter, $\mathbf{d}_{(e,f)}$ denotes the data value of the input matrix at position e and f, and $\mathbf{W}_{l,(e,f)}^{r}$ is the coefficient of the convolutional filter. y_{conv} is the local output of this convolutional layer after the processing of this filter. Pooling layers are designed to downsample and aggregate data because they only extract salient numbers from the specific region. The pooling layers guarantee that CNN is locally invariant, which means that the CNN can always extract the same feature from the input, regardless of feature shifts, rotations, or scales (LeCun et al. 1995). Based

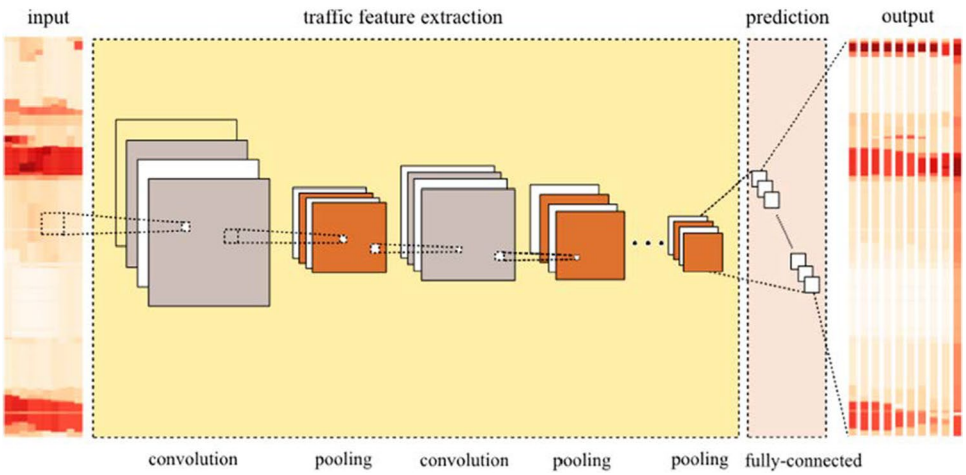


Fig. 3 Deep learning architecture of CNN in the context of transportation (Cui et al. 2020)

on the above mentioned facts, the pooling layers can not only reduce the network scale of the CNN but also identify the most prominent features of input layers. In our paper, the maximum operation is applied as follows:

$$y_{\text{pool}} = \max\left(\mathbf{d}_{(e,f)}\right),\ e \in [1,p], f \in [1,q] \tag{16}$$

where p and q are two dimensions of pooling window size, $\mathbf{d}_{(e,f)}$ is the data value of the local input matrix of this pooling layer. y_{pool} is the pooling output.The pooling procedure of *lth* is formulated as follow:

$$\mathbf{O}_l = \text{pool}\left(\sigma\left(\sum_{k=1}^{c_l} \mathbf{W}_k \cdot \mathbf{x}_k + \mathbf{b}_k\right)\right) \tag{17}$$

where the input, output and parameters of *lth* layer are denoted by $\mathbf{x}_k$, $\mathbf{O}_l$ and $(\mathbf{W}_k, \mathbf{b}_k)$. In the process of useful feature extraction from traffic volume data, the spatiotemporal correlations can be analyzed simultaneously as the convolutional filters have both temporal and spatial dimensions in the process of convolution.

The Hybrid Prediction Model

We developed a parallel hybrid prediction model for traffic volume by combining the strengths of LSTM and CNN to effectively handle both temporal and spatial correlations. The model takes the imputed traffic data as input, shaped as (T, O), where T represents the number of time steps and O denotes the output size. This input is simultaneously fed into two separate blocks: the LSTM block and the CNN block. The LSTM block is designed to capture the temporal dependencies in the traffic data. It includes two LSTM layers as previously explained in Section 2.2. The first LSTM layer has 64 units and is configured to return sequences. The output is then activated using the tanh function and followed by a dropout layer with a rate of 0.5 to prevent overfitting. The second LSTM layer, which has 256 units, further processes the output from the first LSTM layer. The larger number of units in the second layer enhances the model's ability to capture more complex temporal dependencies. This layer also uses the tanh activation function and is followed by another dropout layer with a rate of 0.5.

Simultaneously, the CNN block processes the spatial features of the traffic data. Initially, the input data is reshaped to $(T, O, 1)$ to fit the expected input shape for CNNs. This path includes three convolutional layers (C), each followed by a max-pooling layer and ReLU activation function, as detailed in Section 2.3. The first convolutional layer ((C_1)) has 256 filters with a kernel size specified by K[0], followed by a max-pooling layer ((P_1)) that reduces the spatial dimensions. The second convolutional layer (C_2) has 128 filters with a kernel size defined by K[1], also followed by a max-pooling layer ((P_2)). The third convolutional layer (C_3) has 64 filters with a kernel size specified by K[2], and it too is followed by a max-pooling layer (P_3). The output of the last convolutional layer is then flattened into a vector (F). The outputs from the LSTM and CNN blocks are concatenated to form a combined feature vector. This concatenated vector captures both the temporal dependencies and spatial features of the traffic data. Finally, this vector is passed through a dense layer with O units to produce the final predicted traffic volume. The detail of Concatenate and dense layer calculation provided in Eqs. 18 and 19.

$$H_{\text{concat}} = \text{Concatenate}\left(H_2, F\right) \quad H_{\text{concat}} \in \mathbb{R} \tag{18}$$

$$Y = \text{Dense}\left(H_{\text{concat}}, O\right) \quad Y \in \mathbb{R}^O \tag{19}$$

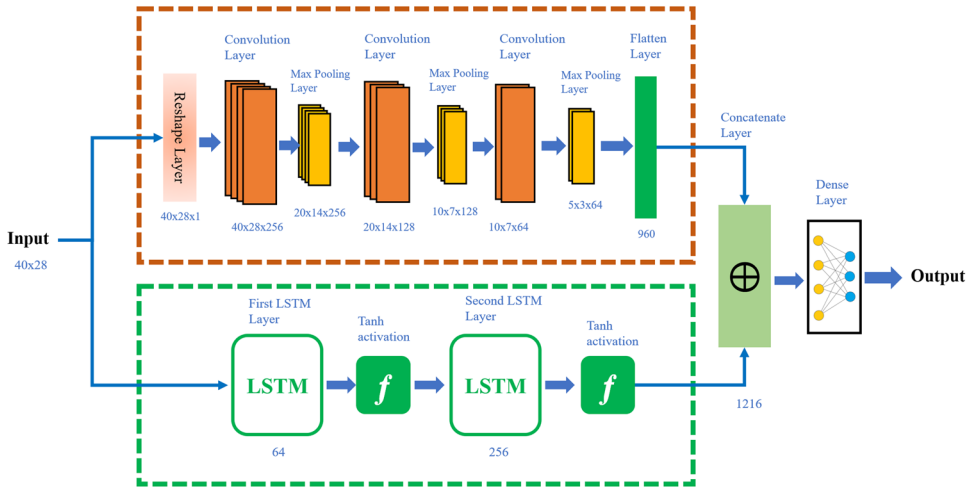


Fig. 4 The structure of the proposed hybrid model

Algorithm 1 Parallel Hybrid Traffic Volume Prediction Model

Input: Time steps T, Output size O, Kernel sizes K
Output: Trained model

function BUILDMODEL(T, O, K)
  $\mathbf{X} \in \mathbb{R}^{T \times O}$ (Traffic data)
  $\mathbf{ip} \leftarrow \text{Input}(\text{shape} = (T, O))$
LSTM Block:
  $\mathbf{h}_1^t \leftarrow \text{LSTM}(64, \text{return\_sequences=True})(\mathbf{ip})$
  $\mathbf{a}_1^t \leftarrow \tanh(\mathbf{h}_1^t)$
  $\mathbf{d}_1^t \leftarrow \text{Dropout}(0.5)(\mathbf{a}_1^t)$
  $\mathbf{h}_2 \leftarrow \text{LSTM}(256)(\mathbf{d}_1^t)$
  $\mathbf{a_2} \leftarrow \tanh(\mathbf{h}_2)$
  $\mathbf{d_2} \leftarrow \text{Dropout}(0.5)(\mathbf{a}_2)$
CNN Block:
  $\mathbf{Y} \leftarrow \text{Reshape}((T, O, 1))(\mathbf{ip})$
  for $i = 1$ ***to*** 3 **do**
    $\mathbf{C}_i \leftarrow \text{Conv2D}(256/2^{i-1}, \text{kernel\_size} = K[i-1], \text{padding} = '\,same')(\mathbf{Y})$
  $\mathbf{P}_i \leftarrow \text{MaxPooling2D}(2)(\mathbf{C}_i)$
  $\mathbf{A}_i \leftarrow \text{ReLU}(\mathbf{P}_i)$
  $\mathbf{Y} \leftarrow \mathbf{A}_i$
  $\mathbf{F} \leftarrow \text{Flatten}(\mathbf{A}_3)$
Concatenate and Output:
  $\mathbf{C} \leftarrow \text{Concatenate}([\mathbf{d}_2, \mathbf{F}])$
  $\mathbf{output} \leftarrow \text{Dense}(O)(\mathbf{C})$
  $\mathbf{model} \leftarrow \text{Model}(\mathbf{ip}, \mathbf{output})$
  return model
end function

A detailed illustration of how each component is connected and interacts with each other is provided in Fig. 4, which includes the shapes and parameter sizes of each layer, and the descriptions of each notation are in Table 1. All hyperparameters of this model were chosen based on extensive preliminary experiments, hyperparameter tuning processes, and literature studies. For instance, the selection of 256 units for the LSTM layers ensures sufficient capacity to capture complex temporal dependencies in the traffic data. The dropout rate of 0.5 effectively prevents overfitting

Table 1 Notations used in the model description

Notation	Description
X	Input traffic data shaped as (T, O)
T	Number of time steps
O	Output size (number of traffic features)
H_1	Intermediate output from the first LSTM layer
H_2	Intermediate output from the second LSTM layer
X_r	Reshaped input for CNNs with dimensions $(T, O, 1)$
C_1, C_2, C_3	Outputs from the three convolutional layers
P_1, P_2, P_3	Outputs from the max-pooling layers
F	Flattened output from the final convolutional block
H_{concat}	Concatenated vector combining LSTM and CNN features
Y	Final output (predicted traffic volume)

without significantly hindering the model's learning ability. The choice of 128 and 256 filters in the convolutional layers is based on the need to extract robust spatial features at different scales, allowing the model to handle various patterns in the traffic data. Additionally, other hyperparameters like batch size and learning rate, detailed in the experiment section, were fine-tuned to balance model complexity and performance, ultimately leading to improved prediction accuracy and generalization capabilities. The details of our model construction and pseudocode are presented in Algorithm 1.

Experiments

To validate our proposed methodology, we have formulated a comprehensive experiment plan, that includes systematic processing of raw data to deploying an optimized traffic prediction model. As illustrated in Fig. 5, we initially address the challenge of missing entries in the raw data using the BGCP Imputer, thus preparing a complete dataset for subsequent analysis. This dataset is then meticulously divided into separate sets for training, validation, and testing to establish a solid foundation for model development and rigorous evaluation. The main part of our experiment involves constructing the four aforementioned prediction models and tuning the hyperparameters based on their evaluation to obtain the best-trained model. Subsequently, the performance of these models was compared, and the best one was deployed. In the following section, we discuss the dataset, evaluation criteria, and experimental results in detail.

Data Description

We used our in-house VAMOS dataset for our experiments. VAMOS (Verkehrsanalyse-, Management- und Optimierungssystem) is a urban traffic management system developed in collaboration with City of Dresden and the Saxon State Road Administration. This system utilizes an extensive network of over 1800 traffic detectors, including loops, infrared sensors, floating vehicles, and roadside units (RSUs) to facilitate around-the-clock traffic monitoring and management. For this study, we consider a subset of the data from 28 sensors, comprising loop detectors and infrared sensors. The artery for data collection is shown in Fig. 6a. This dataset provides traffic volume and speed measurements at one-minute intervals. The distribution of the samples in the VAMOS dataset is illustrated in Fig. 6b.

Evaluation metrics

The selection of appropriate evaluation methods for both imputation and prediction is crucial. Following the approach outlined in Yin et al. (2021), we consider three widely used quantitative predictive error-based performance metrics: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE). The MSE is a performance metric that calculates the average of the squared differences between the actual and predicted values. Its differentiability and emphasis on larger errors make it well suited for optimization tasks, although it remains sensitive to outliers. RMSE is a regression metric representing the square root of the average squared difference between the actual and predicted values. Its emphasis on larger errors and rescaling to the original units makes it especially useful when significant deviations are undesirable. MAPE expresses the average forecast error as a percentage of the actual

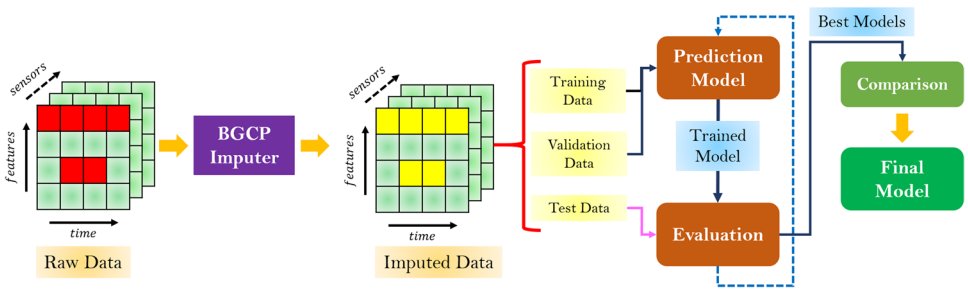
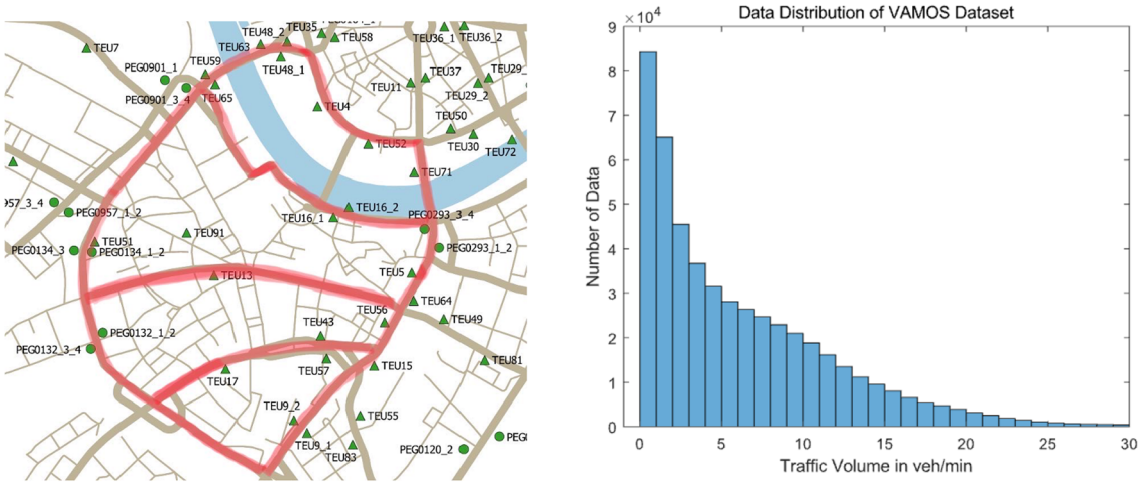


Fig. 5 Workflow diagram for developing the data-driven traffic volume predictor



(a) Arterial view. (b) VAMOS dataset.

Fig. 6 Dataset components: **a** Dresden arterial network (source of the VAMOS traffic data); **b** example of traffic volume data recorded by VAMOS sensors

values, aiding interpretability. However, this can be misleading for near-zero actual values owing to the inflated relative errors.

In this study, the RMSE and MAPE were used to assess the performance of the imputation and prediction tasks. Additionally, MSE was adopted as the loss function during the training of the neural-network-based prediction models. The mathematical expressions of all the metrics are defined as follows:

$$MSE = \frac{1}{n}\Sigma_{i=1}^{n}(y_i - x_i)^2 \tag{20}$$

Fig. 7 Relationship between imputation errors and extra missing (VAMOS)

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}(y_i - x_i)^2} \tag{21}$$

$$MAPE = \frac{100\%}{n}\Sigma_{i=1}^{n}\left|\frac{y_i - x_i}{y_i}\right| \tag{22}$$

where y_i represents the original data points, x_i denotes the imputed or predicted data, and n denotes the total number of data points in the dataset.

Data Preparation

The raw VAMOS dataset contains some missing entries. We can categorize the missing entries in the dataset into two types: random missing and fiber missing. Random missing entries occur unpredictably throughout the dataset and are unrelated to each other. This is typical of data collected by 1-minute interval sensors. Fiber missing entries involve continuous missing data for a specific day or several days, potentially due to sensor malfunctions. The missing rate in the VAMOS raw data varies, with some sensors experiencing

Table 2 Overview of the final dataset

Attributes	Values/Samples
Number of total samples	1,208,480
Number of sensors	28
Number of training set	725,088
Number of validation set	241,696
Number of testing set	241,696

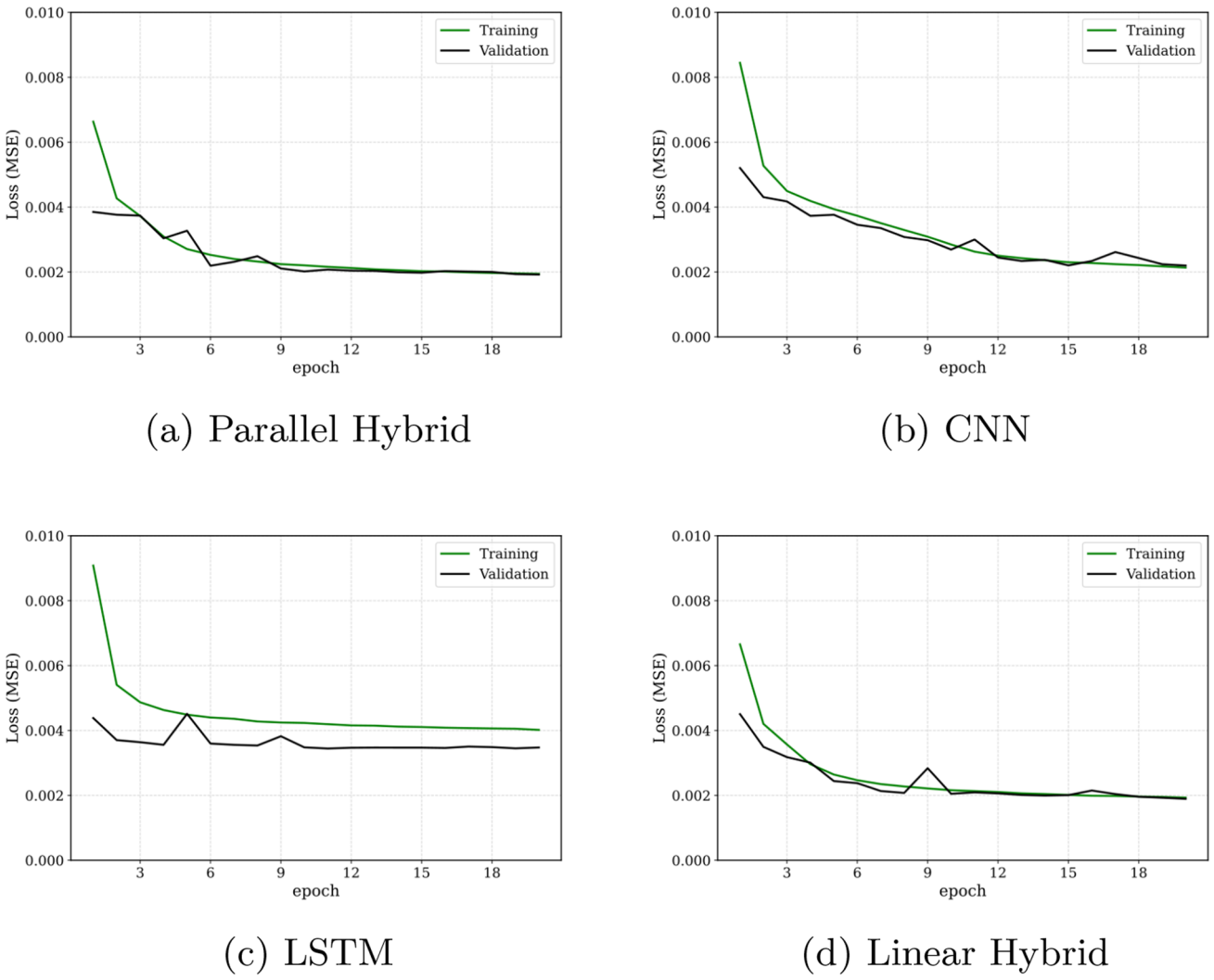


Fig. 8 The variation of the loss function during the training

missing values up to 50%. Although this is not a frequent occurrence, it may still interfere with prediction accuracy.

As the VAMOS data lacks ground truth Yan (2022), we first attempted to reproduce the BGCP model's imputation results by emulating a similar missing pattern in two open-source datasets-Guanzhou and Hangzhou-that are mentioned in the original paper. The descriptions of the two open-source datasets are provided in Appendix A. The detailed results of our imputation experiment are shown in Table 4. Our primary focus was to address the missing values in the raw data; hence, we did not conduct extensive experiments on imputation performance. However, we evaluated the overall performance of imputation by intentionally removing some samples. The results of this evaluation are detailed in Fig. 7. The y-axis of the MAPE is expressed in percentage and only indicates the relative increase compared to the original, complete VAMOS dataset. It is noticeable that as the quantity of additionally deleted data increases, the error of the BGCP imputation based on the VAMOS dataset also rises significantly.

Table 3 RMSE values for different models across various time horizons

Time Horizon (min)	Parallel Hybrid	CNN	LSTM	Linear hybrid
1	0.9577	1.0245	1.1022	1.3392
5	1.0219	1.0756	1.1645	1.3189
10	1.0421	1.1163	1.2161	1.5913
15	1.1555	1.2527	1.2465	1.3848
20	1.1790	1.2698	1.3826	1.4653
40	1.4168	1.3734	1.5572	1.5477
60	1.6975	1.6280	1.7923	1.9191
120	2.3481	2.3125	2.5155	2.9841

We used this imputed data for training, validation, and testing to evaluate our model's performance. We split the data in a 6:2:2 ratio for the training, validation, and testing datasets, respectively. The details of the final processed data are shown in Table 2.

Results

Experimental Results and Analysis

We conducted experiments on our prediction models by using the imputed VAMOS dataset. The experiment involved utilizing historical traffic volume data from all 28 sensors over the past 40 min to forecast traffic volumes at these locations for several future intervals (e.g., 1, 5, and 10 min). The input data for the traffic volume were structured in a tensor with dimensions representing minutes by sensors. During the training process, the loss function is determined by MSE. It is employed to measure the distance between predictions and ground-truth traffic volume. Thus, minimizing the MSE is taken as the training goal of the four NN models, let the set of all the trainable parameters in a neural network model be Θ, and the training strategy can be described as follows:

$$\Theta = \arg\min_{\Theta} MSE. \tag{23}$$

The MSE for both the training and validation sets was computed for all four models after each epoch. Using the training of the 1-minute prediction model as an example, Fig. 8 illustrates a consistent decrease in MSE across the 10 training epochs, indicating effective training. Notably, during the last two or three epochs, the MSE for the validation set stabilized, despite initial fluctuations, which suggests model convergence. However, it was observed that the MSE values for the LSTM and linear hybrid models were significantly higher than those for the other two models. This implies that the training efficacy for these two models might be somewhat constrained. Whether these limitations translate to inferior performance on the test set remains an area for further investigation. Given the data and settings described, seven distinct predictors were developed for each neural network architecture to accommodate various prediction time horizons: 1 min, 5 min, 15 min, 20 min, 40 min, 60 min, and 120 min.

The evaluation of different predictors was conducted using a test set comprising traffic volume data from October 10th to 15th, 2018. We only consider RMSE as the primary goal of the study is to evaluate and compare the prediction accuracy of different models Hodson (2022). For this purpose, RMSE is an appropriate and sufficient metric. It directly measures the deviation of predictions from actual values, aligning with our study objectives. The average RMSE for these predictors was assessed across different prediction time horizons, as depicted in Table 3. Although the loss functions were computed on normalized data (ranging from 0 to 1), the RMSE values reported here were calculated using actual, unnormalized data, providing a more intuitive performance measure in vehicles per minute. As shown in table, LSTM predictors consistently underperformed compared to CNN and parallel hybrid models. This underperformance is likely due to the LSTM's emphasis on temporal correlations and its approach of treating data from various sensors as distinct time series. Consequently, as the

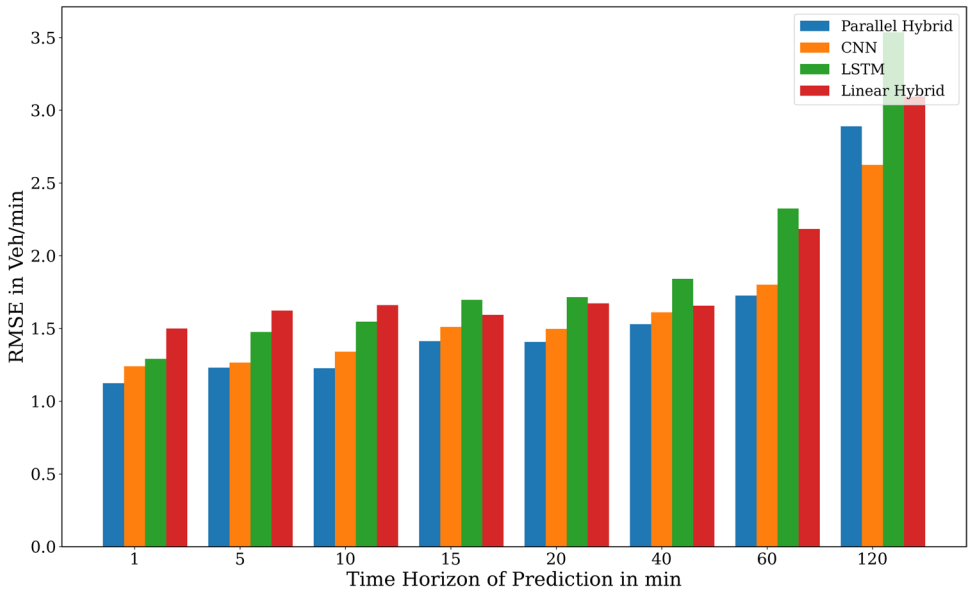


Fig. 9 Relationship between RMSE and prediction time horizons during rush hour

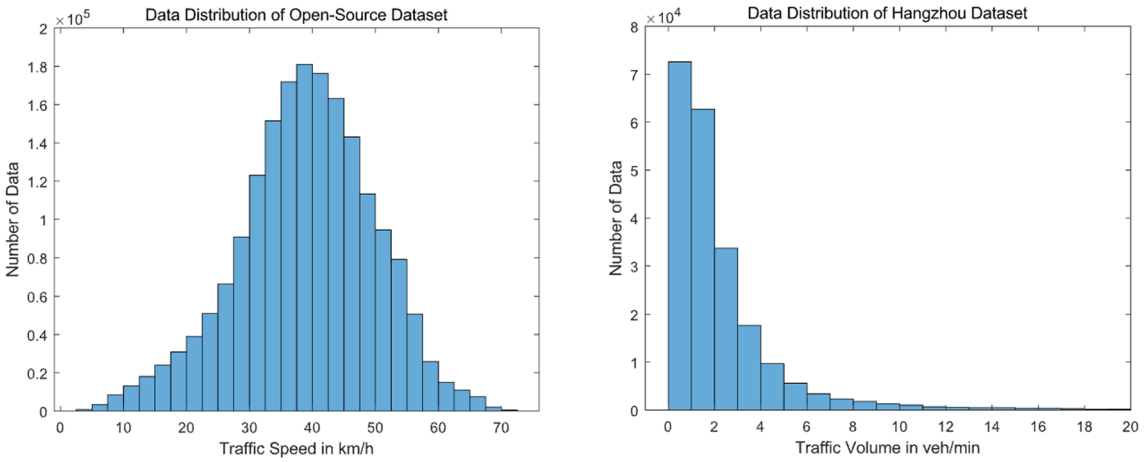
prediction horizon extended, the accuracy of LSTM-based models diminished noticeably.

Table 3 highlights that the parallel hybrid model achieved the lowest RMSE across most time horizons, indicating superior performance in traffic volume prediction. The CNN model followed closely, outperforming the LSTM model in shorter time horizons but showing a less pronounced accuracy degradation over longer prediction periods. Conversely, the linear hybrid model exhibited the highest RMSE values, especially as the prediction horizon increased, reflecting its relatively poorer performance in capturing complex traffic patterns.

We also further investigated the performance of the four models across different time horizons for specific scenarios, such as during rush hours. Specifically, the focus was on the 6:00 to 10:00 timeframe, which corresponds to Dresden's morning rush hour. As illustrated in Fig. 9, the proposed parallel hybrid structure consistently demonstrated superior or comparable prediction accuracy using the RMSE metric. In the overall traffic scenario, the CNN model performed better in some cases. However, in special scenarios such as rush hours, it is evident that the parallel hybrid model outperforms or matches the performance of the other models, reinforcing its robustness and reliability for short-term traffic predictions. This indicates that the hybrid model is more robust and consistently delivers superior performance across various traffic conditions.

Conclusion and Future Works

In this paper, an imputation-enhanced traffic volume prediction framework is proposed. The framework employs a Bayesian Gaussian Copula Process (BGCP) based traffic imputation model to prepare the data, ensuring that the model receives error-free inputs. Additionally, a novel parallel hybrid deep learning approach is designed to enhance the predictability of traffic volume. The efficacy of the proposed method is validated using the VAMOS dataset from Dresden, Germany. Experimental results demonstrate that the framework can accurately predict traffic volume even with



(a) Guangzhou dataset. (b) Hangzhou dataset.

Fig. 10 The data distribution: **a** speed data of Guangzhou dataset, and **b** volume data of Hangzhou dataset

Table 4 Values of RMSE and MAPE under random missing (Guangzhou)

Location	Type	Error	10%	20%	30%	40%	50%	60%	70%
Guangzhou	Random	RMSE	3.3384	3.3530	3.3671	3.3869	3.4258	3.4782	3.5387
		MAPE	0.0774	0.0778	0.0781	0.0784	0.0790	0.0801	0.0813
	Fibre	RMSE	3.6025	3.7986	4.0524	4.5985	4.9651	5.4099	6.0790
		MAPE	0.0793	0.0825	0.0858	0.0919	0.1002	0.1085	0.1190
Hangzhou	Random	RMSE	0.2577	0.3148	0.3978	0.4579	0.5141	0.6085	0.7762
		MAPE	0.1441	0.1521	0.1596	0.1697	0.1812	0.1970	0.2353
	Fibre	RMSE	0.2258	0.3272	0.4266	0.4786	0.6072	0.7042	0.8869
		MAPE	0.1433	0.1513	0.1608	0.1711	0.1808	0.1946	0.2417

sparse raw data. This study emphasizes the integration of spatiotemporal information for multi-step prediction.

Testing different imputation models is not the focus of the paper; instead, we used a state-of-the-art model in this study. We reproduced BGCP imputation results using two open-source datasets and conducted an indirect evaluation of imputation method. However, exploring different imputation models and actively evaluating the imputation results warrants a future research direction. The imputation algorithm in the proposed framework is computationally demanding, and there have been significant developments in deep learning-based imputation models. Future work will focus on improving computational efficiency and utilizing newer models for imputation. Additionally, the influence of road topology within the urban network will be considered in the next stage (e.g., graph convolution network (GCN) Rahman and Hasan (2023)).

Guangzhou and Hangzhou Dataset, China

The first one is a large-scale traffic dataset collected in Guangzhou, China. The dataset is generated by a widely used navigation application on smartphones, which contains travel speed observations from 214 road segments in 61 days, the time period from 1*st* August, 2016 to 30*th* September, 2016. The time interval of Guangzhou dataset is 10 min. The total amount of observations is 1.88M. The data distribution of Guangzhou dataset is shown in Fig. 10a. As for the Hangzhou dataset, it contains the traffic volume information from 80 different metro stations in Hangzhou, China within 25 days. The total amount of Hangzhou dataset is 21,600 samples. The data distribution histogram is shown in Fig. 10b.

Acknowledgements The authors would like to thank the Road Administration Office of the City of Dresden for providing access to VAMOS traffic data for this work.

Author Contributions Yan Peng: Conceptualization, Methodology, Programming, Experimental Design, Visualization, Writing - original draft. Zirui Li: Programming, Experimental Design, Visualization, Writing - original draft. Jyotirmaya Ijaradar: Programming, Testing, Writing - review & editing. Sebastian Pape: Conceptualization, Writing - review & editing, Supervision. Matthias Körner: Conceptualization, Writing - review & editing, Supervision. Meng Wang: Conceptualization, Methodology, Writing - review & editing, Supervision.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data availability The code generated from the study is accessible upon request from the corresponding author. The traffic data used in this study can be requested from the corresponding author and is accessible subject to approval by the Road Administration Office of the City of Dresden.

Declarations

Competing interests The authors declare no competing interests.

References

Asif MT, Mitrovic N, Dauwels J, Jaillet P (2016) Matrix and tensor based methods for missing data estimation in large traffic networks. IEEE Trans Intell Transp Syst 17(7):1816–1825

Azfar T, Li J, Yu H, Cheu RL, Lv Y, Ke R (2024) Deep learning-based computer vision methods for complex traffic environments perception: a review. Data Sci Transport 6(1):1–27

Boquet G, Vicario JL, Morell A, Serrano J (2019) Missing data in traffic estimation: a variational autoencoder imputation method. In: 2019 IEEE International Conference on acoustics, speech and signal processing (ICASSP), pp. 2882–2886. IEEE

Cao M, Li VO, Chan VW (2020) A CNN-LSTM model for traffic speed prediction. In: 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), pp 1–5. IEEE

Castro-Neto M, Jeong Y-S, Jeong M-K, Han LD (2009) Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions. Expert Syst Appl 36(3):6164–6173

Chan RKC, Lim JM-Y, Parthiban R (2023) Missing traffic data imputation for artificial intelligence in intelligent transportation systems: review of methods, limitations, and challenges. IEEE Access 11:34080–34093

Chang G, Zhang Y, Yao D (2012) Missing data imputation for traffic flow based on improved local least squares. Tsinghua Sci Technol 17(3):304–309

Chen X, He Z, Sun L (2019a) A Bayesian tensor decomposition approach for spatiotemporal traffic data imputation. Transport Res Part C Emerg Technol 98:73–84

Chen X, He Z, Chen Y, Lu Y, Wang J (2019b) Missing traffic data imputation and pattern discovery with a Bayesian augmented tensor factorization model. Transport Res Part C Emerg Technol 104:66–77

Chen X, Lei M, Saunier N, Sun L (2021) Low-rank autoregressive tensor completion for spatiotemporal traffic data imputation. IEEE Trans Intell Transp Syst 23(8):12301–12310

Chen Y, Huang J, Xu H, Guo J, Su L (2023) Road traffic flow prediction based on dynamic spatiotemporal graph attention network. Sci Rep 13(1):14729

Cui Z, Ke R, Pu Z, Wang Y (2020) Stacked bidirectional and unidirectional LSTM recurrent neural network for forecasting network-wide traffic state with missing values. Transport Res Part C Emerg Technol 118:102674

Do LN, Taherifar N, Vu HL (2019) Survey of neural network-based models for short-term traffic state prediction. Wiley Interdiscipl Rev Data Min Knowl Discov 9(1):1285

Furno A, Zanella AF, Stanica R, Fiore M (2024) Spatial and temporal exploratory factor analysis of urban mobile data traffic. Data Sci Transport 6(4):1–18

Gershman SJ, Blei DM (2012) A tutorial on Bayesian nonparametric models. J Math Psychol 56(1):1–12

Ghosh B, Basu B, O'Mahony M (2007) Bayesian time-series model for short-term traffic flow forecasting. J Transp Eng 133(3):180–189

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, pp. 922–929

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

Hodson TO (2022) Root-mean-square error (rmse) or mean absolute error (mae): when to use them or not. Geosci Model Dev 15(14):5481–5487

Huang T, Chakraborty P, Sharma A (2023) Deep convolutional generative adversarial networks for traffic data imputation encoding time series as images. Int J Transport Sci Technol 12(1):1–18

Islam Z, Abdel-Aty M, Mahmoud N (2022) Using CNN-LSTM to predict signal phasing and timing aided by high-resolution detector data. Transport Res Part C Emerg Technol 141:103742

Jiang W, Luo J (2022) Graph neural network for traffic forecasting: a survey. Expert Syst Appl 207:117921

LeCun Y, Boser B, Denker J, Henderson D, Howard R, Hubbard W, Jackel L (1989) Handwritten digit recognition with a back-propagation network. Adv Neural Inform Process Syst 2:15

LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324

LeCun Y, Bengio Y et al (1995) Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks 3361(10):1995

Lee S, Fambro DB (1999) Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting. Transp Res Rec 1678(1):179–188

Li Q, Tan H, Wu Y, Ye L, Ding F (2020) Traffic flow prediction with missing data imputed by tensor completion methods. IEEE Access 8:63188–63201

Li G, Li Z, Knoop VL, Lint H (2024) Unravelling uncertainty in trajectory prediction using a non-parametric approach. Transport Res Part C Emerg Technol 163:104659

Liang Y, Zhao Z, Sun L (2022) Memory-augmented dynamic graph convolution networks for traffic data imputation with diverse missing patterns. Transport Res Part C Emerg Technol 143:103826

Liu A, Li C, Yue W, Zhou X (2018) Real-time traffic prediction: a novel imputation optimization algorithm with missing data. In: 2018 IEEE Global Communications conference (GLOBECOM), pp 1–7. IEEE

Liu X, Xia Y, Liang Y, Hu J, Wang Y, Bai L, Huang C, Liu Z, Hooi B, Zimmermann R (2024) Largest: a benchmark dataset for large-scale traffic forecasting. Adv Neural Inform Process Syst 36

Lv Y, Duan Y, Kang W, Li Z, Wang F-Y (2015) Traffic flow prediction with big data: a deep learning approach. IEEE Trans Intell Transp Syst 16(2):865–873

Lyu C, Lu Q-L, Wu X, Antoniou C (2024) Tucker factorization-based tensor completion for robust traffic data imputation. Transport Res Part C Emerg Technol 160:104502

Ma X, Tao Z, Wang Y, Yu H, Wang Y (2015) Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. Transport Res Part C Emerg Technol 54:187–197

Ma X, Dai Z, He Z, Ma J, Wang Y, Wang Y (2017) Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. Sensors 17(4):818

Ma J, Chan J, Ristanoski G, Rajasegarar S, Leckie C (2019) Bus travel time prediction with real-time traffic information. Transport Res Part C Emerg Technol 105:536–549

Medsker LR, Jain L (2001) Recurrent neural networks. Des Appl 5(64–67):2

Qu L, Hu J, Li L, Zhang Y (2009) PPCA-based missing data imputation for traffic flow volume: a systematical approach. IEEE Trans Intell Transp Syst 10(3):512–522

Rahman R, Hasan S (2023) Data-driven traffic assignment: a novel approach for learning traffic flow patterns using graph convolutional neural network. Data Sci Transport 5(11):1–20

Rahmani S, Baghbani A, Bouguila N, Patterson Z (2023) Graph neural networks for intelligent transportation systems: a survey. IEEE Trans Intell Transp Syst 24(8):8846–8885

Rajalakshmi V, Ganesh Vaidyanathan S (2022) Hybrid CNN-LSTM for traffic flow forecasting. In: Proceedings of 2nd International Conference on artificial intelligence: advances and applications: ICAIAA 2021, pp. 407–414

Ramana K, Srivastava G, Kumar MR, Gadekallu TR, Lin JC-W, Alazab M, Iwendi C (2023) A vision transformer approach for traffic congestion prediction in urban areas. IEEE Trans Intell Transp Syst 24(4):3922–3934

Ran B, Tan H, Feng J, Wang W, Cheng Y, Jin P (2015) Estimating missing traffic volume using low multilinear rank tensor completion. J Intell Transport Syst 20(2):152–161

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

Salman D, Direkoglu C, Kusaf M, Fahrioglu M (2024) Hybrid deep learning models for time series forecasting of solar power. Neural Comput Appl 36:1–18

Shah J, Vaidya D, Shah M (2022) A comprehensive review on multiple hybrid deep learning approaches for stock prediction. Intell Syst Appl 16:200111

Sharma S, Lingras P, Zhong M (2004) Effect of missing values estimations on traffic parameters. Transport Plan Technol 27(2):119–144

Smith BL, Scherer WT, Conklin JH (2003) Exploring imputation techniques for missing data in transportation management systems. Transport Res Rec J Transport Res Board 1836(1):132–142

Tian T, Zhang L, Shen J, Jiang Y, Zhou L, Chang R, Zhao S, Xu D (2023) Missing data imputation for traffic flow data using saegan-sad. In: China National Conference on big data and social computing, pp. 375–388. Springer

Vlahogianni EI, Karlaftis MG, Golias JC (2014) Short-term traffic forecasting: where we are and where we're going. Transport Res Part C Emerg Technol 43:3–19

Wang Y, Zheng J, Du Y, Huang C, Li P (2022) Traffic-ggnn: predicting traffic flow via attentional spatial-temporal gated graph neural networks. IEEE Trans Intell Transp Syst 23(10):18423–18432

Wang A, Ye Y, Song X, Zhang S, James J (2023) Traffic prediction with missing data: a multi-task learning approach. IEEE Trans Intell Transp Syst 24(4):4189–4202

Wang J, Gu Q, Wu J, Liu G, Xiong Z (2016) Traffic speed prediction and congestion source exploration: a deep learning method. In: 2016 IEEE 16th International Conference on data mining (ICDM), pp. 499–508. IEEE

Wu Y, Tan H (2016) Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework. arXiv preprint arXiv:1612.01022

Xue J, Ka E, Feng Y, Ukkusuri SV (2024) Network macroscopic fundamental diagram-informed graph learning for traffic state imputation. Transport Res Part B Methodol, p 102996

Yan P (2022) Traffic volume prediction using neural network approach. Diploma thesis, Technische Universität Dresden

Ye J, Zhao J, Ye K, Xu C (2020) How to build a graph-based deep learning architecture in traffic domain: a survey. IEEE Trans Intell Transp Syst 23(5):3904–3924

Ye Y, Xiao Y, Zhou Y, Li S, Zang Y, Zhang Y (2023) Dynamic multi-graph neural network for traffic flow prediction incorporating traffic accidents. Expert Syst Appl 234:121101

Ye Y, Zhang S, Yu JJ (2021) Spatial-temporal traffic data imputation via graph attention convolutional network. In: International Conference on artificial neural networks, pp 241–252. Springer

Yin X, Wu G, Wei J, Shen Y, Qi H, Yin B (2021) Deep learning on traffic prediction: Methods, analysis, and future directions. IEEE Trans Intell Transp Syst 23(6):4927–4943

Zhang HM (2000) Recursive prediction of traffic conditions with neural network models. J Transp Eng 126(6):472–481

Zhang Z, Lin X, Li M, Wang Y (2021) A customized deep learning approach to integrate network-scale online traffic data imputation and prediction. Transport Res Part C Emerg Technol 132:103372

Zhao Z, Li Z, Li F, Liu Y (2021) CNN-LSTM based traffic prediction using spatial-temporal features. J Phys Conf Ser 2037(1):012065

Zheng W, Lee D-H, Shi Q (2006) Short-term freeway traffic flow prediction: Bayesian combined neural network approach. J Transp Eng 132(2):114–121

Zhong M, Lingras P, Sharma S (2004) Estimation of missing traffic counts using factor, genetic, neural, and regression techniques. Transport Res Part C Emerg Technol 12(2):139–166

Zhong M, Sharma S, Lingras P (2004) Genetically designed models for accurate imputation of missing traffic counts. Transp Res Rec 1879(1):71–79