



# A Traffic Flow Dependency and Dynamics based Deep Learning Aided Approach for Network-Wide Traffic Speed Propagation Prediction

Hanyi Yang<sup>a</sup>, Lili Du<sup>b,\*</sup>, Guohui Zhang<sup>a</sup>, Tianwei Ma<sup>a</sup>

<sup>a</sup> University of Hawaii at Manoa

<sup>b</sup> University of Florida

## ARTICLE INFO

### Keywords:

Traffic Speed Propagation  
Deep Learning  
LSTM  
v-CTM  
Flow Dependency  
Dynamics

## ABSTRACT

The information of network-wide future traffic speed distribution and its propagation is beneficial to develop proactive traffic congestion management strategies. However, predicting network-wide traffic speed propagation is non-trivial. This study develops a traffic flow dependency and dynamics based deep learning aided approach (TD<sup>2</sup>-DL), which predict network-wide high resolution traffic speed propagation by explicitly integrating temporal-spatial flow dependency, traffic flow dynamics with deep learning method techniques. Specifically, we first develop a graph theory-based method to identify the local temporal-spatial traffic dependency of each road among neighboring roads adaptive to the prediction horizon and traffic delay. Then, traffic speed propagation on every road is mathematically described by v-CTM based on traffic initial and boundary conditions. Next, the long short-term memory (LSTM) model is employed to predict boundary conditions factoring the traffic temporal-spatial dependency and historical data predicted by v-CTM. In this way, we well couple the physical models (traffic dependency and v-CTM) with the deep learning approach, and further make them coevolution under this framework. Last, an EKF is used to assimilate predicted traffic speed predicted by v-CTM coupled with the LSTMs and the field traffic data; an FNN is introduced to impute missing and corrupted data for improving the traffic speed prediction accuracy. The numerical experiments indicated that the TD<sup>2</sup>-DL predicted the network-wide traffic speed propagation in 30 minutes with accuracy varying from 85%-98%. It outperformed the tested models recently developed in literature. The ablation experimental results confirmed the significance of factoring traffic dependency and integrating data imputation and assimilation techniques for improving the prediction accuracy.

## 1. Introduction

Traffic state prediction is a vital component of Intelligent Transportation Systems (ITS). Among the three traffic characteristics, traffic speed is an important measurement to illustrate traffic conditions, and it can facilitate traffic control and management as well as individual vehicle trip decisions. By predicting traffic speed propagation, the transportation administration department can develop proactive traffic management and control strategies to improve traffic efficiency. Consequently, traffic speed prediction and monitoring have obtained a significant interest in literature, using various data such as probe vehicles data, fixed point data, and so on (Ma

\* Corresponding author: Dr. Lili Du. University of Florida, United States

E-mail address: [lilidu@ufl.edu](mailto:lilidu@ufl.edu) (L. Du).

et al., 2015; Qi and Ishak, 2014; Tang et al., 2017; Williams and Hoel, 2003; Wang et al., 2019; Xu et al., 2017; Zhang and Xie, 2007; Zhang and Liu, 2009). Although statistical approaches have good theoretical interpretability and direct interferences (Tang et al., 2017), it shows shortcomings when dealing with complex traffic prediction problems with high dimensional data (Bogaerts et al., 2020). Consequently, many studies employed machine learning approaches, such as Hidden Markov Model, Fuzzy Neural Network, Support Vector Machines, etc., for traffic speed prediction (Ma et al., 2015; Qi and Ishak, 2014; Zhang and Xie, 2007; Zhang and Liu, 2009) since they demonstrate a strong ability to address high-dimensional data and extract nonlinear relationships (Bogaerts et al., 2020). However, until recent years, the majority of existing studies have focused on predicting traffic speed on a single road at a single point using data-driven approaches (Tang et al., 2017; Williams and Hoel, 2003; Xu et al., 2017; Wang et al., 2019). Nevertheless, recent technological advances in communication, information, computing, and vehicular technologies enable smarter and emerging traffic management and control at both system and individual vehicle levels. They require support from more comprehensive network-wide temporal and spatial traffic speed information in real time. For example, connected and autonomous vehicles (CAVs) seek to conduct adaptive motion planning according to network-wide traffic speed evolution dynamics. With the help of real-time temporal-spatial speed distribution information down a path, electronic vehicles (EV) can practice real-time energy usage management. Moreover, traffic managers can develop dynamic speed harmonization strategies (Hegyi et al., 2005) by knowing the real-time spatial traffic speed distribution in a corridor involving multiple nonhomogeneous road segments.

However, predicting such high-resolution network-wide traffic speed propagation in real time is not trivial. One of the critical challenges is the flow interdependency among roads. Namely, each road is not isolated in a network, and traffic propagation on a road will affect others with a flow propagation delay. Those machine learning approaches developed for a single road cannot capture this temporal-spatial dependency in a road network. To address this issue, we need more advanced techniques. Encouraged by the success of deep learning methods in the field of image identification (Karpathy et al., 2015; Krizhevsky et al., 2012) and language modeling (Sundermeyer et al., 2012), recent studies use them to analyze the temporal-spatial relations on the road network and conduct network-wide traffic prediction, such as (Zhang et al., 2018; Zhang et al., 2019). However, we noticed that these data-driven approaches mainly focus on predicting point traffic speed (Zhang et al., 2019) or the average traffic speed (Zhang et al., 2019) on a road, while ignoring the spatial variation of traffic speed along the road due to the lack of sufficient data (e.g., it is not practical to densely deploy fixed sensors to capture spatial speed distribution on a road, while probe vehicles only represent partial flow). Apparently, this low spatial resolution is not sufficient to speed information on a long-stretch road. This weakness limits their applications for emerging transportation management and control we mentioned above. On the other hand, this study recognized that many theoretical traffic flow models had been developed (Asif et al., 2013; Bogaerts et al., 2020; Daganzo, 1997; Work et al., 2008; Daganzo, 1992; Daganzo, 1995) to describe the traffic propagation dynamics over a closed road segment. Still, their performance for real-time traffic prediction is not well satisfied since theoretical models cannot timely respond to traffic variation in reality. Overall, the state-of-the-art shows that few studies can predict network-wide traffic speed propagation dynamics with a high spatiotemporal resolution. It is an important research gap and calls for novel approaches.

Motivated by the above research gap, this study seeks to predict a high-resolution temporal-spatial traffic speed distribution and propagation (see the visualization of the result in Fig. 7) over a road network, using real-time traffic data collected by probe vehicles and/or fixed sensors. The novelty of the research raises several research challenges regarding how to capture flow dependency, temporal-spatial speed distribution and their propagation dynamics. To address those challenges, we want to construct a traffic flow dependency and dynamics based deep-learning aided framework (TD<sup>2</sup>-DL) that maps a historical network speed profile within a time interval  $\bar{h}$ , to a network speed profile in a future prediction horizon  $\bar{h}$ , taking real-time collected traffic speed data as inputs. Specifically, the physical model ( $v$ -CTM, a discrete partial differential equation) describes traffic dynamics theoretically, and the deep learning model (LSTM combined with data imputation approaches) is used to predict the boundary conditions, which the physical model needs to implement in real time. Fig. 1 illustrates the structure of the framework. We explain the ideas and methodologies as

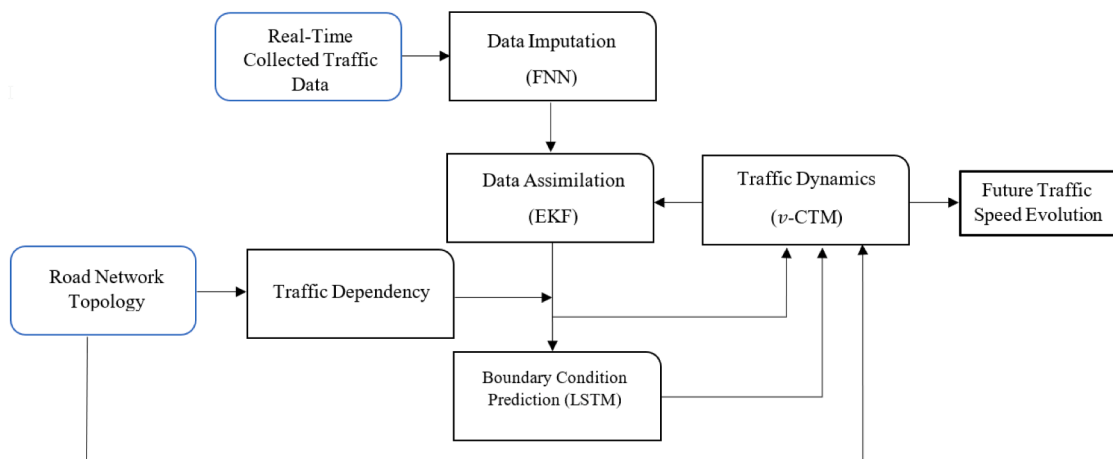


Fig. 1. The Architecture of Traffic Speed Propagation Prediction Model.

follows. First, we develop a dynamic programming algorithm (labeled as [Algorithm 1](#)) to capture the traffic temporal-spatial dependence among neighboring roads based on the graph theory method. Different to existing data-driven approaches, such as Graph Recurrent Neural Network (GRNN), which capture global flow dependency over a network, [Algorithm 1](#) enables us to adaptively identify each road's furthest upstream and downstream flow-dependent neighbors within a given prediction horizon according to network topology and real-time traffic propagation speed. It reduces the data request and computation load. Next, to capture the spatial distribution of the traffic speed on a road, we discretize each road in a network into many connected homogeneous segments and then each road segment (also referred to as road hereafter in this paper) is further split into uniform cells. Then, a road's traffic speed profile is described by a speed vector with each element representing the traffic speed in a cell. Traffic speed propagation among cells on a road is mathematically described by  $v$ -CTM, requiring traffic initial and predicted boundary conditions at inputs. Given traffic flow present time series pattern, the predicted traffic speed profiles on the roads in one time step, say  $k$ , will further be taken as inputs to predict later boundary conditions,  $k + 1, k + 2, \dots$  so that we can capture traffic continuity.

Then, we noticed one more key technical challenge is to accurately estimate the initial and boundary traffic conditions of each road at each sample time to feed the  $v$ -CTM, factoring in traffic flow dependency and continuity. Accordingly, we develop a *LSTM network* built upon the physical network topology, in which each node (i.e., a LSTM for a road) in charge of updating the boundary conditions for a road at each sample time step. To be noted, each LSTM uses dual-speed data sources from both this road and its local neighbors. Moreover, each data source involves the historical traffic speed predicted by  $v$ -CTM and the field-collected data. Thus, the LSTMs and  $v$ -CTM are coupled. They sequentially provide input data to each other (see the loop between the LSTM and the State Transition Model in [Fig. 1](#)) and co-evolve under this  $TD^2$ -DL framework to move forward the speed prediction on each road over the network. More importantly, the LSTMs on all roads in the network simultaneously function, and they are connected through a local traffic dependency identified by [Algorithm 1](#). In short, the  $TD^2$ -DL couples the traffic flow physical models ( $v$ -CTM and flow dependency) with a deep learning approach.

Last, we noticed that purely using trajectory data from probe vehicles or fixed sensor data to predict initial/boundary conditions will introduce prediction bias since they either only represent partial flow or point traffic state in a cell and often involve errors due to device malfunctions. On the other hand, only using the theoretical prediction of  $v$ -CTM in previous step as the initial condition for the next time step cannot capture real-time traffic conditions affected by various uncertainties. To address this issue, this study uses Extended Kalman Filter to assimilate the prediction traffic speed from the  $v$ -CTM, integrating the field observations to get a better prediction. Additionally, considering field data may involve missing observations and corrupted data, we apply Feedforward Neural Network (FNN) to address this data imputation issue.

The performance of the  $TD^2$ -DL was validated by the numerical experiments using the simulated trajectory data generated by field-collected loop detector data around exit 164B on I-5, WA18, and WA 167, Seattle, WA. The experiment results confirmed that the  $TD^2$ -DL could predict future traffic speed propagation within 30 minutes with 2% to 15% relative accuracy. The  $TD^2$ -DL performed better than the traditional baseline models and classical machine learning methods. More importantly, the out-of-train-distribution (OOD) test illustrated that the  $TD^2$ -DL performed well for the scenarios out of the training. The ablation tests demonstrated that the key components, including traffic dependency, data assimilation, and data imputation in the  $TD^2$ -DL did contribute to the improvement of traffic speed accuracy.

Overall, the main contributions of this study are summarized as follows. (i) This study conducted network-wide high-resolution traffic speed prediction considering temporal and spatial speed distribution and propagation dynamics on each road and flow interdependencies among roads over a network. This is in contrast and extends existing studies in the literature, which mainly focus on speed predictions of a single road or at isolated locations (i.e., point speeds) of some roads in a network. [Fig. 7](#) later visualizes the predicted speed profile for one road by the  $TD^2$ -DL in our experiments, through which we can identify shockwaves and their propagation on this road. This comprehensive speed information can support proactive traffic control and management better than point traffic speed prediction does. (ii) We developed a physics-model-based deep learning-aided approach. It is an innovative framework seamlessly integrating traffic flow modes ( $v$ -CTM), flow dependency, and deep learning approaches (LSTM), aiming to take advantage of physical models' rigors and data-driven approaches' quick and adaptive responses to real-time traffic conditions. Although this concept has been discussed widely in the literature, this study conducts a real application for a complex transportation application. (iii) We developed a dynamic programming algorithm to determine the scope of spatial traffic dependency adaptive to the prediction horizon. The whole framework also employed data imputation approaches such as FNN and EKF to fix data error issues. Our experiment results show that the  $TD^2$ -DL has a great potential to be integrated into the latest route navigation tools, such as Google Map to help provide more efficient route guidance. Moreover, it is a tremendous added-on capability to a current real-time traffic information system and promotes smart city development by monitoring traffic events in a transportation network.

The presented research efforts are organized as follows. [Section 2](#) reviews the previous literature to motivate this study and well positions the contribution of this study. [Section 3](#) formally defines this study and presents the preliminaries. Building upon that,

#### Algorithm 1:

Pseudo-Code of Incidence Matrix and Impact Delay Matrix Calculation

---

```

for  $i$  in  $\text{range}(\text{len}(A^0))$ : #  $\text{len}(A^0)$  is the number of links in the road network
  for  $j$  in  $\text{range}(\text{len}(A^0))$ :
    if  $A^{n-1}(\text{row} = i, \text{col} = :) \cap A^{0T}(\text{row} = j, \text{col} = :) \neq \emptyset$ :
       $A^n(i, j) = 1$ 
       $K = A^{n-1}(\text{row} = i, \text{col} = :) \cap A^{0T}(\text{row} = j, \text{col} = :)$ 
       $\mathcal{T}^n(i, j) = \min_{k \in K} \{ \mathcal{T}^{n-1}(\text{row} = i, \text{col} = k) + TP(A^0(\text{row} = k, \text{col} = j)) \}$ 

```

---

Section 4 develops the TD<sup>2</sup>-DL framework for predicting traffic speed dynamic propagation. Its performance is then validated by the numerical experiments developed in Section 5. We summarize the whole study and propose future studies in Section 6.

## 2. Literature review

This study falls into the extensive area of traffic state prediction in a road network. Existing studies show that three main types of approaches have been used to predict traffic speed, including statistical methods, traditional machine learning methods, and deep learning methods. Our review shows that the main body of the existing efforts focuses on point or average traffic state prediction on a link based on different traffic data sources, loop detectors, AVI, or probe vehicles. Little attention is given to predicting the traffic state propagation in a road network in higher spatiotemporal resolution. Moreover, most of the previous studies capture the time-series pattern of traffic state but do not fully factor in the temporal-spatial traffic variation and dependency among the neighbor roads. These weaknesses worsen the prediction accuracy and limit its application in practice. The discussion below demonstrates these research gaps in detail and highlights the research needs and contributions.

The statistical methods often conduct traffic state prediction assuming it follows a certain time-series pattern, i.e., the current traffic state,  $v(t)$ , depends on recent historical data,  $v(t-1)$ ,  $v(t-2)$ , ...,  $v(t-h)$ . Accordingly, the autoregressive integrated moving average (ARIMA) model was employed by most of studies, assuming constant mean, variance, and auto-correlation (Hamed et al., 1995; Ma et al., 2015; Williams and Hoel, 2003). For example, the study in (Hamed et al., 1995) used the Box-Jenkins approach to develop an ARIMA model for predicting the traffic flow on a link, taking traffic flow collected on the target link as input data. Williams and Hoel (Williams and Hoel, 2003) developed a seasonal ARIMA process to predict point traffic flow on a link, factoring in seasonal patterns of traffic variation, but temporal-spatial flow dependency is not considered. It is noticed that those statistical approaches often cannot handle data with high dimensions, and they perform poorly as extended to network-wide traffic state prediction.

Consequently, many machine learning algorithms were applied to traffic state prediction to address high dimensional data, including Support Vector Machine (SVM) regression, Artificial Neural Network (ANN), Hidden Markov Model (HMM) (Ma et al., 2015; Ye et al., 2012; Yao et al., 2017; Qi and Ishak, 2014; Tang et al., 2017; Huang and Ran, 2003). Among these methods, ANN was most often used with many different ANN structures such as Feedforward Neural Network (FNN), Neuro-Fuzzy Neural Network (NFNN), Time-Delay Neural Network (TDNN), Long Short-term Memory (LSTM), etc. (Ma et al., 2015). For example, the study in (Ye et al., 2012) proposed a three-layer FNN to predict traffic speed, taking speed and acceleration data as inputs. An improved fuzzy neural network was proposed in (Tang et al., 2017) to predict point traffic speed using multiple periodic traffic characteristics involving recurring traffic patterns and point traffic speed data from three traffic data collection stations. The study in (Ma et al., 2015) introduced the LSTM model to model predict point traffic speed to illustrate the long temporal dependency in traffic data taking advantage of the automatically optimal time lags determination ability of LSTM. We can recognize that all these studies only factor the temporal trend of traffic state on a road, and they only predict traffic state at some specific points.

Although such temporal pattern is a prominent characteristic of traffic state, the spatial dependency among the traffic states on different links in a network or different points on the same link still cannot be ignored. It presents a significant impact as we perform a network-wide traffic state prediction. Motivated by this view, many deep learning models were invited to predict traffic state while learning the temporal and/or spatial flow dependency (Wang et al., 2019; Zhang et al., 2019; Li et al., 2017; Ma et al., 2022; Zheng et al., 2021; Wang et al., 2019; Zhao et al., 2019; Wang et al., 2018; Xie et al., 2019; Song et al., 2020; Zheng et al., 2020; Guo et al., 2019). Briefly, the study of (Wang et al., 2019) proposed a path-based deep learning model built on bidirectional LSTM (Bi-LSTM). This model takes traffic speed on each road as input and builds their temporal-spatial relationship by Bi-LSTM. The output of the Bi-LSTM is further taken as the input of a fully connected neural network to predict traffic speed in the road network. The experiment indicates good performance in a network-wide traffic speed prediction. A Gated Recurrent Units (GRU) based multitask deep learning model was studied in (Zhang et al., 2019) to predict traffic speed in a network. This model takes traffic speed on each road collected by probe vehicles as input, uses the Granger causality test to determine the interaction between different links to build the spatial relationship, and leverages Bayesian optimization to estimate the parameters in this model. It also demonstrates a good performance in the numerical study. The study of (Wang et al., 2018) developed a graph-based neural network method to predict average traffic speed using the trajectory data collected from probe vehicles on each road over a network. More exactly, the deep learning framework involves a linkage network to capture the road network topographical structure, based on which regressor Graph Recurrent Neural Network (GRNN) is used to capture temporal variation of traffic speed over a network. The study of (Xie et al., 2019) combines the graph neural network and recurrent neural network to predict traffic speed variation with time. The spatiotemporal traffic interaction is learned by a graph-to-graph module according to traffic data. Both (Zheng et al., 2020; Guo et al., 2019) employed the attention mechanism in their road traffic flow prediction to model the temporal dependency intensity of traffic flow in consecutive time steps. The authors of (Song et al., 2020) noticed the complex spatial-temporal correlations and heterogeneities among network traffic flow data and trained a localized temporal-spatial graph based neural network to predict point traffic flow on each road over a network, in which a Spatial-Temporal Synchronous Graph Convolutional Module (STSGCM) is trained based on historical traffic data to capture the localized spatial-temporal correlations.

In the meantime, many emerging studies have considered integrating machine learning approaches with traffic flow theory and traffic information from historical traffic data to overcome their own different disadvantages in interpretability, massive historical data requirement, e.g., (Yuan et al., 2021; Shi et al., 2021; Barreau et al., 2021; Rempe et al., 2021; Shi et al., 2021). Mainly, the study in (Yuan et al., 2021) developed a physics regularized machine learning (PRML) method to estimate traffic state on a road by integrating macroscopic traffic flow models with Gaussian process (GP) on a road without involving flow dependency among neighboring roads. The PRML is more robust to the noisy data and more explainable with the help of physics models. In the study of (Shi et al., 2021), a

physics-informed deep learning (PIDL) framework was developed for high-quality TSE with small amounts of observed data, in which a physics-informed neural network is used for regularization. The study of (Barreau et al., 2021) employs the neural network in reconstructing the traffic based on a limited number of probe vehicles by minimizing the obtained traffic measurement and the estimated one.

Although the latest developed graph-based deep learning models have demonstrated higher prediction accuracy compared to classical methods by integrating the temporal and/or spatial traffic dependency through data-driven (learning) approaches, those efforts fusing traffic domain knowledge with machine learning techniques have improved the performance of machine learning models in various aspects of scalar traffic state estimation or prediction, majority of them focus on scalar traffic state prediction (e.g., the point/average traffic speed or flow). None of them worked on capturing traffic speed spatial distribution and propagation dynamics on a road over a road network. This research gap interests this study particularly. We aim to provide a high-resolution temporal-spatial traffic speed distribution and propagation dynamics map (a matrix) for each road over a local transportation network (see the example given in Fig. 7), which provides more comprehensive speed information such as shockwaves for real-time traffic management and individual vehicle trip plan. To achieve this research objective, we employed a macroscopic traffic flow model and its physical features rather than data-driven approaches to capture flow temporal-spatial traffic flow dynamics and dependency. This new research focus and learning framework differentiates our study from most of the existing efforts and also highlights our unique methodology contribution. The following sections will formally define our problem and then introduce our methodology.

### 3. Problem formulation

This research seeks to predict the network-wide traffic speed propagation at each discrete time step  $k \in \mathbb{Z}_+$  with a uniform interval  $\Delta t$ , using imperfect/incomplete traffic speed data, vehicles' trajectory data collected by probe vehicles, such as connected vehicles (CVs), and/or fixed points data collected by fixed point sensors, like loop detectors, cameras, and so on, at the ends of some roads. To do that, we consider a freeway network, as shown in Fig. 2 (a), with the traffic stream moving in it in one direction. Specifically, the road network is split into a series of homogeneous closed road segments ("road" represents a homogeneous closed segment in this paper), denoted by the set  $S = \{s\}_{s=1}^S$ , and the intersection areas like merge, diverge, and weave areas are defined as nodes<sup>1</sup>. For a road,  $s$ , with a length of  $L_s$ , we further define its upstream road set by  $s^- = \{s_m^-, s_{m-1}^-, \dots, s_1^-\}$ , and its downstream road set by  $s^+ = \{s_1^+, s_2^+, \dots, s_n^+\}$ . Here,  $s_m^-$  and  $s_n^+$  are the farthest roads that affect the speed of the road,  $s$ , in a study time range. At each discrete time stamp  $k \in \mathbb{Z}_+$ , we predict the traffic speed propagation on each road in a network in a prediction horizon  $T$  involving  $\bar{h}$  time steps. Accordingly, we have  $\bar{h}\Delta t$ . Furthermore, we split each road,  $s$ , into  $N_s$  spatial cells with the uniform length of  $\Delta l_s$  (see an example in Fig. 2 (b)). Then, we denote the traffic speed profile of a road,  $s$ , at the time step  $k$  by the vector  $\mathbf{v}_s(k) = [v_1(k), v_2(k), \dots, v_{N_s}(k)]^T$ . Putting together the traffic speed profiles of road  $s$  in a prediction horizon  $T$  generates a matrix, which describes a high-resolution temporal-spatial traffic speed distribution and propagation map for the road. Accordingly, the traffic speed profile of a network at time step  $k$  is denoted as  $\mathbf{v}(k) = \{\mathbf{v}_s(k)\}_{s=1}^S$  at a discrete time step  $k$ . In addition, considering the traffic speed propagation on a road (e.g.,  $s$ ) is affected by its downstream (e.g.,  $s^+$ ) and upstream traffic (e.g.,  $s^-$ ), we next define a neighboring traffic speed profile for the road,  $s$ , at a time step  $k$  by  $\mathbf{V}_s(k) = [\mathbf{v}_{s^-}(k'), \mathbf{v}_s(k), \mathbf{v}_{s^+}(k'')] = [v_{s,1}(k^1), v_{s,2}(k^2), \dots, v_{s,m+n+1}(k^{m+n+1})]$ ,  $s \in S$ , which involves the speed profiles of an individual road as well as its upstream neighbors  $\mathbf{v}_{s^-}(k)$ , and downstream neighbors,  $\mathbf{v}_{s^+}(k)$ . Accordingly, we introduce  $\mathbf{V}(k) = \{\mathbf{V}_s(k)\}_{s=1}^S$  to represent the complete speed profile at a time step  $k$  over a network.

This study seeks to develop a deep learning approach,  $F(\cdot)$ , which predicts the high-resolution network-wide traffic speed profile  $\mathbf{V}(k)$  within a prediction horizon  $\bar{h}$ , i.e.,  $\{\mathbf{v}(k + \kappa)\}_{\kappa=1}^{\bar{h}}$ , using the inputs including historical complete traffic speed profile within a historical time interval  $\underline{h}$ , i.e.,  $\{\mathbf{V}(k - \kappa)\}_{\kappa=1}^{\underline{h}}$  and newly collected traffic data  $\mathbf{X}(k)$  at the current step  $k$ . Mathematically, this complicated speed map over a network can be presented by Eq. (1).

$$\{\mathbf{V}(k + \kappa)\}_{\kappa=1}^{\bar{h}} = F\left(\{\mathbf{V}(k - \kappa)\}_{\kappa=1}^{\underline{h}}, \mathbf{X}(k)\right) \quad (1)$$

This research goal raises the following challenges. Briefly, the complete speed profile  $\mathbf{V}(k)$  involves the complicated temporal-spatial dependency among traffic flows on different roads, which is hard to be captured by traditional mathematical approaches (Daganzo, 1992). Next, the traffic speed profile  $\mathbf{v}(k)$  on a road demonstrates strong temporal-spatial dynamics, which is another important characteristic that should be considered in our approach development. Last, field data often involve various errors and data missing issues, which significantly affect the effectiveness of the prediction approaches. For example, probe vehicles only represent partial flow, and their traffic speed cannot accurately capture traffic speed in a mixed flow. Due to unknown software or hardware technical issues, field traffic data may involve errors. We will develop proper data imputation approaches to address these difficulties. Overall, this study will develop a traffic dependency and dynamics-based deep learning (TD<sup>2</sup>-DL) approach, which seamlessly integrates deep learning, data sciences, and traffic flow knowledge together. The next section introduces the development of our approach in technical detail.

<sup>1</sup> A ramp connecting two freeway links is considered as a road if its length is more than seven times of average vehicle length (150 ft) (Daganzo, 2006); otherwise, it is counted as a node.

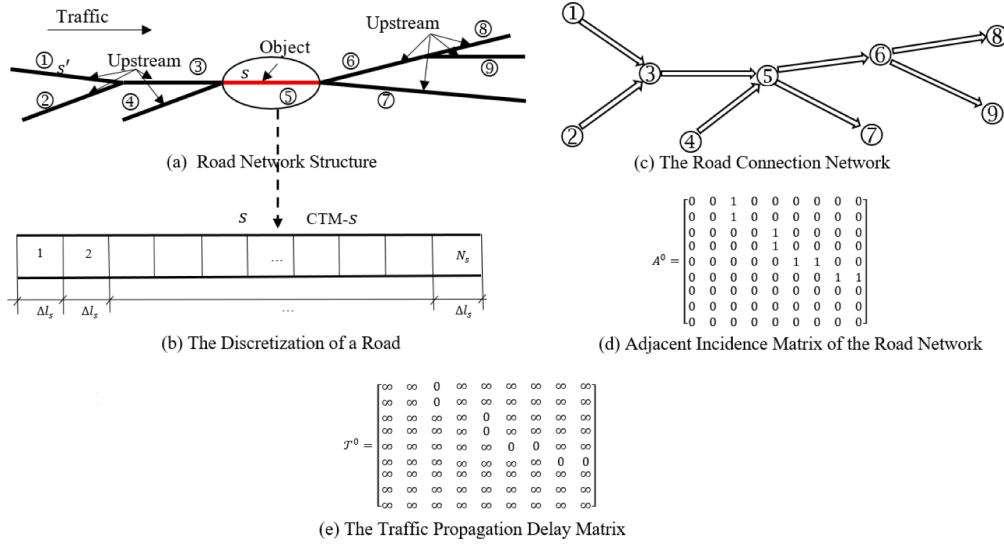


Fig. 2. Freeway Network.

#### 4. Methodology

The TD<sup>2</sup>-DL approach for predicting the high-resolution traffic speed propagation at discrete time step follows the structure shown in Fig. 1. Technically, it integrates the following methods. First, we develop a graph-based approach to capture the temporal-spatial dependency of traffic flow according to its topological structure and traffic flow characteristics. Next, we use a state transmission (v-CTM) model to capture the dynamic process of traffic speed propagation on every individual road according to its boundary conditions and initial condition. Note that the boundary conditions of each road present temporal-spatial dependency. We develop a deep learning framework built upon the flow dependency graph to capture this complicated feature. To be noted, the LSTM is well integrated with the traffic flow physical models, and they co-evolve along the traffic speed propagation prediction process. Besides, the initial condition is another vital factor affecting traffic speed propagation prediction on each road. We will employ an existing mathematical model combined with data imputation and assimilation to generate accurate initial traffic condition estimation. Implementing this deep learning framework for each road, we can capture the network-wide traffic speed dynamic propagation. We introduce each of these methods in the following subsections.

##### 4.1. Temporal-Spatial Flow Dependency Analysis

This section develops a graph theory-based approach to capture the temporal-spatial traffic flow dependency among roads in a network with the help of dynamic programming. Using the example in Fig. 2 (a), we explain our approach as follows. Clearly, the traffic on a road  $s'$  will affect the traffic on road  $s$ , but with a time delay, which depends on the network topology and traffic propagation speed (Daganzo, 1997). Existing studies have recognized the significance of this flow interdependency on traffic state prediction. Accordingly, various graph-based learning approaches have been employed to conduct traffic state prediction, such as GRNN (Wang et al., 2018). However, we noticed that those data-driven approaches usually recognize global spatiotemporal dependency without factoring in the length of the prediction horizon. Namely, some recognized flow dependencies may not be activated during a short prediction horizon due to traffic propagation delay. By recognizing this weakness, this study seeks to develop a new approach, which help us to identify each road's furthest upstream and downstream flow-dependent neighbors within a given prediction horizon according to network topology and traffic speed. It requires less traffic data and computation load than existing data-driven approaches do. To do that, we transfer the physical road network to a connection network (CN), which is a directed graph with each vertex representing a road and each arc representing a connection between two roads. Then, the road network of Fig. 2 (a) is represented by a directed graph in Fig. 2 (c). Furthermore, the topology of the connection network is represented by an adjacent incidence matrix  $A^0$ , in which the row and column index corresponds to the index of the vertices in the connection network; element  $A^0(i,j) = 1$  denotes that vertex  $i$  and vertex  $j$  are connected; otherwise  $A^0(i,j) = 0$ . We can see that the matrix  $A^0$  (e.g., Fig. 2 (d)) demonstrates the first-order connection among vertices in the connection network (e.g., Fig. 2 (c)) (i.e., only records immediate adjacent roads for each road in the physical network). Based on  $A^0$ , we can further discover the indirect connections between vertices by the higher-order adjacent incidence matrices,  $A^1, A^2, \dots, A^n$  with  $A^n = (A^0)^n + 1$ , in which the superscribe illustrates the number of intermediates vertices connecting any pair of vertices, such as  $A^2(2,6) = 1, A^2(2,8) = 0$  as for Fig. 2 (a).

We next define the cost in the connection network, which represents the traffic propagation time delay between two vertices (i.e., roads). Corresponding to the adjacent incidence matrix  $A^n$ , we introduce matrix  $\mathcal{T}^n$  to denote this interest, Fig. 2 (e). Specifically, each element  $\mathcal{T}^n(i,j)$ ,  $i, j \in S$  represents the traffic propagation delay along an arc ( $n = 0$ ) / a path ( $n \neq 0$ ) from vertex  $i$  to vertex  $j$  in the

connection network  $A^n$ , while  $\mathcal{T}^n(i)$  or  $\mathcal{T}^n(j)$  represents the delay for going through individual vertex (a physical road/path). Consequently, we set  $\mathcal{T}^0(i, j) = 0$  if  $A^0(i, j) = 1$ ; otherwise  $\mathcal{T}^0(i, j) = \infty$ , but that of passing through every vertex is defined by  $\mathcal{T}^0(i) = \frac{L_i}{v_{fi}}$ ,  $i \in S$ , where  $v_{fi}$  is the free-flow speed and  $L_i$  is the length of the road corresponding to vertex  $i$ . It is noticed that the traffic propagation from vertex  $i$  to  $j$  depends on the traffic state on the roads in general. But this real-time traffic characteristic is untraceable. This study thus uses the minimum traffic propagation delay under free flow traffic speed to capture such propagation delay, which later will provide the largest neighborhood local dependency network for our deep learning model to predict the speed profile of each individual road segment. Moreover, given  $\mathcal{T}^0(i)$ ,  $i \in S$ ,  $\mathcal{T}^n(i, j)$ ,  $i, j \in S$  is captured by the delay of the shortest path between vertex  $i$  and  $j$ . Consequently, we can capture the flow temporal-spatial dependency of all the vertices by a series of  $A = \{A^n\}$  and  $\mathcal{T} = \{\mathcal{T}^n\}$ .

At the same time, this study noticed that  $A$  is a large sparse matrix. As we implement our approach over a large network, it will cause computation difficulty. Thus, it is necessary to have a method to store and conduct the matrix operation more efficiently. At the same time, we need a method to calculate the corresponding  $\mathcal{T} = \{\mathcal{T}^n\}$  efficiently. Motivated by this view, we develop the following algorithm to calculate the matrix in  $A$  and  $\mathcal{T}$ . Specifically, we will transfer the adjacent incidence matrix to a dense matrix represented by a list of tuples of (row, col, TP), where row and col represent the index of each vertex and TP indicates the delay between them. The multiplication method of this dense matrix is given in (Blueloch, 1996), by which we can find any  $A^n$  from  $A^0$ . Accordingly, the TP can be calculated by the cost of the shortest path connecting the corresponding pair of vertices, e.g., vertex  $i$  and  $j$ . The pseudo-code to calculate and represent  $A^n$  and  $\mathcal{T}^n$  is shown in Algorithm I by integrating the sparse matrix operation method in (Blueloch, 1996) and dynamic programming (Bellman, 1966).

#### 4.2. State Transmission Model (v-CTM)

This section presents our idea to capture traffic speed propagation over a network. Specifically, we employ the Godunov scheme (Zhang, 1993) to describe the traffic speed propagation (i.e., traffic speed dynamics) on each closed road segment. It is an effective numerical discretization method to solve the LWR model (demonstrated it as v-CTM) (Daganzo, 1997; Work et al., 2008) with the given initial condition and weak boundary conditions (Strub I and M Bayen, 2006) on a closed road segment. Below briefly introduces Godunov scheme with the main focus on how we implement it into our study. First, the Godunov scheme requires discretizing the temporal-spatial space into small cells with the size of  $\Delta t \times \Delta l$ . This is consistent to the way we describe road speed profile, but we need to carefully choose the spatial ( $\Delta l$ ) and temporal ( $\Delta t$ ) step sizes so that it follows the Courant-Friedrichs-Lewy (CFL) condition (Eq. (6)) and then ensures the stability of the numerical discretization (Work et al., 2008) for the Godunov scheme. Next, given the initial speed profile  $v_s(k)$  and boundary conditions  $v_s^B(k)$  for road  $s \in S$  at the sample time interval  $k \in \mathbb{Z}_+$ , the traffic speed in each cell  $i \in N_s$  of road  $s \in S$  at the time stamp  $k + 1$  (i.e.,  $\hat{v}_s(k + 1)$ ) can be estimated by in Eqs. (2)–(4) below.

$$\hat{v}_{\mathcal{J}}(k + 1) = v_{\mathcal{J}}(k) - \frac{\Delta t}{\Delta x} (g(v_{\mathcal{J}}(k), v_{\mathcal{J}+1}(k)) - g(v_{\mathcal{J}-1}(k), v_{\mathcal{J}}(k))), \mathcal{J} \in N_s, s \in S \quad (2)$$

$$g(v_i(k), v_{i+1}(k)) = \begin{cases} E(v_{i+1}(k)) & v_i(k) \leq v_{i+1}(k) \leq v^c \\ E(v^c) & v_i(k) \leq v^c \leq v_{i+1}(k) \\ E(v_i(k)) & v^c \leq v_i(k) \leq v_{i+1}(k) \\ \max(E(v_i(k)), E(v_{i+1}(k))) & v_i(k) \geq v_{i+1}(k) \end{cases}, i \in N_s, s \in S \quad (3)$$

$$v_s(0) = v_s^0, \text{ and } v_s^B(0) = \bar{v}_s^B, s \in S, v_s^0 \in [0, v_f], v_s^B \in [0, v_f] \quad (4)$$

$$E(v) = v^2 + vv_f \quad (5)$$

$$\left| \frac{\Delta t}{\Delta l} \max(E'(v)) \right| \leq 1 \quad (6)$$

Where,  $k$  represents the index of time step;  $v$  represents traffic speed;  $v_f$  is the free flow speed;  $v^c$  refers to the critical speed corresponding to the maximal flow;  $v_i(k)$  denotes the traffic speed on the cell  $i$  at time  $k$ ;  $\hat{v}_i(k + 1)$  refers to the predicted traffic speed on the cell  $i$  at time  $k + 1$ ; and  $\bar{v}_s^B$  are the boundary conditions at  $k = 0$ ; at each time step  $k$ , we consider  $v(k)$  as the initial condition for predicting  $v(k + 1)$ ; in particular  $v(0) = \{v_s^0\}_{s=1}^S$  is the initial traffic condition at  $k = 0$ , which can be observed by field traffic data.

To facilitate our discussion in the future, we further introduce the notation CTM-s to represent the model of the v-CTM on a road  $s$ . Theoretically, by implementing the v-CTM on individual roads, we are able to predict the speed distribution and propagation over a network. However, we meet two difficulties to make it work ineffectively in reality. First of all, Eqs. (2)–(4) demonstrate that the speed prediction relies on the estimation of the boundary conditions as each time step. This estimation is affected by the complicated temporal-spatial flow dependency. For example, the traffic flow entering road  $s$  at time step  $k + 1$  with speed  $v_s^B(k + 1)$  may depends on the traffic flow on a upstream road  $s'$ ,  $v_{s'}(k'), k' < k$ , as shown in Fig. 2 (b). Clearly, simply putting together the v-CTM models on individual roads without considering the flow dependency cannot work effectively in predicting speed propagation over a road network. It calls for a new approach. In addition, we noticed that the v-CTM in Eqs. (2)–(4) needs initial conditions as inputs, which are provided by the field data at the beginning of the prediction. Accordingly, data errors become a critical issue to affect the effectiveness of the proposed approach. We address these challenges the following sections.

### 4.3. Traffic Flow Dependency and Dynamics based deep learning framework

This section develops the deep learning framework that couples an LSTM network and the  $v$ -CTM network built upon the road connection network to predict traffic temporal-spatial speed distribution and propagation dynamics over a road network. Corresponding to the example of the road connection network shown in Fig. 2 (c), the structure of the deep learning framework is illustrated by Fig. 3. It includes three layers. The first layer (the bottom one) illustrates the network's topological structure (connection relationship among the roads) according to a given traffic flow direction. Each node represents a road and then the link demonstrates the connection between roads. The second layer (the middle one) represents the  $v$ -CTM network. Each node ( $v$ -CTM- $s$ ,  $s \in S$ ) represents a  $v$ -CTM model capturing the traffic speed propagation dynamics along the corresponding road. The two  $v$ -CTM cells at the two ends of the road are considered as the boundary cells (see the green cells in Fig. 3). Their speed conditions are considered as the boundary conditions of the  $v$ -CTM. The third layer (the top one) demonstrates the LSTM network. Each LSTM is built upon each road (LSTM- $s$ ,  $s \in S$ ). The LSTM layer and the  $v$ -CTM layer are coupled to predict speed distribution and propagation on a road and then over a network. Specifically, LSTM- $s$  is related to the CTM- $s$  on road  $s$ . LSTM- $s$  is used to predict the boundary conditions for the  $v$ -CTM on the corresponding road  $s$ . The inputs of LSTM- $s$  is the traffic speed profiles generated by the CTM- $s$  and the  $v$ -CTM models on the neighboring roads (see the inputs of LSTM 6 in Fig. 3). The scope of the neighboring roads is mathematically determined by Eqs. (8)–(10) provided later and implemented by our dynamic programming Algorithm I. The output of each LSTM- $s$  is the boundary conditions of the road  $s$  (see the dash read arrow line in Fig. 3). They are used as the inputs of CTM- $s$  to perform the speed propagation prediction among the sequential cells on the road (see Fig. 3). Mathematically, the outputs of each LSTM- $s$  is the inputs of CTM- $s$  in Eqs. (2–4). Therefore, each LSTM- $s$  and  $v$ -CTM- $s$  pair forms a domain knowledge-based AI unit on road  $s$  to predict traffic speed temporal and spatial speed distribution and its propagation on the road. The LSTM network relates the traffic speed distribution and propagation predictions via the  $v$ -CTM network to roads in the network.

In short, with the predicted the boundary conditions from each LSTM- $s$  and the initial condition addressed by subsection 4.4 introduced later, the traffic speed profile on road  $s$  at next time step can be predicted by the  $v$ -CTM  $s$ . The predicted speed profile at the current time step will be used as the inputs of LSTM- $s$  and other LSTMs to predict the boundary conditions in road  $s$  and other roads in next time step. This procedure is repeated at every time step in the prediction horizon to provide the traffic speed propagation map on the road network. Thus, the LSTM network together with the traffic flow dynamic models ( $v$ -CTM network) will predict the temporal-spatial traffic speed distribution and propagation map for each road over a network. Therefore, this learning framework seamlessly integrates the vantages of LSTM deep learning approach with the traffic flow dynamics model to solve a network-level high-resolution traffic speed propagation and prediction problem.

Below we introduce the mathematical map of LSTM- $s$ ,  $s \in S$  on an individual road and then we present how we connect them to the road network topology (Fig. 3). Mathematically, we would like to build up the relationship,  $F_s(\cdot)$  in Eq. (7) for a road  $s$ , which predicts future the boundary conditions of a road at the next time step using historical neighbor traffic speed data profile in the network. To implement this deep learning approach, we explain each element in Eq. (7) as follows. First, we choose the LSTM introduced in

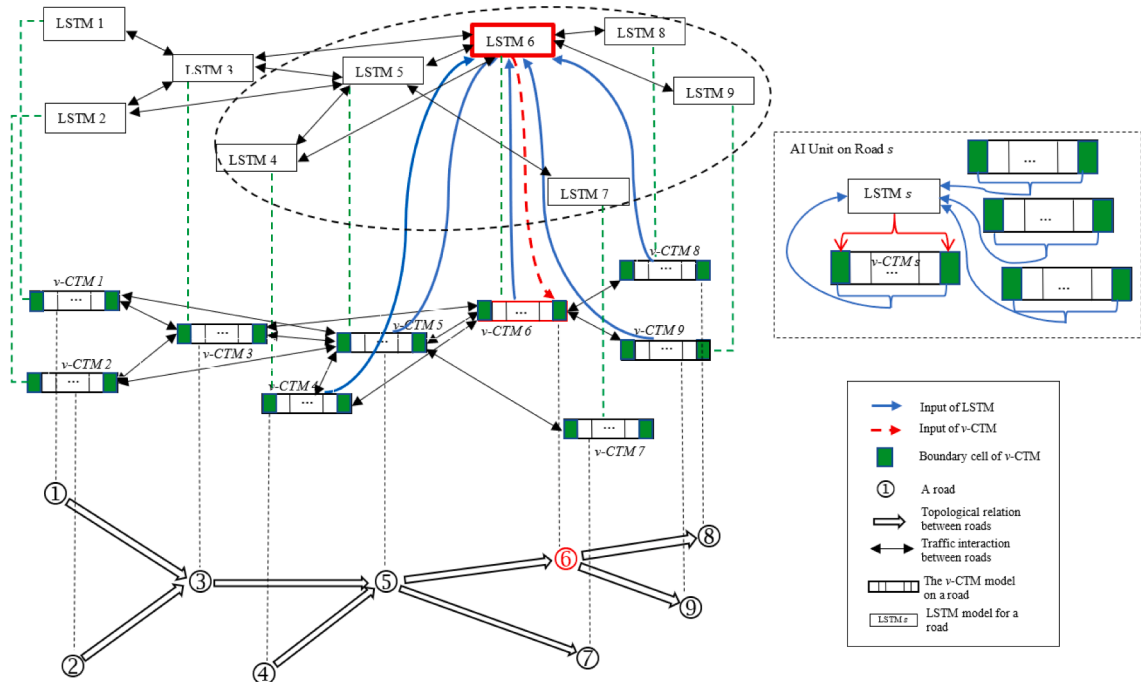


Fig. 3. The Deep Learning Framework of TD<sup>2</sup>-DL.

(Hochreiter and Schmidhuber, 1997) as our deep learning machine  $F_s(\cdot)$  since it is able to memorize the time series process and has been successfully used to solve many and time-series analysis problems (Sundermeyer et al., 2012; Zhang and Xie, 2007), including traffic flow analysis (Bogaerts et al., 2020; Fu et al., 2016; Hua et al., 2019) in the past. Each LSTM-s is with multiple-input and single-output structure. The inputs of the LSTM include the neighboring-roads speed profile of road  $s$  within the historical horizon  $\underline{h}$ ,  $\{V_s(k - \kappa)\}_{\kappa=0}^{\underline{h}}$  and the speed profile of road  $s$  at current time step,  $v_s(k)$ . Note that those input data are generated by the corresponding  $v$ -CTM. The output of LSTM-s is the boundary conditions of road  $s$  at next time step,  $v_s^B(k + 1)$ . Clearly, the parameter  $\underline{h}$  is critical since it determines how much temporal-spatial flow dependency should be considered by the input  $\{V_s(k - \kappa)\}_{\kappa=0}^{\underline{h}}$  by a LSTM. Moreover, to build the map of the LSTM, it is important to specify the temporal-spatial scope involved in the historical neighboring-roads speed profiles  $\{V_s(k - \kappa)\}_{\kappa=0}^{\underline{h}}$ . They affect the computation load and prediction accuracy of the LSTM. We explain this technical detail as follows.

$$v_s^B(k + 1) = F_s\left(\{V_s(k - \kappa)\}_{\kappa=0}^{\underline{h}}, v_s(k)\right), s \in S \quad (7)$$

$$V_s(k - \kappa) = \left[ v_{s'}(k_0'), v_s(k - \kappa), v_{s''}(k_0''), s' \in \{s^-, s^+\} \right], 0 \leq \kappa < \underline{h}, s \in S \quad (8)$$

$$k' = \{k_0'\} \text{ and } k'' = \{k_0''\}$$

$$\{s^-, s^+\} = \{s' | \mathcal{T}_{xy} < \underline{h}, \mathcal{T}_{xy} = \mathcal{T}_{s's} \text{ or } \mathcal{T}_{ss'} s' \in S\} \quad (9)$$

$$k_0' = k - \kappa + 1 - \frac{T_{s's}}{\Delta t}, k_0'' = k - \kappa + 1 - \frac{T_{ss'}}{\Delta t}, s' \in \{s^-, s^+\}, 0 \leq \kappa < \underline{h} \quad (10)$$

We use one element  $V_s(k - \kappa)$  in Eq. (8) as an example to explain our approach to determine it. Recall that  $V_s(k - \kappa) = [v_{s^-}(k'), v_s(k - \kappa), v_{s^+}(k'')]$ ,  $\forall k \in \mathbb{Z}_+, s \in S$ , which involves the speed profiles of road  $s$  at time step  $k - \kappa$ , the speed profiles of upstream road in sets  $s^-$  at a time step in set  $k'$ , and the speed profiles of downstream road in sets  $s^+$  at a time step in set  $k''$ . Therefore, to determine  $V_s(k - \kappa)$ , we need to specify a local neighborhood network formed by the roads in the united set  $\{s^-, s^+\}$  and also determine the corresponding time step sets  $k' = \{k_0'\}$  and  $k'' = \{k_0''\}$ . Mainly, to predict  $v_s^B(k + 1)$ , the speed profile of a neighbor road,  $s'$  will be considered in  $V_s(k - \kappa)$  if its flow propagates to road  $s$  within the lookback horizon  $\underline{h}$ , i.e.,  $\mathcal{T}_{xy} < \underline{h}$ ,  $\mathcal{T}_{xy} = \mathcal{T}_{s's} \text{ or } \mathcal{T}_{ss'} s' \in S$ . Accordingly, Eq. (9) determines a local road network formed by roads in the set of  $\{s^-, s^+\}$ , whose speed profile would be considered in the LSTM built for road  $s$ . Next, we noticed that  $v_{s^-}(k') = [v_{s_1^-}(k_1'), v_{s_2^-}(k_2'), \dots, v_{s_m^-}(k_m')]$ . Therefore, each speed profile in the set of  $v_{s^-}$  is interested at different time step  $k_0'$  since their flow takes different delay to impact the road flow on road  $s$ . We thus use Eq. (10) to determine the time step  $k_0'$  for each road within the set of  $s^-$ . Specifically, if the traffic on  $s'$  need to spend  $\mathcal{T}_{s's}$  in propagating to road  $s$ , the  $k_0'$  will be equal to  $k - \kappa + 1 - \frac{T_{s's}}{\Delta t}$ . The same approach is applied to determine  $k_0'' \in k''$  for each road within set  $s^+$ . Combining Eqs. (8)–(10) and also the corresponding historical speed data stored by LSTM on each road, we obtain the historical neighborhood speed profile  $V_s(k - \kappa)$  and further the input  $\{V_s(k - \kappa)\}_{\kappa=0}^{\underline{h}}$  in Eq. (7). Fig. 4 illustrates the evolvement of the boundary condition prediction on road  $s$  from step  $k$  to step  $k + 1$ , and move forward to reach the perdition horizon  $\bar{h}$ .

Overall, Figs. 3 and 4 together illustrate that LSTM-s and  $v$ -CTM-s sequentially provide input data to each other, and then they evolve under this TD<sup>2</sup>-DL framework to predict speed distribution and propagation on a road over time and then extend to network-wide through flow interdependency. Specifically, using predicted boundary conditions on every road at step  $k$  (like  $v_s^B(k)$ ,  $s \in S$ ) from the LSTM models as inputs, the  $v$ -CTM models predict the speed profile ( $\hat{v}_s(k)$ ) for each respective roads  $\in S$ . Then, for a road  $s \in S$ , we also obtain its neighboring traffic speed profile  $\{V_s(k)\}$ . Next, the predicted speed profile of road  $s$ ,  $\hat{v}_s(k)$  combining with the neighboring

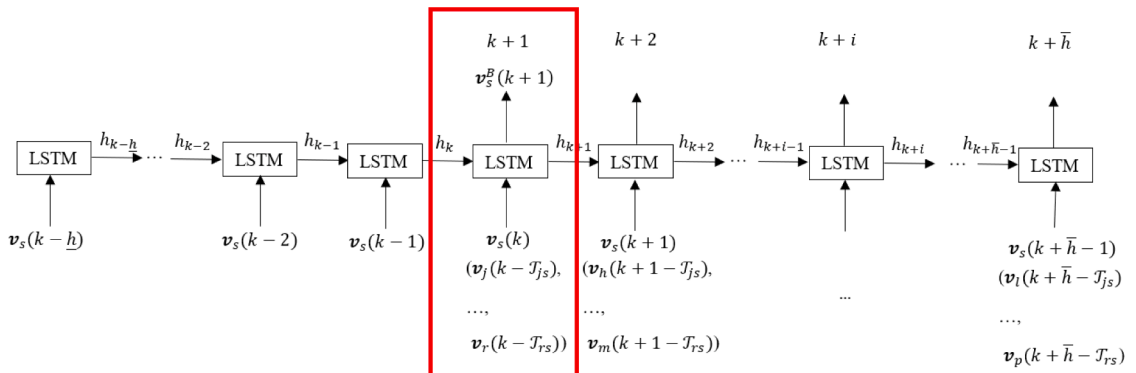


Fig. 4. The boundary condition of CTM-s update by LSTM-s on a road  $s$  at different time steps.

roads' speed profiles  $\{V_s(k)\}$  is taken as inputs to the LSTM on road  $s$  for predicting the boundary conditions at the next time step  $k + 1$  ( $v_s^B(k + 1)$ ). Repeating this process on each road, the network speed profiles  $\{V(k + \kappa)\}_{\kappa=1}^{\bar{h}}$  can be predicted at each time step in a rolling time horizon  $\bar{h}$ , considering the traffic dependency among neighbors.

#### 4.4. Initial condition estimation by Extended Kalman Filter

This section discusses how to accurately estimate the initial condition of the state transmission model by integrating the field data and the mathematical prediction. Recall that the initial condition is defined as the road speed vector at the time step  $k$ ,  $v_s(k)$ , which will be used as one of the inputs to Eq. (2) to predict the road speed vector in a prediction horizon,  $\{v_s(k + \kappa)\}_{\kappa=1}^{\bar{h}}$ . The initial condition can be obtained from two sources in this study. On the one hand, we may use the field data  $v_s^o(k)$  collected by probe vehicles or fixed traffic sensors to estimate the initial traffic speed in each cell at step  $k \in \mathbb{Z}_+$ , i.e.,  $v_s^o(k) = v_s(k)$ . The traffic data observed by probe vehicles perhaps cannot accurately reflect the real traffic speed profiles because not all vehicles are probe vehicles. On the other hand, we use the estimated road speed from the state transmission model from  $k - 1$  step,  $\hat{v}_s(k)$  to represent the initial traffic condition at time step  $k \in \mathbb{Z}_+$ , i.e.,  $v_s(k) = \hat{v}_s(k)$ . It is theoretically correct but cannot demonstrate the randomness and uncertainty of traffic propagation in reality. By factoring the pros and cons of both data sources, this section estimates initial conditions,  $v_s(k)$ , by assimilating the theoretical assessment,  $\hat{v}_s(k)$ , with the real-time observation  $v_s^o(k)$  at each time step  $k \in \mathbb{Z}_+$ . Specifically, we will implement Extended Kalman Filter (EKF) (Tampère and Immers, 2007), to calibrate initial conditions,  $v_s(k)$ , by assimilating the theoretical value,  $\hat{v}_s(k)$ , and the observation value,  $v_s^o(k)$ . Fig. 5 demonstrates the integration of EKF,  $\nu$ -CTM, and the LSTM. Namely, we implement an EKF,  $\nu$ -CTM and LSTM on each road. For a given road  $s$ , both the traffic speed profiles estimated from the  $\nu$ -CTM and the speed observation in field are taken as inputs of the EKF on the road. The output of the EKF will be used for two purposes: (i) as an input of the LSTMs on this road and other roads to predict future boundary conditions; (ii) as the initial condition of the  $\nu$ -CTM model on road  $s$  to predict its future traffic speed propagation in our TD<sup>2</sup>-DL. Below we justify the application of EKF and introduce the technical detail for implementing the EKF in this study.

The Kalman filter is the optimal sequential data assimilation method developed for linear dynamics and measurement processes with Gaussian error statistics. The EKF is a variant of the Kalman filter that can be used for nonlinear problems. The LWR model, as well as its discretized formation, CTM or  $\nu$ -CTM (Eq. (2)) are nonlinear derivative dynamic functions. Thus, the least square  $v_s(k)$  can be obtained by using EKF to assimilate the theoretical value and observation. Accordingly, we have Eqs. (11) and (12), respectively to relate  $v_s(k)$ ,  $v_s^o(k)$  and  $\hat{v}_s(k)$ .

$$v_s(k) = \hat{v}_s(k) + \xi(k), \quad s \in S \quad (11)$$

where,  $\xi(k)$  denotes the error of the model output with the assumption of zero-mean Gaussian with covariance  $O^{N_s \times N_s}$ ,  $\xi(k) \sim \mathcal{N}(0, O)$ .

$$v_s(k) = v_s^o(k) + \varepsilon(k), \quad s \in S \quad (12)$$

Where,  $\varepsilon(k)$  is the Gaussian observation noise with covariance  $R^{N_s \times N_s}$ ,  $\varepsilon(k) \sim \mathcal{N}(0, R)$ .

The process of implementing EKF is illustrated by Eqs. (13)–(17). Briefly, Eqs. (13) and (14) indicate that the assimilated value,  $v_s(k)$ , is estimated by the weighted sum of the theoretical value,  $\hat{v}_s(k)$ , and the observation value,  $v_s^o(k)$ . The weight,  $K_s(k)$  is estimated by Eq. (15). It relies on the covariance of  $v_s(k)$  and  $P_s(k)$  which is determined by Eqs. (16) and (17). Namely, the estimation of  $P_s(k)$  relies on both the updated covariance,  $\tilde{P}_s(k - 1)$ , conducted in the previous step and the Jacobian matrix of the state transmission model,  $F_s$ .

$$v_s(k) = \hat{v}_s(k) + K_s(k) \text{dif}, \quad s \in S \quad (13)$$

$$\text{dif} = v_s^o(k) - \hat{v}_s(k), \quad s \in S \quad (14)$$

$$K_s(k) = P_s(k)(R + P_s(k))^{-1}, \quad s \in S \quad (15)$$

$$P_s(k) = F_s(k - 1)\tilde{P}_s(k - 1)F_s(k - 1)^T + O, \quad s \in S \quad (16)$$

$$\tilde{P}_s(k) = (I - K(k)F_s(k))P(k), \quad s \in S \quad (17)$$

where,  $F_s(k)$  is the Jacobian matrix of the state transmission model in each time stamp with  $N_s \times N_s$ ,  $R$  and  $O$  are the state transmission model error covariance and observation error covariance for  $N_s$  cells on link  $s$ .

The traffic dynamics model has different formations for different problems. This study uses  $\nu$ -CTM as a state transmission model shown in Eqs. (2)–(4). The  $\nu$ -CTM is a piece-wise function, thus the corresponding Jacobian matrix,  $F_s(k - 1)$  in Eq. (3) needs to consider different scenarios as shown in Eq. (3). Each of them has a specific formation, as illustrated by Eqs. (18)–(20). In addition, the input parameters  $R$ ,  $O$ , and  $P(0)$  are critical in applying EKF and is very difficult to be determined. Thus, they are taken as variable, which can be optimized by implementing the EKF on historical data with the objective function of Root Mean Square Error (RMSE) using heuristic algorithms.

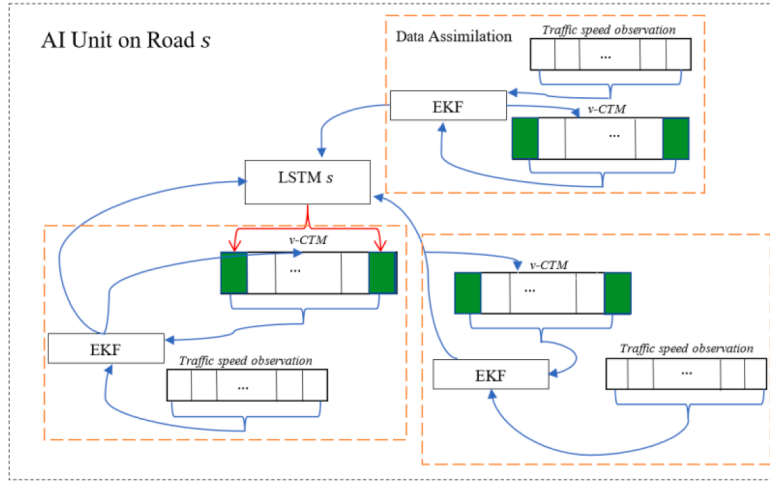


Fig. 5. The data imputation operation in an AI Unit.

$$F_s(k-1) = \begin{bmatrix} \frac{\partial f_1}{\partial v_1} & \frac{\partial f_1}{\partial v_2} & \dots & \frac{\partial f_1}{\partial v_{N_s}} \\ \frac{\partial f_2}{\partial v_1} & \frac{\partial f_2}{\partial v_2} & \dots & \frac{\partial f_2}{\partial v_{N_s}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_{N_s}}{\partial v_1} & \frac{\partial f_{N_s}}{\partial v_2} & \dots & \frac{\partial f_{N_s}}{\partial v_{N_s}} \end{bmatrix} \Big|_{\tilde{v}(k-1)}, s \in S, k \in \mathbb{Z}_+ \quad (18)$$

$$\begin{aligned} \frac{\partial f_i}{\partial v_i} &= 1 - \frac{\Delta t}{\Delta x} \left[ \frac{dg(v_i, v_{i+1})}{dv_i} - \frac{dg(v_{i-1}, v_i)}{dv_i} \right], i \in N_s, s \in S \\ \frac{\partial f_i}{\partial v_{i-1}} &= \frac{\Delta t}{\Delta x} \frac{dg(v_{i-1}, v_i)}{dv_{i-1}}, i \in N_s, s \in S \\ \frac{\partial f_i}{\partial v_{i+1}} &= -\frac{\Delta t}{\Delta x} \frac{dg(v_i, v_{i+1})}{dv_{i+1}}, i \in N_s, s \in S \\ \frac{\partial f_i}{\partial v_j} &= 0, j \neq i-1, i, \text{ and } i+1, i, j \in N_s, s \in S \end{aligned} \quad (19)$$

$$\begin{aligned} \frac{dg(v_i, v_{i+1})}{dv_i} &= \begin{cases} 0 & v_i \leq v_{i+1} \leq v^c \\ 0 & v_i \leq v^c \leq v_{i+1} \\ 2v_i - v_f & v^c \leq v_i \leq v_{i+1} \\ 2v_i - v_f & v_i \geq v_{i+1} \text{ \& } E(v_i) > E(v_{i+1}) \\ 0 & v_i \geq v_{i+1} \text{ \& } E(v_i) \leq E(v_{i+1}) \end{cases}, i \in N_s, s \in S \\ \frac{dg(v_{i-1}, v_i)}{dv_i} &= \begin{cases} 2v_i - v_f & v_{i-1} \leq v_i \leq v^c \\ 0 & v_{i-1} \leq v^c \leq v_i \\ 0 & v^c \leq v_{i-1} \leq v_i \\ 0 & v_{i-1} \geq v_i \text{ \& } E(v_{i-1}) > E(v_i) \\ 2v_i - v_f & v_{i-1} \geq v_i \text{ \& } E(v_{i-1}) \leq E(v_i) \end{cases}, i \in N_s, s \in S \end{aligned} \quad (20)$$

where,  $f_i$  is given in Eq. (2),  $v^c$  is the speed corresponding minimal value of  $E(v) = v^2 - vv_f$  and  $v_f$  is the free flow speed.

#### 4.5. Data Imputation

The TD<sup>2</sup>-DL approach requires support from a large quantity of traffic data to estimate initial conditions and boundary conditions.

We may incur missing data in some cells  $i \in N_s$ ,  $s \in S$  at a specific time interval  $k \in \mathbb{Z}_+$  due to lack of sensors (fixed sensors or probe vehicles) or their malfunctions. Those data imputation problems will affect the accuracy of traffic speed propagation prediction and needs to be carefully addressed (Duan et al., 2016). To accurately impute the missing or corrupted data at time step  $k$  on cell  $i$ , say  $v_i(k)$ , it is beneficial to leverage high dimensional data, like the historical data,  $\bar{v}_i(k)$ , the predicted data at time step  $k - 1$ ,  $\hat{v}_i(k)$ , by Eq. (2), as well as the data observed at neighboring cells,  $v_{i-1}^o(k)$ ,  $v_{i+1}^o(k)$  for cell  $i$ . On the other hand, the relation between  $v_i(k)$  and the available information presents complicated nonlinearity. The Feedforward Neural Network (FNN) is an effective tool to capture this complexity without the knowledge of the model formation. This study implements an FNN to do the data imputation described as follows.

First of all, we need to recognize the corrupted trajectory data (outliers) collected from probe vehicles. To do that, we develop a tensor (3-dimensional matrix) with the first two dimensions representing the temporal index and spatial index, after we discretize the time-space area for implementing  $v$ -CTM, while the third dimension has seven elements corresponding to the days of a week. For each entry in this tensor, the historical traffic data at the corresponding time, location, and day of a week are analyzed statistically to determine the thresholds for the outliers. Here, the lower and upper thresholds are defined as  $Q1-3IQ$  (0 if  $Q3-3IQ$  is less than 0) and  $Q3+3IQ$  as (Snedecor and Cochran, 1989), in which  $Q1$  is the lower quartile,  $Q3$  is the upper quartile, and  $IQ=(Q3 - Q1)$  is the interquartile range (Snedecor and Cochran, 1989). The quartiles ( $Q1$  and  $Q3$ ) are determined by the data in the tensor. The obtained threshold table will be used to identify corrupted data.

Next, we impute the missing or corrupted data by leveraging three pieces of information: the historical mean speed,  $\bar{v}_i(k)$ , the predicted traffic speed at the previous time step,  $\hat{v}_i(k - 1)$ , and the traffic speed<sup>2</sup> in neighborhood cells,  $v_{i-1}(k)$ ,  $v_{i+1}(k)$ . Each of them provides essential features for us to develop an efficient learning approach. Mainly, the historical mean speed,  $\bar{v}_i(k)$ , can provide us with information about traffic continuity, and it takes the average speed at cell  $i$  at the same time every week. The estimated data from the state transmission model tells us the possible traffic speed according to traffic physics (LWR model). Last, the traffic speed at upstream and downstream cells can provide valuable information to estimate the traffic speed in a specific cell due to the temporal-spatial traffic dependency. In order to integrate these three features and accurately capture the complicated nonlinear relationships thereof, we build a fully connected FNN, which maps the three features mentioned above to the traffic speed,  $v_i(k)$ . Eq. (21) shows that we can attribute the missing or corrupted entries leveraging the average traffic speed,  $\bar{v}_i(k)$ , the current estimated traffic speed,  $\hat{v}_i(k)$ , and the observed neighboring traffic speed,  $v_{i-1}^o(k)$ ,  $v_{i+1}^o(k)$ , through the trained FNN model.

$$v_i(k) = FNN(\bar{v}_i(k), v_{i-1}(k), v_{i+1}(k), \hat{v}_i(k)), i \in N_s, s \in S, k \in \mathbb{Z}_+ \quad (21)$$

## 5. Experiment

### 5.1. Experiment setting

This study develops numerical experimentations to validate the proposed approaches. This section introduces our experiment setting, including the tested, dataset, hyper-parameters for the  $v$ -CTM model and Neural Network architectures and the performance measures. Our case study is built upon a road network shown in Fig. 6, which includes seven labeled roads: a 1.8-mile-long road section on the I-5 interstate freeway northbound, a 1-mile-long road on I-5 interstate freeway southbound, two 4000-foot-long roads and a 2.1-mile-long road on WA-18 highway eastbound and two roads on WA-167 highway eastbound respectively with length 1.46 miles, 2.06 miles. The number of lanes on these roads varies from 2 to 5. Loop detectors installed on these roads provide traffic data, including volume, occupancy, and mean speed, from 11/01/2015 to 12/31/2015 with 5 minutes data collection interval. This study will simulate vehicle trajectories on the road network and then randomly pick  $a\%$  ( $a = 1, 3, 5, 10, 30, 50, 80$ ) of the data as connected vehicle data. As for the corrupted data, 0.1% of traffic data are randomly selected and replaced by a random value between 100 and 200. In addition, the traffic data are divided into two groups, the training set, and the testing set. The training set includes the first six weeks of this data, and the others are taken as the testing set. During the training process for the LSTM model of each road, the traffic speed on the neighboring roads is taken as input features, and the traffic speed at the two boundaries is taken as the output. Note that the trained LSTM model cannot fully guarantee the predicted traffic speed always satisfies the constraint in Eq. (4). Given all our training data positive, we barely got negative boundary traffic speed. But we did observe the predicted boundary traffic speed greater than free-flow speed. When this happens, we will round the predicted value down to the free flow speed. To implement the TD<sup>2</sup>-DL approach, we set up the parameters shown in Table 1. More exactly, we estimate the free-flow speed ( $v_f$ ) for all the roads according to Greenshields' model, which is fitted by historical data. The traffic speed prediction update interval,  $\Delta t$ , is set to be 5 seconds. Accordingly the length of each cell,  $\Delta l$ , is given in Table 1, while satisfy Eq. (6) to maintain stability in the Godunov numerical scheme (Work et al., 2008).

This experiment employs three layers of the FNN model for data imputation and three layers of the LSTM model for boundary conditions prediction with 600 time steps (50 min) "look back" to predict future 360 time stamps (30 min). The hyperparameters of the FNN and the LSTM are set according to the parameters in a previous study (Bogaerts et al., 2020), as shown in Table 2 and Table 3 separately. The training iteration is set to be 1000. In addition, ADAM (Adaptive Momentum Estimation) is used as the optimizer to train the LSTM model, and its dropout regularization method is applied to prevent overfitting with a 20% dropout rate.

<sup>2</sup> As for the first or the last cells, the corresponding boundary condition is used as neighbor traffic speeds.

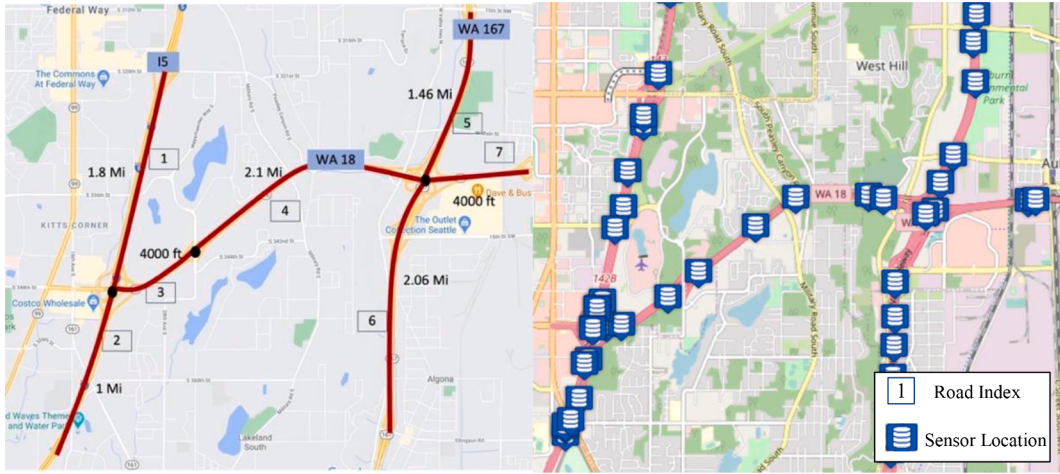


Fig. 6. Real Traffic Network and Loop Detectors Position.

Table 1

| Links      | 1    | 2    | 3     | 4    | 5     | 6     | 7     |
|------------|------|------|-------|------|-------|-------|-------|
| $v_f$      | 68.6 | 67.8 | 62.5  | 63.1 | 60.3  | 60.8  | 62.7  |
| $\Delta l$ | 0.18 | 0.1  | 0.085 | 0.21 | 0.146 | 0.206 | 0.085 |
| Dimensions | 10   | 10   | 9     | 10   | 10    | 10    | 9     |

Table 2

| FNN   | Major Parameters | Activation | Dimensions     |
|-------|------------------|------------|----------------|
| Dense | $n = 50$         | None       | $3 \times 50$  |
| Dense | $n = 50$         | None       | $50 \times 50$ |
| Dense | $n = 50$         | Sigmoid    | $50 \times 1$  |

Table 3

| LSTM  | # of cells            | Activation | Dimensions               |
|-------|-----------------------|------------|--------------------------|
| Dense | $n = 200$             | None       | $Dim_{input} \times 200$ |
| LSTM  | $n = 200$ dropout=0.2 | ReLU       | $200 \times 200$         |
| LSTM  | $n = 200$ dropout=0.2 | ReLU       | $200 \times 200$         |
| LSTM  | $n = 200$ dropout=0.2 | ReLU       | $200 \times 200$         |
| Dense | $n = 200$             | Sigmoid    | $200 \times 2$           |

## 5.2. Prediction Performance

This section will demonstrate the capabilities and performance of the TD<sup>2</sup>-DL. The accuracy of the proposed method is evaluated by the well-accepted metrics proposed in (Bogaerts et al., 2020), including Root-Mean-Square-Error (RMSE) and Mean-Absolute-Percentage-Error (MAPE), as shown in Eqs. (22) and (23).

$$RMSE = \frac{1}{S} \sum_{s=1}^S \sqrt{\frac{1}{N_s} \sum_{i \in N_s} |v_{i,s} - \hat{v}_{i,s}|^2} \quad (22)$$

$$MAPE = \frac{1}{S} \sum_{s=1}^S \frac{1}{N_s} \sum_{i \in N_s} |v_{i,s} - \hat{v}_{i,s}| / v_{i,s} \quad (23)$$

where  $\hat{v}_i$  is the predicted value, and  $v_i$  is the ground truth value in cell  $i \in N_s$ ,  $s \in S$ .

Fig. 7 illustrates an example of our high-resolution traffic speed propagation prediction for Road 2 on 12/17/2015.

Mathematically, it can be presented as a matrix in Fig. 7 (a), and we further visualize it using the heatmap, Fig. 7 (b). This traffic speed profile can tell us the traffic speed spatial distribution propagation over time. More exactly, Fig. 7 shows that the congestion occurred at 6:53:10 from Road 2's downstream end (cell 10). It took around 35 seconds to propagate back and reached approximately 0.4 miles away from the downstream end. Using this prediction, we can estimate that the backward shockwave speed is about 40 mph. In addition, this heat map tells us that there are two shockwaves occurring in this temporal-spatial area. A forward shockwave occurred from the third time step (6:53:20) from the upstream end (cell 1), and a backward shockwave occurred from the first time step (6:53:10) from the downstream end (cell 10). These two shockwaves met at the seventh time step (6:53:40), at which a new shockwave formed. Those comprehensive traffic information will support various applications better and demonstrate the strong capabilities of our approach.

We next conduct the experiments to demonstrate the accuracy of the predicted high-resolution speed profile under the prediction horizons ( $\bar{h}$ ) varying from 5 seconds to 1800 seconds with the step size equal to 5 seconds. More exactly, for each prediction horizon, we run the experiments in 30 minutes and calculate the accuracy (MAPE and RMSE) of the predicted speed profile as compared to its ground truth at each time step. Then, we measure the average accuracy for the setting of each prediction horizon. The results in Fig. 8. show that the prediction error is less than 3% (RMSE around two mph) as the prediction horizon is less than 10 minutes. The error increases quickly as we increase the prediction horizon from 10 minutes to 25 minutes and then slows down after the horizon is longer than 25 minutes. These results indicate that the TD<sup>2</sup>-DL approach performs well for a short prediction horizon (e.g., less than 10 minutes in this experiment) thanks for the consideration of the temporal-spatial flow dependency and traffic flow continuity. However, the prediction error will accumulate as the prediction horizon increases (e.g., 10–25 minutes). Then it will worsen the prediction performance.

This study also applied the TD<sup>2</sup>-DL to the other two other datasets collected during 05/01/2011–06/31/2011 and 09/01/2011–10/31/2011, which present different traffic conditions during the summer and fall two seasons. For each dataset, we take the data collected in the first six weeks as the training set and the rest as the testing set. The average performance is illustrated in Fig. 9, from which we observed a similar performance to Fig. 8. This validates the effectiveness of the TD<sup>2</sup>-DL.

Furthermore, we tested the out-of-training-distribution (OOD) generalization performances of the TD<sup>2</sup>-DL under nonrecurrent traffic conditions. More exactly, we test the performance of the TD<sup>2</sup>-DL framework on our testbed given a traffic accident occurred from 4:51 PM to 5:20 PM on 12/15/2015 at the 8<sup>th</sup> cell on the road 2. This accident blocked a lane on the road. The results in Fig. 10. illustrates that the prediction error (i.e., MAPE) varies from around 4% to 17% within the tested prediction horizon. Compared to the scenarios without traffic accidents, the prediction accuracy reduced from about 1.6% to 4.2%. This corresponds to a speed error of around 1 to 2 mile (s) per hour, which is acceptable in many applications. Therefore, we confirm that the TD<sup>2</sup>-DL also performs well even under the interference of a traffic accident.

### 5.3. Comparison versus baseline algorithms

This section conducts experiments to illustrate the merits of the TD<sup>2</sup>-DL approach versus the other existing methods to show its competitiveness. The selected benchmark methods include a traditional time-series analysis method (ARIMA), a basic machine learning method (SVR), a deep learning model (GRU), a physics-inform neural network: PIDL (Shi et al., 2021), a graph-based deep learning method: Graph CNN-LSTM neural network (Bogaerts et al., 2020), and a temporal-spatial attention neural network: APTN (Shi et al., 2020). The autoregressive integrated moving average (ARIMA) is set up by the parameters ( $p, d, q$ ) used in (Kumar and Vanajakshi, 2015), in which the degree of differencing ( $d$ ) is set to be 1, and the other parameters, the lag order ( $p$ ) and the order of moving average ( $q$ ) are set as 1 and 0. The input data takes the boundary conditions,  $v_s(t, 0), v_s(t, N_s + 1), t = k, k - 1, k - 2, \dots, k - 600, s = 1, 2, \dots, 7$ . Support vector regression (SVR) calculates the curve by fitting the input feature vectors into the high-dimensional space using the kernel function. The corresponding input data select the time series estimated link speed,  $v_s(t), t = k, k - 1, k - 2, \dots, k - 600$ . For each road, a PIDL neural network is built up taking the speed data collected from probe vehicles during  $t = k, k - 1, k - 2, \dots, k - 600$  as input to predict its future traffic speed distribution. The Graph CNN-LSTM neural network and APTN takes the collected traffic speed on every road in the whole network as input, and its output is also the network-wide traffic speed. Recall that the TD<sup>2</sup>-DL predicts

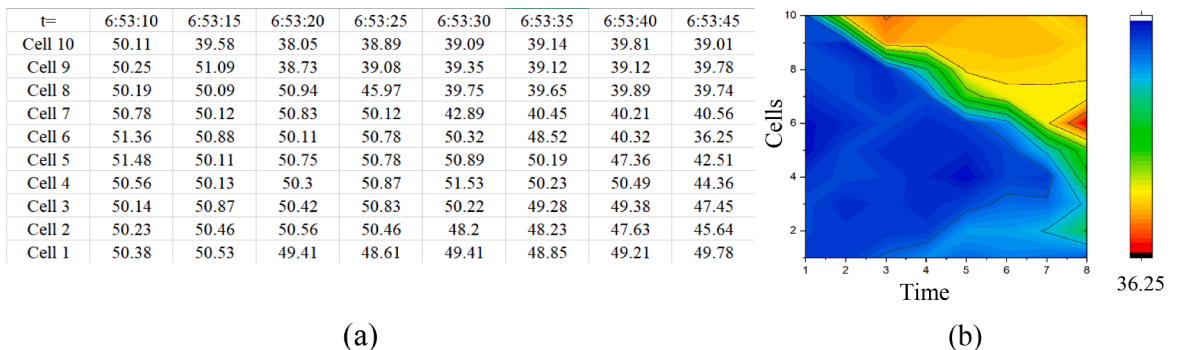


Fig. 7. Traffic Speed Distribution on Road 2.

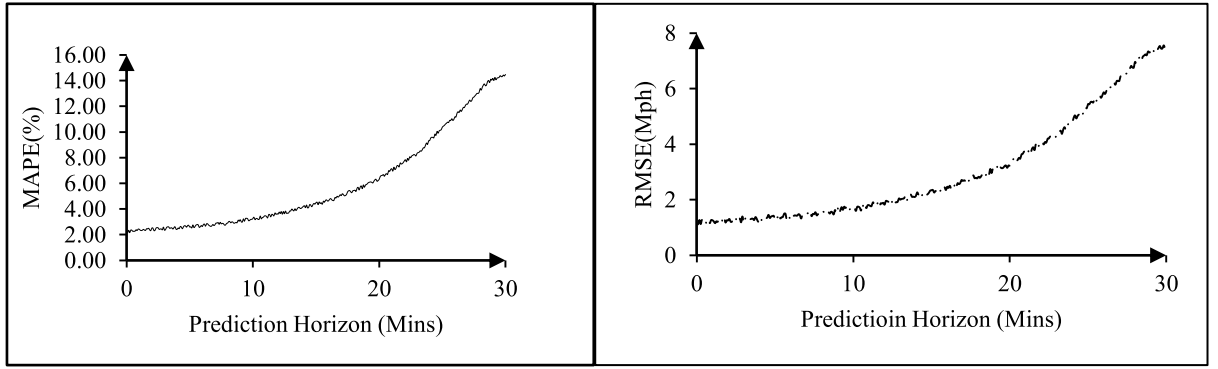


Fig. 8. The Average Prediction Accuracy vs. Prediction Horizon.

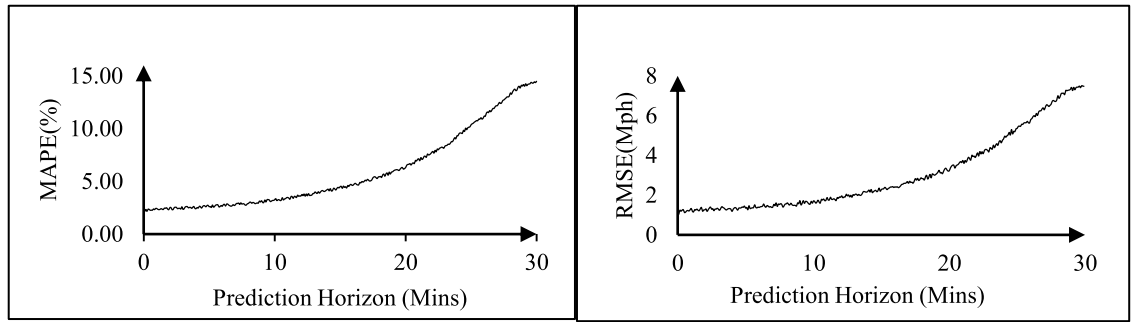


Fig. 9. The Average Performance of TD<sup>2</sup>-DL on The Other Two Datasets.

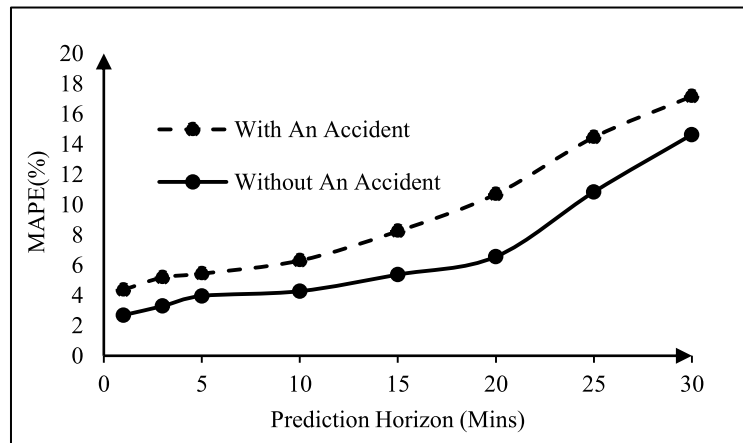


Fig. 10. Prediction Accuracy on 12/15/2015 with a Traffic Accident at Road 2.

a speed vector covering spatial variation at each time step. However, all those benchmark approaches provide a scalar value representing a point/average speed prediction of a road. To them comparable, the speed vector predicted by the TD<sup>2</sup>-DL on each road at each time step (a vector at each time step) is averaged to a scalar value and then compared to the ground truth to measure the errors. Fig. 11 provides the results for predicting network-wide traffic speed on the testbed in the future 30 minutes, using all the tested approaches. The results show that the TD<sup>2</sup>-DL approach is significantly better than ARIMA and SVR and shows apparent vantage to the recently developed method, PIDL, Graph CNN-LSTM, and APTN. Overall, the TD<sup>2</sup>-DL approach outperforms all other methods in this comparison, especially when the prediction horizon is set within 10 to 20 minutes. Moreover, we noticed that deep learning methods such as PIDL, Graph CNN-LSTM, APTN, and the TD<sup>2</sup>-DL perform better than ARIMA and SVR. Also, PIDL, Graph CNN-LSTM, and APTN have the similar performance in this study. Overall, these experiments confirmed that the TD<sup>2</sup>-DL can reach a better performance than other deep learning methods.

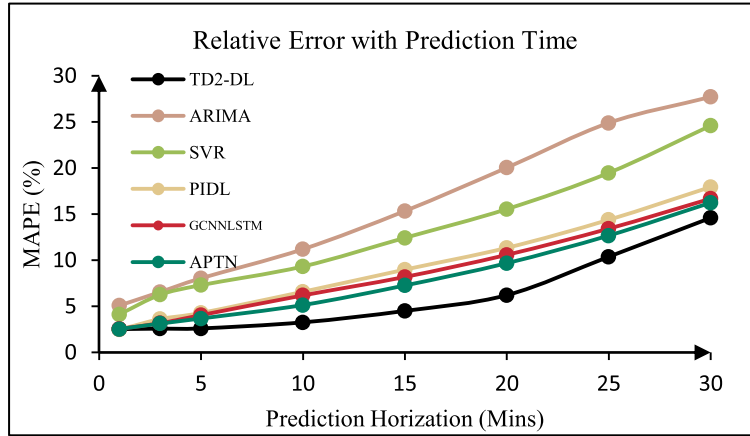


Fig. 11. The Comparison of the TD<sup>2</sup>-DL with Other Approaches.

#### 5.4. Ablation study

This study further conducted experiments to do the ablation study to test the effect of different components of the TD<sup>2</sup>-DL on prediction accuracy. More exactly, we want to investigate the performance of TD<sup>2</sup>-DL by removing the components, including data imputation, traffic dependency, and data assimilation, to better understand their contributions. Note that *v*-CTM is the base of the methodology and is unremovable.

We first validate the contribution of the FNN in the TD<sup>2</sup>-DL for imputing missing or corrupted data under different CAV penetrations. Here, 0.1% of the probe vehicles' speed observations are replaced by a random number between 100 mph and 200 mph. We compare the performances of the TD<sup>2</sup>-DL, respectively using FNN and the baseline method, which fills/replaces the missing/corrupted data by the mean value of historical data at the same cell in the time-space (note that we cannot completely remove data imputation for missing data; Otherwise, the *v*-CTM cannot work). The experiment results in Fig. 12 demonstrate that the FNN leads to a more accurate prediction than using the mean value of the historical data (around 10% improvement in MAPE), even when the probe vehicles' penetration rate is at a lower level (less than 20%). To be noted, we will have more chance to incur missing data under low probe vehicle penetration. Therefore, FNN demonstrates a more apparent contribution. We also noticed that the error decreases fast as we increase the penetration rate of probe vehicles from 0% to 20% and then becomes relatively stable after that. Therefore, we claim that when the probe vehicles' penetration is more than 20%, they can provide us with sufficient traffic data for our approach.

We also set up experiments to validate the contribution of temporal-spatial flow dependency in the TD<sup>2</sup>-DL. Namely, we run the LSTM implemented on each road with and without inputting neighborhood speed profiles. The results in Fig. 13 indicate that factoring temporal-spatial flow dependency among neighbor roads can significantly improve the prediction performance.

Finally, an experiment is set up to validate the effect of data assimilation in the TD<sup>2</sup>-DL. Namely, we run the *v*-CTM implemented on each road under two scenarios. (i) The TD<sup>2</sup>-DL excludes EKF, and then the *v*-CTM only uses the theoretical prediction at the previous time step as the initial condition (e.g., the speed profile predicted at the time step  $k - 1$  is used as the initial condition at step  $k$ ). (ii) The TD<sup>2</sup>-DL includes EKF to assimilate theoretical prediction and real-time collected data to predict initial traffic conditions as the input for *v*-CTM. The results in Fig. 14 indicate that EKF can improve the prediction performance, and the vantage increases with the prediction horizon increasing. This result indicates the significance of the EKF in the TD<sup>2</sup>-DL for improving traffic speed prediction accuracy.

## 6. Conclusion

Traffic speed propagation prediction is an essential component of ITS. It will help promote preventive/proactive traffic management/guidance for mitigating traffic congestion. However, network-wide traffic speed propagation presents complicated dynamics and temporal-spatial dependency. Existing traffic speed prediction approaches, mainly providing average/point traffic speed on a link, cannot efficiently capture such traffic dynamics. This study, therefore, developed a traffic flow dynamics and dependency based deep learning aided approach. It integrates the following methods. We first develop a graph theory-based method and dynamic programming to capture the scope of the traffic temporal-spatial dependency among neighbor links in a road network within a prediction horizon. Then, we discretize a traffic network into homogenous roads and further temporal-spatial cells, built upon which we apply the traffic dynamics model (*v*-CTM) to catch traffic speed propagation on a road. Having recognized the inputs (boundary and initial traffic conditions) of a *v*-CTM highly depend on traffic flow on its neighbors and the speed prediction should keep traffic continuity, this study develops an LSTM deep learning network considering a local temporal-spatial network identified by our dynamic programming. Different from the majority of existing machine learning approaches for traffic flow prediction, this TD<sup>2</sup>-DL joins traffic flow models with deep learning and makes them co-evolve over the network-wide traffic speed propagation prediction. Last, the TD<sup>2</sup>-DL integrates EKF, and FNN approaches to overcome the errors introduced by data collection and inaccurate theoretical estimation. Our numerical experiments validate the efficiency of the TD<sup>2</sup>-DL approaches. The results show that the TD<sup>2</sup>-DL presents accurate predictions on

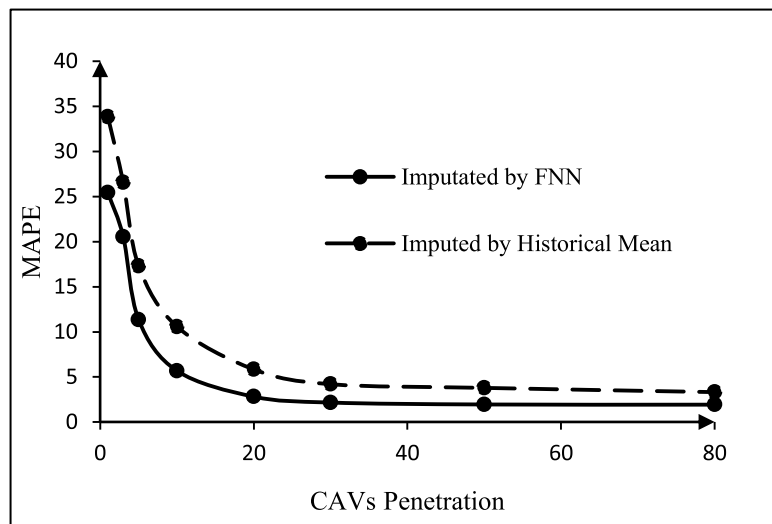


Fig. 12. Significance of FNN under Different CAVs Penetration Rates.

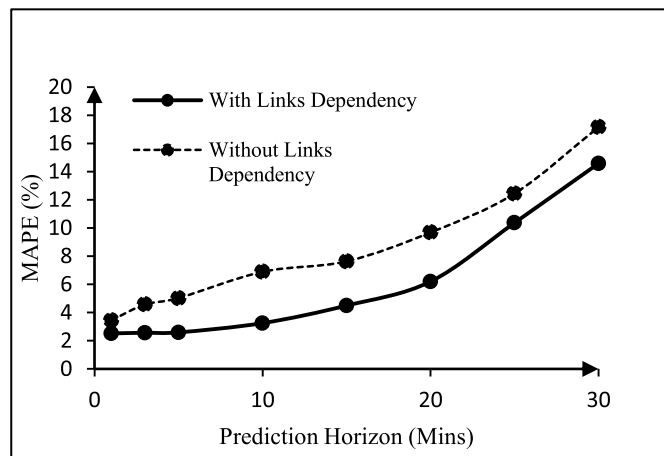


Fig. 13. Significance of factoring traffic dependency.

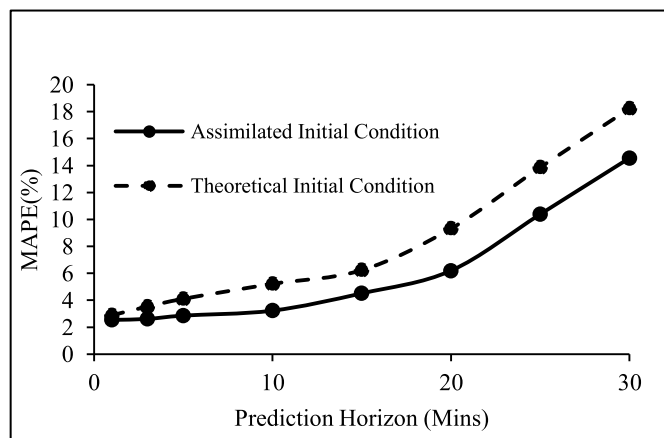


Fig. 14. Significance of factoring data imputation.

average. It outperforms the other baseline models (ARIMA, SVR, and GRU). Factoring temporal-spatial flow dependency among neighbor roads can significantly improve prediction performance. In addition, our data imputation presents a satisfying performance. It enhances the prediction capability of the TD<sup>2</sup>-DL when traffic data are collected from CAVs with low flow penetration rates. Several promising future research can be developed for extending this study. For example, the current TD<sup>2</sup>-DL is developed for freeway networks. But it can be extended to the urban road network by combining v-CTM with traffic signal information and a more complicated deep learning model for predicting the boundary conditions considering neighboring traffic and signal effect. We are interested in integrating this speed propagation prediction with traffic control strategies for improving traffic efficiency. We propose this extension for our future work.

## 7. Author Statements

The authors confirm the contributions to the paper as follows. Dr. Du and her Ph.D. student Hanyi Yang (currently Dr. Yang) initiated the research idea. Dr. Du led and supervised the main methodologies development. Dr. Yang contributed to the development of the technical details in traffic analysis, mathematical derivative, and modeling and experiments. Dr. Zhang and Dr. Ma took part in the discussions and provided the data sources for the experiments. All authors reviewed the results and approved the final version of the manuscript.

## Acknowledgment

This research is partially supported by the National Science Foundation awards CMMI-1901994, CMMI-2213459, and CMMI-1818526. The authors would like to extend their gratitude to the reviewers and editor for their insightful comments, which have increased the quality of this paper.

## REFERENCES

- Asif, M.T., Dauwels, J., Goh, C.Y., Oran, A., Fathi, E., Xu, M., Jaillet, P., 2013. Spatiotemporal patterns in large-scale traffic speed prediction. *IEEE Trans. Intell. Transp. Syst.* 15 (2), 794–804.
- Bogaerts, T., Masegosa, A.D., Angarita-Zapata, J.S., Onieva, E., Hellinckx, P., 2020. A graph CNN-LSTM neural network for short and long-term traffic forecasting based on trajectory data. *Transp. Res. Part C Emerg. Technol.* 112, 62–77.
- Daganzo, C.F., 1997. Fundamentals of transportation and traffic operations, 30. Pergamon, Oxford, pp. 67–160.
- Work, D.B., Tossavainen, O.P., Blandin, S., Bayen, A.M., Iwuchukwu, T., Tracton, K., 2008. An ensemble Kalman filtering approach to highway traffic estimation using GPS-enabled mobile devices. In: 2008 47th IEEE Conference on Decision and Control. IEEE, pp. 5062–5068.
- Strub I, S., M Bayen, A., 2006. Weak formulation of boundary conditions for scalar conservation laws: An application to highway traffic modeling. *Int. J. Robust and Nonlinear Control: IFAC-Affiliated Journal* 16 (16), 733–748.
- Daganzo, Carlos. "The cell transmission model. Part I: A simple dynamic representation of highway traffic." (1992).
- Daganzo, C.F., 1995. The cell transmission model, part II: network traffic. *Transp. Res. B: Methodol.* 29 (2), 79–93.
- Duan, Y., Lv, Y., Liu, Y.L., Wang, F.Y., 2016. An efficient realization of deep learning for traffic data imputation. *Transp. Res. Part C Emerg. Technol.* 72, 168–181.
- Hamed, M.M., Al-Masaeid, H.R., Said, Z.M.B., 1995. Short-term prediction of traffic volume in urban arterials. *J. Transp. Eng.* 121 (3), 249–254.
- Karpathy, A., Johnson, J., & Fei-Fei, L. (2015). I was visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*.
- Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* 1097–1105.
- 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. *Transp. Res. Part C Emerg. Technol.* 54, 187–197.
- Qi, Y., Ishak, S., 2014. A Hidden Markov Model for short-term prediction of traffic conditions on freeways. *Transp. Res. Part C Emerg. Technol.* 43, 95–111.
- Sundermeyer, M., Schlüter, R., Ney, H., 2012. LSTM neural networks for language modeling. In: the Thirteenth annual conference of the international speech communication association.
- Tang, J., Liu, F., Zou, Y., Zhang, W., Wang, Y., 2017. An improved fuzzy neural network for traffic speed prediction considering periodic characteristics. *IEEE Trans. Intell. Transp. Syst.* 18 (9), 2340–2350.
- Wang, J., Chen, R., He, Z., 2019. Traffic speed prediction for urban transportation network: A path-based deep learning approach. *Transp. Res. Part C Emerg. Technol.* 100, 372–385.
- Williams, B.M., Hoel, L.A., 2003. Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. *J. Transp. Eng.* 129 (6), 664–672.
- Xu, D.W., Wang, Y.D., Jia, L.M., Qin, Y., Dong, H.H., 2017. Real-time road traffic state prediction based on ARIMA and Kalman filter. *Front. Inf. Technol. Electron. Eng.* 18 (2), 287–302.
- Ye, Q., Szeto, W.Y., Wong, S.C., 2012. Short-term traffic speed forecasting is based on data recorded at irregular intervals. *IEEE Trans. Intell. Transp. Syst.* 13 (4), 1727–1737.
- Zhang, J., Zheng, Y., Qi, D., Li, R., Yi, X., Li, T., 2018. Predicting citywide crowd flows using deep Spatio-temporal residual networks. *Artificial Intelligence* 259, 147–166.
- Zhang, Z., Li, M., Lin, X., Wang, Y., He, F., 2019. Multistep speed prediction on traffic networks: A deep learning approach considering Spatio-temporal dependencies. *Transp. Res. Part C Emerg. Technol.* 105, 297–322.
- Zhang, K., Zheng, L., Liu, Z., Jia, N., 2019. A deep learning-based multitask model for network-wide traffic speed prediction. *Neurocomputing*.
- Zhang, Y., Xie, Y., 2007. Forecasting of short-term freeway volume with v-support vector machines. *Transp. Res. Rec.* 2024 (1), 92–99.
- Zhang, Y., Liu, Y., 2009. Traffic forecasting using least squares support vector machines. *Transportmetrica* 5 (3), 193–213.
- Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. *Neural Comput.* 9 (8), 1735–1780.
- Fu, R., Zhang, Z., Li, L., 2016. Using LSTM and GRU neural network methods for traffic flow prediction. In: 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC). IEEE, pp. 324–328.
- Hua, Y., Zhao, Z., Li, R., Chen, X., Liu, Z., Zhang, H., 2019. Deep learning with long short-term memory for time series prediction. *IEEE Commun. Mag.* 57 (6), 114–119.
- Tampère, C.M., Immers, L.H., 2007. An extended Kalman filter application for traffic state estimation using CTM with implicit mode switching and dynamic parameters. In: 2007 IEEE Intelligent Transportation Systems Conference. IEEE, pp. 209–216.
- Snedecor, G.W., Cochran, W.G., 1989. Statistical methods, 8thEdn, 54. Ames: Iowa State Univ. Press Iowa, pp. 71–82.
- Daganzo, C.F., 2006. In traffic flow, cellular automata= kinematic waves. *Transp. Res. B: Methodol.* 40 (5), 396–403.

- Zhang, B., 1993. Convergence of the Godunov scheme for a simplified one-dimensional hydrodynamic model for semiconductor devices. *Commun. Math. Phys.* 157 (1), 1–22.
- Blelloch, G.E., 1996. Programming parallel algorithms. *Commun. ACM* 39 (3), 85–97.
- Bellman, R., 1966. Dynamic programming. *Science* 153 (3731), 34–37.
- Yao, B., Chen, C., Cao, Q., Jin, L., Zhang, M., Zhu, H., Yu, B., 2017. Short-term traffic speed prediction for an urban corridor. *Comput.-Aided Civ. Infrastruct. Eng.* 32 (2), 154–169.
- Qi, Y., Ishak, S., 2014. A Hidden Markov Model for short term prediction of traffic conditions on freeways. *Transp. Res. Part C Emerg. Technol.* 43, 95–111.
- Tang, J., Liu, F., Zou, Y., Zhang, W., Wang, Y., 2017. An improved fuzzy neural network for traffic speed prediction considering periodic characteristic. *IEEE Trans. Intell. Transp. Syst.* 18 (9), 2340–2350.
- Huang, S.H., Ran, B., 2003. An application of neural network on traffic speed prediction under adverse weather condition. University of Wisconsin, Madison. Doctoral dissertation.
- Kumar, S.V., Vanajakshi, L., 2015. Short-term traffic flow prediction using seasonal ARIMA model with limited input data. *Eur. Transp. Res. Rev.* 7 (3), 1–9.
- Hegyi, A., De Schutter, B., Hellendoorn, H., 2005. Model predictive control for optimal coordination of ramp metering and variable speed limits. *Transp. Res. Part C Emerg. Technol.* 13 (3), 185–209.
- Yuan, Y., Zhang, Z., Yang, X.T., Zhe, S., 2021. Macroscopic traffic flow modeling with physics regularized Gaussian process: A new insight into machine learning applications in transportation. *Transp. Res. B: Methodol.* 146, 88–110.
- Shi, R., Mo, Z., Huang, K., Di, X., & Du, Q. (2021). Physics-informed deep learning for traffic state estimation. *arXiv preprint arXiv:2101.06580*.
- Barreau, M., Aguiar, M., Liu, J., & Johansson, K. H. (2021). Physics-informed learning for identification and state reconstruction of traffic density. *arXiv preprint arXiv:2103.13852*.
- Rempe, F., Loder, A., Bogenberger, K., 2021. Estimating motorway traffic states with data fusion and physics-informed deep learning. In: 2021 IEEE International Intelligent Transportation Systems Conference (ITSC). IEEE, pp. 2208–2214.
- Shi, R., Mo, Z., Huang, K., Di, X., Du, Q., 2021. A physics-informed deep learning paradigm for traffic state and fundamental diagram estimation. *IEEE Trans. Intell. Transp. Syst.*
- Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2017). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*.
- Ma, C., Zhao, Y., Dai, G., Xu, X., Wong, S.C., 2022. A Novel STFS-CNN-GRU Hybrid Model for Short-Term Traffic Speed Prediction. *IEEE Trans. Intell. Transp. Syst.*
- Zheng, G., Chai, W.K., Katos, V., Walton, M., 2021. A joint temporal-spatial ensemble model for short-term traffic prediction. *Neurocomputing* 457, 26–39.
- Wang, J., Chen, R., He, Z., 2019. Traffic speed prediction for urban transportation network: A path based deep learning approach. *Transp. Res. Part C Emerg. Technol.* 100, 372–385.
- Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., Li, H., 2019. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Trans. Intell. Transp. Syst.* 21 (9), 3848–3858.
- Wang, X., Chen, C., Min, Y., He, J., Yang, B., & Zhang, Y. (2018). Efficient metropolitan traffic prediction based on graph recurrent neural network. *arXiv preprint arXiv:1811.00740*.
- Xie, Z., Lv, W., Huang, S., Lu, Z., Du, B., Huang, R., 2019. Sequential graph neural network for urban road traffic speed prediction. *IEEE Access* 8, 63349–63358.
- Song, C., Lin, Y., Guo, S., Wan, H., 2020. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence* 34 (01), 914–921.
- Zheng, C., Fan, X., Wang, C., Qi, J., 2020. Gman: A graph multi-attention network for traffic prediction. *Proc. Innov. Appl. Artif. Intell.* 34 (01), 1234–1241.
- Guo, S., Lin, Y., Feng, N., Song, C., Wan, H., 2019. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. *Proc. Innov. Appl. Artif. Intell.* 33 (01), 922–929.
- Shi, X., Qi, H., Shen, Y., Wu, G., Yin, B., 2020. A spatial-temporal attention approach for traffic prediction. *IEEE Trans. Intell. Transp. Syst.* 22 (8), 4909–4918.