Towards Predicting Traffic Shockwave Formation and Propagation: A Convolutional Encoder–Decoder Network

Mohammadreza Khajeh Hosseini[1] and Alireza Talebpour, Ph.D.[2]

Abstract: Traffic management strategies have been relying on various congestion prediction methodologies. The prediction accuracy of these methodologies has improved over the years, offering reasonable short-term and midterm predictions of macroscopic traffic measures (i.e., flow, speed, and occupancy/density). Unfortunately, by relying on fixed infrastructure sensors and aggregated data, these prediction methodologies fail to include microscopic traffic flow dynamics in their prediction algorithms. Accordingly, they usually fail to capture the onset of congestion and can only predict the propagation of existing shockwaves. That is, in fact, critical for utilizing effective traffic management strategies because predicting the onset of congestion can significantly help with mitigating it. Addressing this shortcoming in traffic predcition algorithms, this study proposes a deep learning methodology to predict the formation and propagation of traffic shockwaves at the vehicle trajectory level. Assuming the existence of communications between vehicles and infrastructure, the time-space diagram of the study segment serves as the input of the deep neural network, and the output of the network is the predicted propagation of shockwaves on that segment. It is the capability to extract the features embedded in a time-space diagram that allows this methodology to predict the propagation of traffic shockwaves. The proposed approach was tested on both simulation and real-world data, and results show that it can accurately predict shockwave formation and propagation. **DOI: 10.1061/JTEPBS.TEENG-7209.**

Introduction and Background

Traffic shockwaves denote the transition between two different traffic states. With changes in vehicle speeds and space between vehicles, driving dynamics change from state to state. Therefore, vehicles need to react by adjusting their speed and acceleration when the traffic condition changes, which causes shockwaves to propagate. Traffic shocwaves are the main indicator of congestion formation and propagation. They are also important to safety because failure to respond adequately and timely to the change in the traffic conditions (e.g., abrupt speed variations) can result in unsafe driving instances (Chen et al. 2010; Rahman et al. 2012; Gaweesh et al. 2021). Accordingly, capturing and predicting the onset of shockwave formation and predicting its dynamics (i.e., magnitude and propagation speed) can significantly improve traffic safety and operational efficiency.

Accurate characterization and prediction of traffic shockwaves also play key roles in the operations of automated vehicles. One of the core competencies of automated vehicle software is the planning system, which is in charge of making decisions at different levels to take the automated vehicle from an origin to a destination. One of the challenges with the planning process is that the driving environment is dynamic (due to the evolution of the traffic state, e.g., traffic shockwaves, and the movement of traffic agents). Consequently, the planner needs to continuously predict the changes in the driving environment to plan a safe path (i.e., local path planning) and adjust the speed and spacing of the vehicle with respect to the surrounding driving environment (i.e., local feedback control) (Pendleton et al. 2017; Bautista-Camino et al. 2022).

An accurate prediction of the driving environment dynamics can ensure safety and efficiency [including reducing the energy consumption based on the needed power demand (Zhou et al. 2019)] of automated driving. For instance, currently, the approaches adopted for the guidance of automated vehicles (e.g., lane changing) involve the consideration of predicting the future state of the surrounding vehicles based on their current state in terms of their location and speed (physics-based models) without taking into account the other vehicles' response to their environment and how the traffic state could change (e.g., shockwaves) (Claussmann et al. 2019; Lefèvre et al. 2014). Thus, predicting how traffic shockwaves propagate in space and time can help improve both the safety and performance of automated vehicles.

Based on the valuable information that connected vehicles could provide through vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications, this study proposes a methodology to predict the formation and propagation of traffic shockwaves that accounts both for individual drivers' behavior as well as collective traffic state changes. Density, flow, and speed of traffic, as well as their evolution over time and space, are considered indicators of traffic state. Van Lint and Van Hinsbergen (2012) categorized traffic prediction methodologies into three general categories: naive, parametric, and nonparametric. Naive approaches have no data-driven model parameters, and they are generally established on the average of the historical observations or the prediction based on the current state (Eglese et al. 2006).

Parametric methodologies apply to traffic prediction models that utilize a traffic flow model with parameters estimated based on historical data or in conjunction with new observations. Several well-researched parametric models of the macroscopic characteristics of traffic states have been developed, including the fundamental

[1]Ph.D. Candidate, Dept. of Civil and Environmental Engineering, Univ. of Illinois at Urbana-Champaign, Urbana, IL 61801. Email: mk48@illinois.edu

[2]Assistant Professor, Dept. of Civil and Environmental Engineering, Univ. of Illinois at Urbana-Champaign, Urbana, IL 61801 (corresponding author). Email: ataleb@illinois.edu

Note. This manuscript was submitted on December 20, 2021; approved on October 27, 2022; published online on February 3, 2023. Discussion period open until July 3, 2023; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Transportation Engineering, Part A: Systems***, © ASCE, ISSN 2473-2907.

diagram and first- and second-order traffic flow models (Zheng and Su 2016). Balancing between accuracy and model complexity to address irregularities and time changing dynamics is one of the challenges of adopting a parametric traffic prediction methodology. In detail, macroscopic traffic flow models fail to capture the interactions among vehicles and only focus on changes in average traffic states. On the other hand, although microscopic traffic flow models address this limitation of macroscopic models, they often contain strong assumptions and they are computationally expensive.

Nonparametric refers to methodologies that do not depend on traffic flow models and are built on data-driven techniques such as linear regression (Wilby et al. 2014), neural networks (Polson and Sokolov 2017; Lee et al. 2019; Ke et al. 2020), and support vector regression (Castro-Neto et al. 2009), as well as time-series analysis (Kumar and Vanajakshi 2015). With the increased availability of data resources in the last decades, data-driven approaches for traffic prediction have become more popular (Akhtar and Moridpour 2021; Razali et al. 2021). Currently, nonparametric models are mainly based on macro-level data, including flow, density, and speed, but few studies considered individual-level trajectory data for traffic state prediction (Elfar et al. 2018; Khajeh Hosseini and Talebpour 2019; Bogaerts et al. 2020). The impact of abrupt individual behavior on traffic state increases with the increase in flow and density levels. As a result, this study considers the microscopic level interactions for better prediction of the traffic state. A more comprehensive review of traffic prediction models has been given by Seo et al. (2017).

The connected vehicle technology provides opportunities for sharing valuable data with drivers or complementing onboard sensors for automated vehicles. Connected vehicles share safety-related data about their speed and location with other vehicles and traffic control centers to enhance road safety and improve travel efficiency. With such information, it is possible to monitor traffic and track its evolution over time and space. A time-space diagram can be constructed using the individual-level location data transmitted by connected vehicles. Time-space diagrams visualize traffic without abstraction or aggregation. The time-space diagram integrates traffic flow dynamics, vehicle interactions, and shockwave formation and propagation. Khajeh Hosseini and Talebpour (2019) utilized a matrix representation of the time-space diagram to predict the traffic state. Building upon the findings of that study, the objective of this study is to develop a methodology for predicting traffic shockwaves' formation and propagation (i.e., the change of flow and density over time and space) using a similar matrix representation of the time-space diagram. The proposed methodology incorporates a model that can learn from the features embedded in the time-space diagram to predict the propagation of traffic shockwaves.

The remainder of this paper is organized as follows: In the next section, we present details of the proposed methodology, a problem description, and the training process. Following that section, we will discuss the model's qualitative and quantitative performance. This is followed by a validation of the calibrated model using the Next Generation Simulation Models (NGSIM) data set and a discussion of the findings. Finally, the paper ends with concluding remarks.

Methodology

The proposed approach relies heavily on the concept of time-space diagrams. This diagram represents the trajectory of every vehicle traversing a roadway segment over a particular time period. Accordingly, this diagram can provide accurate estimation of the traffic state based on individual vehicle movements. Moreover, traffic shockwavese can be captured as the boundary between different traffic states. Considering all the invaluable insights that can be collected from the time-space diagram to understand traffic flow dyanmics, this study utilizes this concept as the input to the proposed methodology and predicts the propagation of shockwaves using a time-space diagram.

Time-Space Diagram

The construction of the time-space diagram follows the concept proposed by Khajeh Hosseini and Talebpour (2019). Accordingly, assuming a connected driving environment, the time-space diagram is constructed based on the information received from the basic safety message (BSM) (SAE 2016) (i.e., location and speed of every vehicle in the segment every 0.1 s). The time-space diagram can be generated in the form of a time-space matrix, as proposed by Khajeh Hosseini and Talebpour (2019). The time-space matrix (Fig. 1) approximates the time-space diagram by dividing the time and space domains into bins of 3.048 m (10 ft) by 100 ms (discretization).

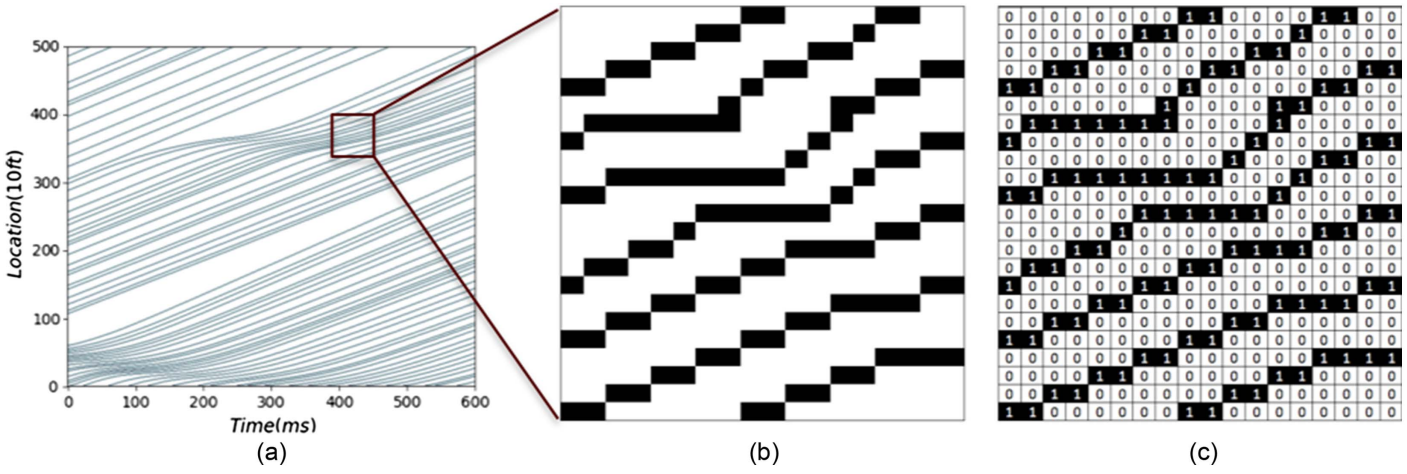

Fig. 1. Time-space diagram: (a) time-space diagram; (b) time-space bins; and (c) time-space matrix. [M. Khajeh Hosseini and A. Talebpour, "Traffic Prediction using Time-Space Diagram: A Convolutional Neural Network Approach." *Transportation Research Record* 2673 (7): 425–435, © 2019 by SAGE, reprinted by permission of SAGE Publications, Ltd.]

J. Transp. Eng., Part A: Systems, 2023, 149(4): 04023019

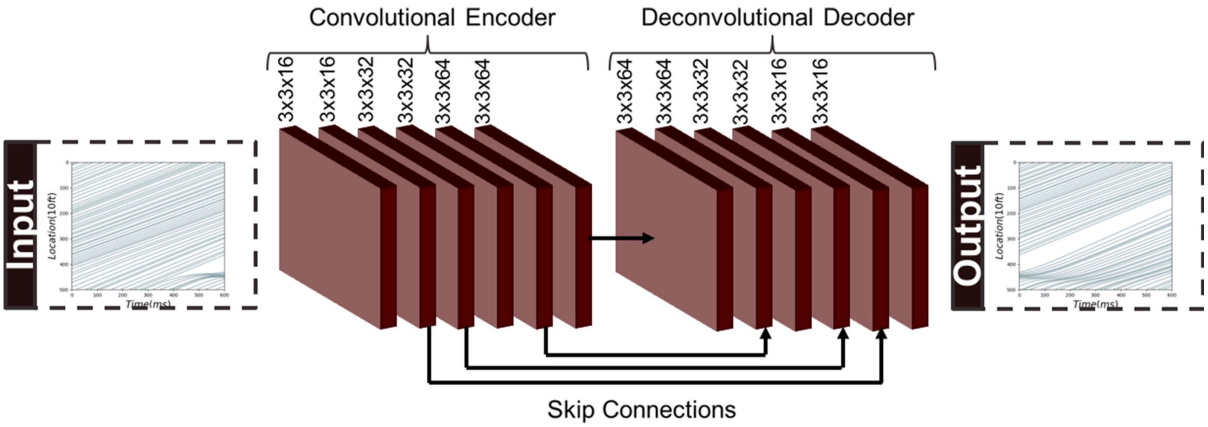

Fig. 2. Shockwave prediction: a convolutional encoder–decoder approach.

The time-space matrix is a binary matrix where the rows represent the discrete space domain and the columns represent the discrete time domain. In this matrix, a cell value of one indicates the presence of a vehicle in that space and time bin, and the value of zero indicates an empty bin. Such a binary representation of the time-space diagram results in a two-dimensional (2D) tensor that can be directly utilized in the convolution process.

Convolutional Encoder–Decoder

This study proposes the use of a deep neural network to predict the propagation of the traffic shockwave from the current time-space diagram of the vehicles, as shown in Fig. 2. The convolutional encoder–decoder structure is an appealing type of mapping function for this study because the input and output of the networks are 2D tensors with similar properties. The convolution is the process of sliding a fixed-size filter (e.g., a three-dimensional receptive window) over the input tensor. Each convolutional layer applies the convolution process to the output of the previous layer and provides an output tensor. The convolution process accounts for the spatial correlation between the units that fit in the receptive window of the filter. Also, sliding the same filter over the input space ensures feature detection independent of its location. This location independence aspect of the convolutional layers makes them a practical choice to encode the time-space matrix because traffic shockwaves can occur at any point in time and space.

There are different convolutional encoder–decoder network architectures depending on the use of convolutional, deconvolutional, pooling, and upsampling layers. Some networks only use convolutional layers in both encoder and decoder components, such as the fully convolutional network (FCN) (Long et al. 2015) and Seg-Net (Badrinarayanan et al. 2017). Other networks such as DeconvNet (Noh et al. 2015) and RED-Net (Mao et al. 2016) use deconvolutional layers in the decoder component. Some of the challenges in the convolutional encoder–decoder networks are the vanishing gradient and reconstructing lost features from the max-pooling and convolution process. The use of skip connections (Mao et al. 2016) and memorizing the maximum features of the pooling process to use for the upsampling (Noh et al. 2015) are the solutions. The skip connections were inspired by the residual network (ResNet) (He et al. 2016) and allow the signal to be propagated to the bottom layers and address the vanishing gradient.

The proposed encoder–decoder architecture of Fig. 2 was inspired by RED-Net (Mao et al. 2016) developed for image restoration. It consists of symmetric layers of convolution and deconvolution with skip-layer connections. The encoder component of the network contains three pairs of convolutional layers with a small receptive window of 3×3 and increasing channels from 16 to 64. Using pairs of convolutional layers with small receptive windows was motivated by the work of Simonyan and Zisserman (2014). A stack of smaller filter size convolutional layers not only can provide the same overall receptive window as one layer with a large window size but also introduce more nonlinearity to the model and reduce the number of trainable parameters.

The decoder component of the network contains pairs of deconvolutional layers with a window size of 3×3 and a number of channels comparable to the convolutional layers, including 64, 32, and 16, creating a symmetrical network architecture. The deconvolution process associates a single input with multiple outputs, unlike the convolution process. Moreover, the rectified linear activation function (ReLU) was adopted as the activation function for both the convolutional and deconvolutional layers. In addition, both the convolutional and deconvolutional layers were used with a stride of one and zero padding to preserve the original input (time-space matrix) size for the output 2D tensor.

The skip layers connect the symmetric convolution and deconvolution layers every two layers. The skip-layer connection sums the convolutional feature map with the deconvolutional feature map elementwise and passes it to the next layer. The encoding convolutional layers focus on extracting the key features that can determine the propagation of shockwaves and traffic flow dynamics, and the deconvolution layers utilize the extracted information to predict the propagation of shockwaves in the traffic stream. Due to the deep nature of the proposed network, the training process can suffer from the the vanishing gradient problem. Accordingly, the skip-layer connections were introduced to allow the propagation of the gradient to the beginning layers of the network. Another key feature of this network is its ability to accept time-space matrices with various sizes as input. This is possible becaduse the network only utilizes the convolutional and deconvolutional layers.

The presented approach captures the interaction between the vehicles by adopting the time-space diagram as an input to the prediction model. The vehicles interact with their surrounding vehicles and the traffic environment by adjusting their speed, spacing, and lane-changing maneuvers. These interactions are embedded in the trajectory of the vehicles in the time-space diagram. The proposed deep learning model takes the averaged time-space matrix as the input, and the model learns the features embedded in the averaged time-space matrix directly from the data without the need for

manual feature extraction. The encoder component encodes the features embedded in the time-space diagram. The decoder component predicts the propagation of the traffic shockwaves in the form of a new time-space matrix.

Data

Simulation

The proposed approach requires a comprehensive and extensive data set for training. The training data set should cover a wide range of traffic shockwaves' formation and propagation. Unfortunately, there are limited vehicle trajectory data sets to create an extensive data set for the training of the proposed deep neural network that can generalize well. As a result, to construct a comprehensive and large data set, this study used a microscopic simulator written in the Python programming language to collect the trajectory of the vehicles. The microscopic simulator adopted the Intelligent Driver Model (IDM) (Treiber et al. 2000) as its car-following logic and minimizing overall braking induced by lane changes (MOBIL) (Kesting et al. 2007) as its lane-changing logic.

A 12,192-m (40,000-ft) three-lane highway was simulated over 15 min. Multiple runs of the simulation were conducted with unique and random IMD and MOBIL parameters to introduce various behaviors to the simulation. In addition, in order to create a data set with different traffic states (from free flow to fully congested), two types of disturbances were used in the simulation. Sudden deceleration of a random vehicle for a small period (e.g., 15 s) createed a speed drop perturbation and disturbed the traffic stream. Another type of disturbance used in the simulation was forcing a random slow-moving vehicle for a more extended period, such as 5 min, to create congestion and traffic breakdown. Both of the disturbances resulted in the formation of shockwaves in the traffic stream.

Additionaly, to introduce further randomness to the simulation, the start time of these two disturbances were selected randomly from the periods of $(20i, 20i + 20)$ s, where i denotes even numbers between zero and 45. This assumption is necessary to exclude the start of these disturbances from the output data, when constructing pairs of input and output data for the training of the model. This is necessary because the model cannot predict random occurrences of these disturbances. Moreover, the desired speed was randomly (uniform distribution) selected from each run from the following set: 48 km/h (30 mi/h), 70 km/h (45 mi/h), 80 km/h (50 mi/h), 90 km/h (55 mi/h), 105 km/h (65 mi/h), 110 km/h (70 mi/h), and 120 km/h (75 mi/h), to create a more comprehensive data set.

Input and Output Data

The proposed encoder–decoder (Fig. 2) of this study takes the time-space matrix as input and predicts the propagation of the traffic shockwaves in the same time-space matrix form. The binary time-space matrix (Fig. 1) not only presents the traffic shockwaves but also depicts the crisp location of the individual vehicles in time and space domains. Training the network to output a binary tensor of shape $(200, 200)$ is challenging, and breaking the binary constraint improves the training. Averaging the cells of the time-space matrix with their neighbors [Fig. 3(a)] blurs the exact location of the vehicles on time and space domains; however, averaging over a small window maintains the propagation of traffic shockwaves [Fig. 3(b)]. The points on the averaged time-space diagram [Fig. 3(b)] change proportional to the value of the cell ranging from zero to one.

Taking the averaged time-space matrix as the type of output improves the training of the network. At every time t, the encoder–decoder model takes as an input an averaged time-space matrix of the last 20 s $(t - 20, t)$ and predicts the averaged time-space matrix for the future 20 s $(t, t + 20)$. The averaged time-space matrix ($\bar{\mathbf{TS}}$) is created by replacing every cell of the binary time-space matrix ($\mathbf{TS}$) with the average of itself and all the neighboring cells within a specified range on both row (space) and column (time) domains similar to the concept of rolling average filter in image processing based on the following equation:

$$\bar{\mathbf{TS}}(r, c) = \frac{1}{(2m + 1)(2n + 1)} \sum_{i=-m}^{m} \sum_{j=-n}^{n} \mathbf{TS}(r + i, c + j) \quad (1)$$

where $\bar{\mathbf{TS}}(r, c)$ = value of the cell at row r and column c of the averaged time-space matrix; and variables m and n = size of the averaging window along the row and column domains, respectively.

In this study, the time-space matrix [Fig. 3(a)] approximates the time-space diagram by dividing the time and space domains into cells of 3.048 m (10 ft) by 0.1 s. The averaged time-space matrix is created by replacing the value of each cell in the binary time-space matrix with the average of itself and its neighbors

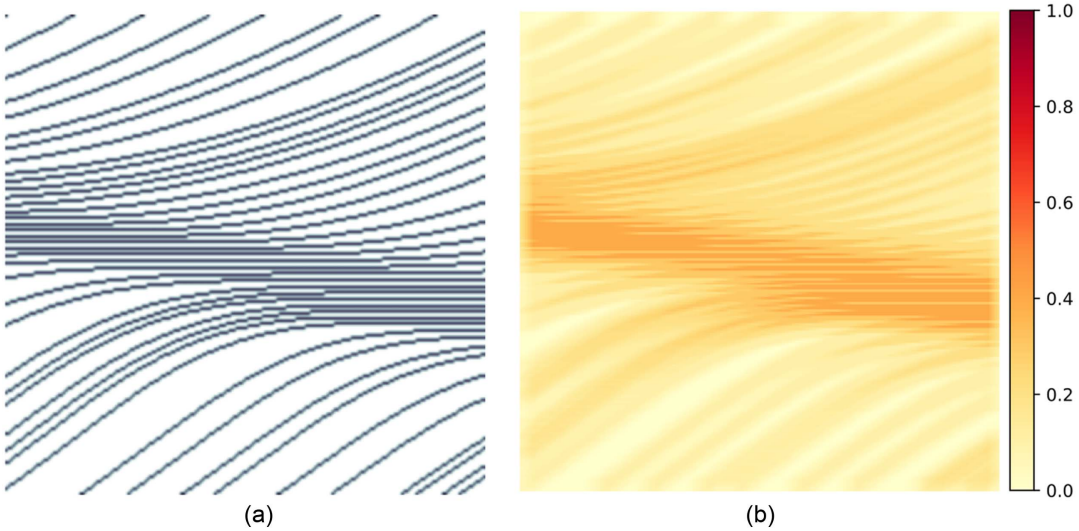

Fig. 3. Averaged time-space diagram: (a) time-space diagram; and (b) averaged time-space diagram.

J. Transp. Eng., Part A: Systems, 2023, 149(4): 04023019

[Eq. (1)] up to 15.24 m (50 ft) ($m = 5$ rows) and 0.5 s ($n = 5$ columns) on each side [i.e., averaging window of 30.48 m (100 ft) by 1 s]. The encoder–decoder network approximates the mapping function from the averaged time-space matrix of segment x over the period of $(t - 20, t)$ to the averaged time-space matrix of the segment x over the period of $(t, t + 20)$.

Training Data

The microscopic traffic simulator provides a 12,192 m (40,000 ft) by 900 s time-space diagram for each lane and every simulation run. This diagram can be divided into 900 smaller time-space diagrams for segments of 609.6 m (2,000 ft) and a shorter period of 20 s. The 900 smaller time-space diagrams are divided into 450 pairs of input and output data. One could extract more pairs of input and output if they relax the limitation on keeping the start of the artificial disturbances in the input data. This study collected data from more than 2,000 simulation runs resulting in more than 0.9 million data points. The collected data were divided into three groups of training, validation, and testing sets with ratios of 80%, 10%, and 10%, respectively.

Training

Training is the iterative process of adjusting the trainable parameters of the model to gradually minimize the loss function. The convolutional encoder–decoder of this study contained 180,449 trainable parameters. Adopting the small receptive window of 3×3 as well as fully convolutional and deconvolutional layers kept the number of network parameters relatively small. The model parameters were updated in multiple iterations (steps). At each iteration, the loss function was estimated for a batch of data points, and the parameters were adjusted based on their loss gradient times the learning rate (a small constant). The Adam optimizer (Kingma and Ba 2014) is a stochastic gradient–based optimizer that was adapted for the training of the network of this study.

Loss Function

The prediction model of this study is a regression model that maps the current averaged time-space diagram to the future averaged time-space diagram. The mean squared error (MSE) [Eq. (2)] is a standard performance measure used as the loss function for the training of regression-type neural networks. The output of network (model) F with parameters Θ for input X^i is $F(X^i; \Theta)$, and the true output value is Y^i. Mean absolute error (MAE) [Eq. (3)] is another performance measure for regression problems. Although the MAE is not useful as the loss function and estimation of gradients in neural networks, it is used for evaluating the performance of the proposed network

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n} \|F(X^i; \Theta) - Y^i\|^2 \tag{2}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} \|F(X^i; \Theta) - Y^i\| \tag{3}$$

The input and output of the model of this study are 2D tensors of size $(200, 200)$. A smoothed version of the output can be constructed by replacing each cell of the output tensor with the average of itself and its neighboring cells [Eq. (1)]. A well-trained neural network model is expected to predict outputs comparable to the true outputs. Besides, it is expected that the smoothed versions of the predicted and true outputs are also comparable. In order to speed up the training (convergence) of the model and to guide the gradients, this study proposes the use of the following custom loss function:

$$\text{Loss} = \text{MSE} + 1,000(\text{MSE}_{10} + \text{MSE}_5 + \text{MSE}_3) \tag{4}$$

where MSE_{10}, MSE_5, and MSE_3 = estimated MSE between the true and predicted outputs when smoothed with sliding average windows of size 10×10, 3×3, and 5×5, respectively. The sliding window size indicates the extent of neighboring cells considered in the estimation of the average for that cell. Adopting this custom loss function improved the convergence of the training process significantly.

The training process of the model was conducted in two steps. In the first step, the model was trained using the loss function of Eq. (4) until the loss value on the validation set starts increasing. In the second step, the model was retrained using MSE [Eq. (2)] as the loss function.

Calibration Results

Deep neural networks are prone to overfitting due to the significant number of trainable parameters. In the case of training a deep learning model using a variant of stochastic gradient descent (e.g., Adam optimizer in this study), batch size and the number of training epochs are among the hyperparameters that could impact the generalization capability (good performance on the testing data set) of the trained model. A smaller batch size is expected to offer a regularization effect and lower generalization error due to the noise added to the learning process when using a smaller batch size (Goodfellow et al. 2016). Each training epoch is a complete iteration over the entire training data set using stochastic gradient descent. With the increase in the number of training epochs, the model's error on the training data set reduces; however, continuous training beyond an optimal number of epochs can result in overfitting the training data set and lower generalization performance. This study adopted a policy known as early stopping to ensure that the model had trained for a sufficiently large number of training epochs, not underfitting (too little training) and overfitting (too much training). The early stopping policy stops the training when a monitored metric (e.g., validation loss) has stopped improving for a certain number of training epochs (Goodfellow et al. 2016; Zhang et al. 2021).

In this study, the training was stopped when the loss function on the validation data set was not improved after five consecutive training epochs. In the training process of the model, a batch size of 60 and the early stopping policy were used to prevent overfitting the training data. The loss function was estimated at the end of every epoch (a complete iteration over the entire data set), and the training was stopped after five epochs from the one with the minimum loss on the validation data set. Tables 1 and 2 present the prediction ability of the proposed network on the validation and testing data sets.

Table 1. Model performance on time-space matrix on validation data set

Model	MSE	$\text{MSE}_{10} + \text{MSE}_5 + \text{MSE}_3$
Model after training Step 1	0.0037	0.0071
Model after training Step 2	0.0029	N/A

Table 2. Model performance on time-space matrix on testing data set

Model	MSE	MAE
Fully trained model	0.0030	0.0408

Training the model with the custom loss function [Eq. (4)] helped the convergence in the first step of training. Also, retraining the model in the second step by adopting the original MSE as the loss function further improved the performance of the model from the MSE error of 0.0037 to 0.0029. According to Tables 1 and 2, the performance of the fully trained model in terms of MSE and MAE on the testing data set was 0.0030 and 0.0408, respectively.

Results and Discussion

Traffic Shockwave Propagation Prediction

Fig. 4 presents the results of shockwave prediction using the proposed approach. As discussed previously, the prediction model received an averaged time-space matrix covering $(t-20, t)$ as input (x), and predicts the future averaged time-space matrix for $(t, t+20)$ (y). The results indicate the model's capability in predicting various traffic flow states and shockwave propagation. The predicted averaged time-space diagrams present dissemination, propagation, and forward and backward movements of the traffic shockwaves over the evaluated segment of the roadway.

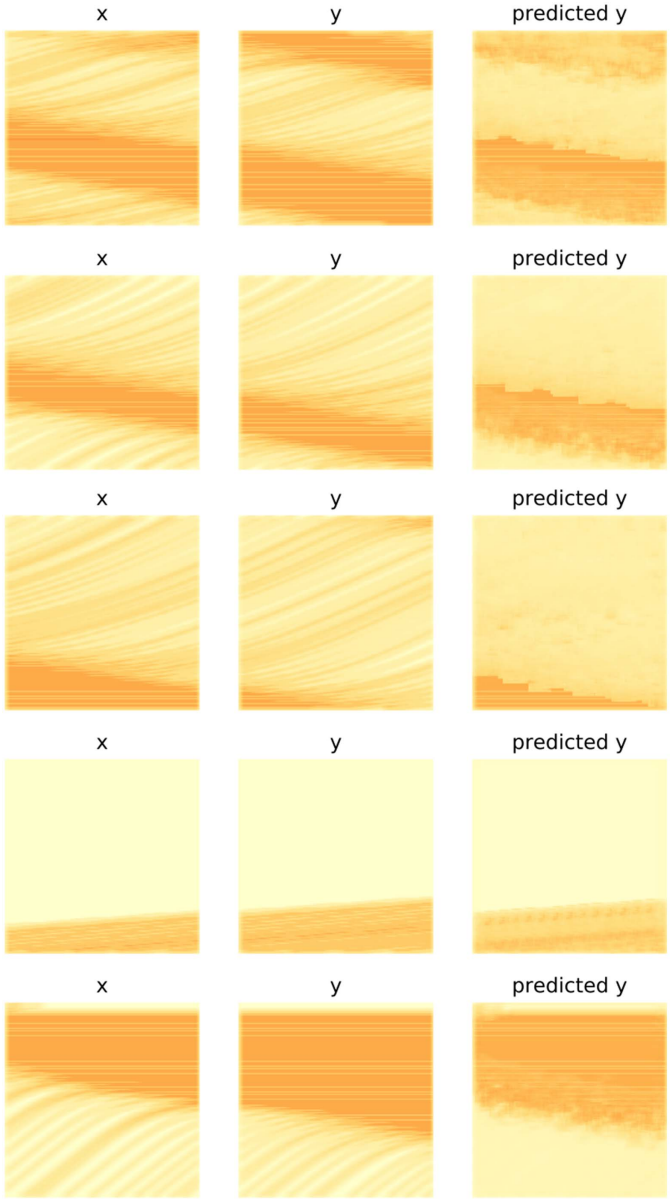

Fig. 4. Traffic shockwave propagation prediction results.

Density Time-Space Matrix

The traffic shockwave propagation prediction of the network can be evaluated more quantitatively. A traffic shockwave is the boundary between two states of traffic. Edie (1961) estimated the average density $k(A)$ for a time-space block of A [e.g., 30.48 m (100 ft) by 1 s] based on the following equation:

$$k(A) = \frac{t(A)}{|A|} \tag{5}$$

where $|A|$ = area of the time-space block A; and $t(A)$ = total time spent by all the vehicles going through block A. As specified in the "Methodology" section, the time-space matrix is a binary matrix constructed by dividing time and space domains into bins of 3.048 m (10 ft) by 100 ms. In this matrix, one represents the presence of a vehicle in that time and space bin, and zero represents an empty bin. The number of occupied bins of the time-space block A is equal to the summation of all the bins of its representative binary time-space matrix [i.e., $\text{sum}(A)$]. As a result, the total time spent by all the vehicles going through any arbitrary time-space block of A is equal to multiplying the number of occupied bins in that block by 0.1 s

$$t(A) = 0.1 \times \text{sum}(A) \tag{6}$$

Considering Edie's (1961) definition of the average density of a time-space block, the averaged time-space matrix ($\bar{\mathbf{TS}}$) can be used to estimate the density time-space matrix ($\mathbf{K}$). Similar to the time-space matrix, the rows of this matrix represent the discrete space domain, and their columns represent the discrete time domain. The values of each cell in the matrix $\mathbf{K}$ is the average density of a time-space block [e.g., 30.48 m (100 ft) by 1 s] centered at that location in time and space. The density time-space matrix depicts the change in traffic state over time and space and, consequently, the propagation of the traffic shockwaves.

As mentioned in the "Methodology" section, the averaged time-space matrix ($\bar{\mathbf{TS}}$) of this study is estimated by replacing every cell in the time-space matrix with the average of itself and its neighbors [Eq. (1)] up to 30.48 m (100 ft) and 1 s. Each cell in the time-space matrix is representative of a bin with dimensions of 3.048 m (10 ft) in the space domain and 0.1 s in the time domain. The averaging window of 30.48 m (100 ft) by 1s is equivalent to a 10×10 averaging window on the time-space matrix. In other words, each cell of the averaged time-space matrix ($\bar{\mathbf{TS}}$) is the average of 100 cells in the time-space matrix. Therefore, if the averaged time-space matrix ($\bar{\mathbf{TS}}$) is multiplied by 100, the cells of the resulting matrix indicate the number of occupied cells in the blocks of 30.48 m (100 ft) by 1 s centered on that location on the time-space matrix. As a result, the density time-space matrix ($\mathbf{K}$) can be estimated based on the following equation:

$$\mathbf{K} = \frac{\mathbf{t}(A)}{|A|} = \frac{100 \times \bar{\mathbf{TS}} \times 0.1}{100 \times 1} \times 3,280.84 = 528 \times \bar{\mathbf{TS}} \tag{7}$$

where the constant 3,280.84 is applied for the unit conversion from feet to miles. Vehicles per mile (vpm) is the unit for the values in the resulting density time-space matrix ($\mathbf{K}$). According to Eq. (7), the

Table 3. Network performance on density time-space matrix on validation data set

Model	MSE	$MSE_{10}+MSE_5+MSE_3$
Model after training Step 1	1,031.50	1,979.36
Model after training Step 2	808.47	NA

Table 4. Network performance on density time-space matrix on testing data set

Model	MSE	MAE
Fully trained model	836.35	21.54

averaged time-space matrix ($\bar{\mathbf{TS}}$) can be converted to the density time-space matrix ($\mathbf{K}$) by a constant scalar. Therefore, the prediction (output) of the model is proportional to the density time-space matrix. The performance of the model in Tables 1 and 2 is updated for the density time-space matrix presented in Tables 3 and 4. According to Tables 3 and 4, the mean absolute error of the model in predicting the density for small blocks of 30.48 m (100 ft) by 1 s is 13.38 vehicles per km.

Root-mean squared error (RMSE) is another valuable performance measure that has the same unit as the output. Based on Tables 3 and 4, the RMSE of the model in the prediction of the density on the testing data set was estimated as 17.96 vehicles per km. Considering a range of 125 vehicles per km for the density, the MAE and RMSE of the model are between 10% to 14% of the density range.

Validating Model on the NGSIM Data Set

The proposed encoder–decoder model in this study was also evaluated using a real-world vehicle trajectory data set. Some of the existing real-world vehicle trajectory data sets are the Federal Highway Administration (FHWA) NGSIM (FHWA 2007), Strategic Highway Research Program (SHRP2) (Hankey et al. 2016), TrafficNet (Zhao et al. 2017), HighD data set (Krajewski et al. 2018), pNEUMA data set (Barmpounakis and Geroliminis 2020), and high-granularity highway simulation (HIGH-Sim) (Shi et al. 2021). This study utilizes the NGSIM data set for the evaluation of the proposed model. NGSIM is a well-known open-source trajectory data set collected in 2006 using digital cameras at different locations, including US Highway 101 and the Interstate 80 freeway. This data set covers three 15-min periods during the morning peak.

The time-space diagram for each lane was recreated from the trajectory data for the middle 609.6 m (2,000 ft) of the segment every 20 s. The NGSIM data sum up to 660 data points for five lanes and were directly used to test the previously trained encoder–decoder model. The performance of the model on this data set is reported in Table 5 in the form of MSE and MAE for both the time-space matrix and the density time-space matrix. The MSE and MAE values for the NGSIM data set were slightly higher than the respective ones for the testing data set. This difference was expected because the real-world collected data are different from simulated data.

According to Table 5, the MAE and RMSE of the model in predicting the density of small blocks of 30.48 m (100 ft) by 1 s are 16.57 and 21.76 vehicles per km, respectively. The

Table 5. Network performance on the NGSIM data set

Matrix	MSE	MAE
Time-space matrix	0.0044	0.0505
Density time-space matrix	1,226.64	26.66

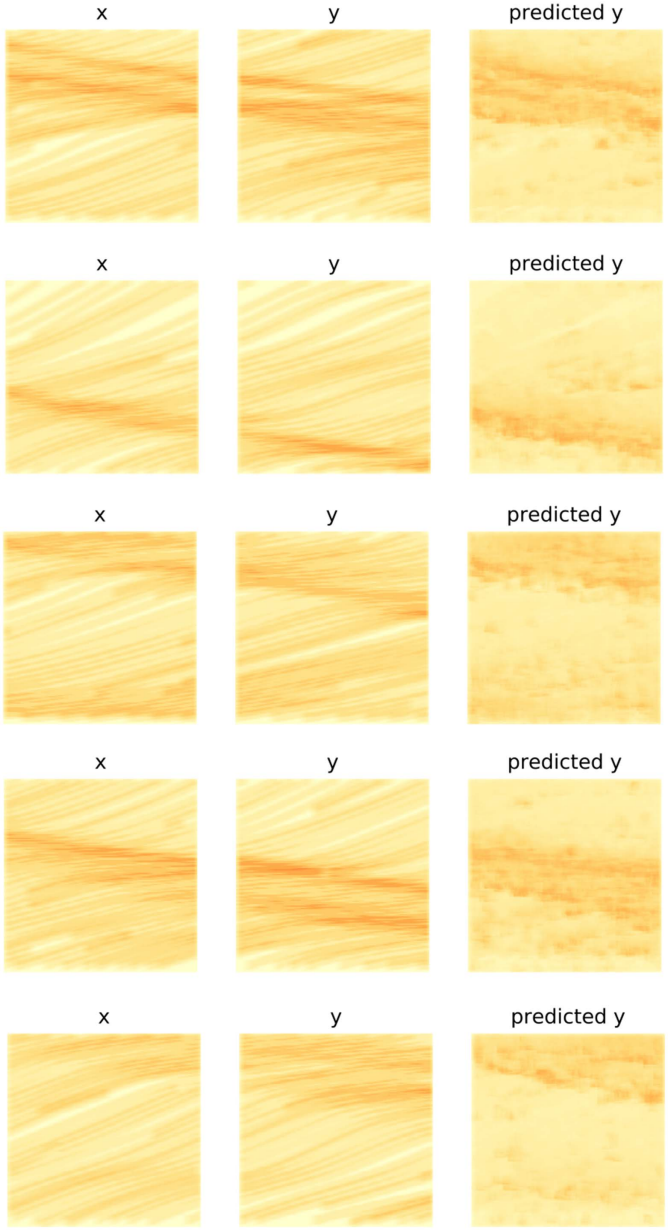

Fig. 5. Traffic shockwave propagation prediction results on the NGSIM data set.

MAE and RMSE values of the trained model on the NGSIM data set are less than 15% of the range of density. Moreover, Fig. 5 compares the model predictions (predicted y) and the true states of the traffic (y) for a few input examples from the NGSIM data set. According to this figure, the model is capable of predicting the propagation of traffic shockwaves on the real-world collected data. It should be noted that the model's performance can further improve by training the model with some of the collected data from NGSIM.

Conclusion

Traffic state prediction has been historically performed at the aggregated level, whereas predicting the driving environment at the individual level has been studied mostly for automated vehicle motion planning (to plan a safe path and avoid collision with other traffic agents and obstacles). This study aimed to utilize the data from connected automated vehicles to link these two approaches and increase the accuracy of traffic state prediction. Although there have been several studies that utilized connected vehicle data for more accurate traffic prediction, the majority of them still followed the same standard aggregate-based prediction approaches (although with more comprehensive and accurate data from connected vehicles). This study aimed to address the existing challenges in this area and improve traffic prediction by capturing the interactions among vehicles.

Accordingly, this study proposed a methodology to predict the propagation of traffic shockwaves in the form of the averaged time-space matrix. The averaged time-space matrix is comparable to a density time-space matrix derived from Edie's (1961) definition of average density. The traffic shockwave is the boundary between two traffic states, and the density time-space matrix depicts the state changes in the form of density.

The result of the analysis indicated that the model is capable of predicting the dissemination, propagation, and forward and backward movements of traffic shockwaves over the study segment. The model cannot predict random braking or slowing down by a single driver, causing disturbances in the traffic stream. However, upon the observance of the disturbances in the input time-space matrix, the model can predict shockwave propagation. Moreover, the performance of the model in the form of MAE and RMSE for predicting the density time-space matrix was 13.38 and 17.96 vehicles per km for the simulated testing data set and 16.57 and 21.76 vehicles per km for the NGSIM data set. Considering a range of 0–125 vehicles per km for the density, the performance of the model is acceptable for the prediction of the traffic shockwaves propagation. It is noteworthy that the prediction accuracy based on the propose approach highly depends on the quality of the training data. Utilizing a more balanced data set that covers more edge cases can increase the prediction accuracy.

The traffic state prediction proposed in this study was based on the assumption of complete knowledge of the observed trajectory of all the vehicles on the study area considering a fully connected environment or from the sensory data of connected and automated vehicles. In practice, the future traffic stream could be a mix of conventional, connected, and connected automated vehicles. Although it is feasible to capture and monitor the majority of the vehicles in the traffic stream based on the sensory data from the connected and automated vehicles when their market penetration rate is above a minimum level, Talebpour et al. (2016) showed that due to signal interference, many information packets would not reach their destinations, even in a fully connected driving environment. Accordingly, it is critical to investigate and improve the proposed model in this study to predict the traffic state based on partial or incomplete data from the traffic stream in future studies.

In addition to the aforementioned limitation, the driving environment evolves as a result of interactions among the individual vehicles. These interactions can be defined as a combination of lateral and longitudinal maneuvers of vehicles in response to their driving environment. At the individual-vehicle-level prediction, structuring the driving task into different maneuvers and considering the vehicles' interactions can improve the vehicle trajectory prediction accuracy. As a result of a vehicle's maneuver, its future trajectory (e.g., its location and speed) changes. In many driving scenarios, more than one maneuver is feasible that could result in different time-space diagrams. Therefore, it would be more realistic to predict the location of the vehicles in a probabilistic manner based on the different maneuvers in future studies.

Finally, the prediction accuracy usually decays with the increase in the prediction horizon due to the uncertainty in drivers' behavior and an increase in the possibility of various configurations and outcomes. The prediction horizon is dependent on the planning horizon, and in general, a longer prediction horizon is preferred for congestion mitigation methods. In future research studies, it would be valuable to investigate the trade-off between prediction accuracy and the prediction horizon, considering the effectiveness of the congestion mitigation methodologies.

Data Availability Statement

Some or all data, models, or codes that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant Nos. 1826410 and 2047937.

References

Akhtar, M., and S. Moridpour. 2021. "A review of traffic congestion prediction using artificial intelligence." *J. Adv. Transp.* 2021 (Jan): 1–18. https://doi.org/10.1155/2021/8878011.

Badrinarayanan, V., A. Kendall, and R. Cipolla. 2017. "Segnet: A deep convolutional encoder-decoder architecture for image segmentation." *IEEE Trans. Pattern Anal. Mach. Intell.* 39 (12): 2481–2495. https://doi.org/10.1109/TPAMI.2016.2644615.

Barmpounakis, E., and N. Geroliminis. 2020. "On the new era of urban traffic monitoring with massive drone data: The pneuma large-scale field experiment." *Transp. Res. Part C Emerging Technol.* 111 (Feb): 50–71. https://doi.org/10.1016/j.trc.2019.11.023.

Bautista-Camino, P., A. I. Barranco-Gutiérrez, I. Cervantes, M. Rodrguez-Licea, J. Prado-Olivarez, and F. J. Pérez-Pinal. 2022. "Local path planning for autonomous vehicles based on the natural behavior of the biological action-perception motion." *Energies* 15 (5): 1769. https://doi.org/10.3390/en15051769.

Bogaerts, T., A. D. Masegosa, J. S. Angarita-Zapata, E. Onieva, and P. Hellinckx. 2020. "A graph cnn-lstm neural network for short and long-term traffic forecasting based on trajectory data." *Transp. Res. Part C Emerging Technol.* 112 (Mar): 62–77. https://doi.org/10.1016/j.trc.2020.01.010.

Castro-Neto, M., Y.-S. Jeong, M.-K. Jeong, and L. D. Han. 2009. "Online-svr for short-term traffic flow prediction under typical and atypical traffic conditions." *Expert Syst. Appl.* 36 (3): 6164–6173. https://doi.org/10.1016/j.eswa.2008.07.069.

Chen, R., W. Jin, and A. Regan. 2010. "Multi-hop broadcasting in vehicular ad hoc networks with shockwave traffic." In *Proc., 7th IEEE Consumer Communications and Networking Conf.*, 1–5. New York: IEEE.

Claussmann, L., M. Revilloud, D. Gruyer, and S. Glaser. 2019. "A review of motion planning for highway autonomous driving." *IEEE Trans. Intell. Transp. Syst.* 21 (5): 1826–1848. https://doi.org/10.1109/TITS.2019.2913998.

Edie, L. C. 1961. "Car-following and steady-state theory for noncongested traffic." *Oper. Res.* 9 (1): 66–76. https://doi.org/10.1287/opre.9.1.66.

Eglese, R., W. Maden, and A. Slater. 2006. "A road timetabletm to aid vehicle routing and scheduling." *Comput. Oper. Res.* 33 (12): 3508–3519. https://doi.org/10.1016/j.cor.2005.03.029.

Elfar, A., C. Xavier, A. Talebpour, and H. S. Mahmassani. 2018. "Traffic shockwave detection in a connected environment using the speed distribution of individual vehicles." *Transp. Res. Rec.* 2672 (20): 203–214. https://doi.org/10.1177/0361198118794717.

FHWA (Federal Highway Administration). 2007. *Next generation simulation: US101 freeway dataset*. Washington, DC: FHWA.

Gaweesh, S. M., A. Khoda Bakhshi, and M. M. Ahmed. 2021. "Safety performance assessment of connected vehicles in mitigating the risk of secondary crashes: A driving simulator study." *Transp. Res. Rec.* 2675 (12): 117–129. https://doi.org/10.1177/03611981211027881.

Goodfellow, I., Y. Bengio, and A. Courville. 2016. *Deep learning*. New York: MIT Press.

Hankey, J. M., M. A. Perez, and J. A. McClafferty. 2016. *Description of the SHRP 2 naturalistic database and the crash, near-crash, and baseline data sets*. Rep. No. Blacksburg, VA: Virginia Tech Transportation Institute.

He, K., X. Zhang, S. Ren, and J. Sun. 2016. "Deep residual learning for image recognition." In *Proc., IEEE Conf. on Computer Vision and Pattern Recognition*, 770–778. New York: IEEE.

Ke, R., W. Li, Z. Cui, and Y. Wang. 2020. "Two-stream multi-channel convolutional neural network for multi-lane traffic speed prediction considering traffic volume impact." *Transp. Res. Rec.* 2674 (4): 459–470. https://doi.org/10.1177/0361198120911052.

Kesting, A., M. Treiber, and D. Helbing. 2007. "General lane-changing model mobil for car-following models." *Transp. Res. Rec.* 1999 (1): 86–94. https://doi.org/10.3141/1999-10.

Khajeh Hosseini, M., and A. Talebpour. 2019. "Traffic prediction using time-space diagram: A convolutional1neural network approach." *Transp. Res. Board* 2673 (7): 425–435. https://doi.org/10.1177/0361198119841291.

Kingma, D. P., and J. Ba. 2014. "Adam: A method for stochastic optimization." Preprint, submitted July 5, 2022. http://arxiv.org/abs/1412.6980.

Krajewski, R., J. Bock, L. Kloeker, and L. Eckstein. 2018. "The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems." In *Proc., IEEE 21st Int. Conf. on Intelligent Transportation Systems (ITSC)*. New York: IEEE.

Kumar, S. V., and L. Vanajakshi. 2015. "Short-term traffic flow prediction using seasonal arima model with limited input data." *Eur. Transp. Res. Rev.* 7 (3): 21. https://doi.org/10.1007/s12544-015-0170-8.

Lee, C., Y. Kim, S. Jin, D. Kim, R. Maciejewski, D. Ebert, and S. Ko. 2019. "A visual analytics system for exploring, monitoring, and forecasting road traffic congestion." *IEEE Trans. Vis. Comput. Graphics* 26 (11): 3133–3146. https://doi.org/10.1109/TVCG.2019.2922597.

Lefèvre, S., D. Vasquez, and C. Laugier. 2014. "A survey on motion prediction and risk assessment for intelligent vehicles." *ROBOMECH J.* 1 (1): 1–14. https://doi.org/10.1186/s40648-014-0001-z.

Long, J., E. Shelhamer, and T. Darrell. 2015. "Fully convolutional networks for semantic segmentation." In *Proc., IEEE Conf. on Computer Vision and Pattern Recognition*, 3431–3440. New York: IEEE.

Mao, X., C. Shen, and Y.-B. Yang. 2016. "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections." In *Advances in neural information processing systems*, 2802–2810. Cambridge, MA: MIT Press.

Noh, H., S. Hong, and B. Han. 2015. "Learning deconvolution network for semantic segmentation." In *Proc., IEEE Int. Conf. on Computer Vision*, 1520–1528. New York: IEEE.

Pendleton, S. D., H. Andersen, X. Du, X. Shen, M. Meghjani, Y. H. Eng, D. Rus, and M. H. Ang. 2017. "Perception, planning, control, and coordination for autonomous vehicles." *Machines* 5 (1): 6. https://doi.org/10.3390/machines5010006.

Polson, N. G., and V. O. Sokolov. 2017. "Deep learning for short-term traffic flow prediction." *Transp. Res. Part C Emerging Technol.* 79 (Jun): 1–17. https://doi.org/10.1016/j.trc.2017.02.024.

Rahman, R., J. Ben-Edigbe, and A. Hassan. 2012. "Extent of traffic kinematic waves and queuing caused by midblock u-turn facilities." In *Proc., ITRN2012*, 8. Seattle, WA: Semantic Scholar.

Razali, N. A. M., N. Shamsaimon, K. K. Ishak, S. Ramli, M. F. M. Amran, and S. Sukardi. 2021. "Gap, techniques and evaluation: Traffic flow prediction using machine learning and deep learning." *J. Big Data* 8 (1): 1–25. https://doi.org/10.1186/s40537-021-00542-7.

SAE (Society of Automotive Engineers). 2016. *J2735 dedicated short range communications (dsrc) message set dictionary*. Troy, MI: SAE.

Seo, T., A. M. Bayen, T. Kusakabe, and Y. Asakura. 2017. "Traffic state estimation on highway: A comprehensive survey." *Ann. Rev. Control* 43 (Jan): 128–151. https://doi.org/10.1016/j.arcontrol.2017.03.005.

Shi, X., D. Zhao, H. Yao, X. Li, D. K. Hale, and A. Ghiasi. 2021. "Video-based trajectory extraction with deep learning for high-granularity highway simulation (high-SIM)" *Commun. Transp. Res.* 1 (Dec): 100014. https://doi.org/10.1016/j.commtr.2021.100014.

Simonyan, K., and A. Zisserman. 2014. "Very deep convolutional networks for large-scale image recognition." Preprint, submitted September 4, 2014. http://arxiv.org/abs1409.1556.

Talebpour, A., H. S. Mahmassani, and F. E. Bustamante. 2016. "Modeling driver behavior in a connected environment: Integrated microscopic simulation of traffic and mobile wireless telecommunication systems." *Transp. Res. Rec.* 2560 (1): 75–86. https://doi.org/10.3141/2560-09.

Treiber, M., A. Hennecke, and D. Helbing. 2000. "Congested traffic states in empirical observations and microscopic simulations." *Phys. Rev. E* 62 (2): 1805. https://doi.org/10.1103/PhysRevE.62.1805.

Van Lint, J., and C. Van Hinsbergen. 2012. "Short-term traffic and travel time prediction models." *Artif. Intell. Appl. Crit. Transp. Issues* 22 (1): 22–41. https://doi.org/10.1109/ITSC.2017.8317886.

Wilby, M. R., J. J. V. Daz, A. B. Rodrguez Gonz lez, and M. Á. Sotelo. 2014. "Lightweight occupancy estimation on freeways using extended floating car data." *J. Intell. Transp. Syst.* 18 (2): 149–163. https://doi.org/10.1080/15472450.2013.801711.

Zhang, C., S. Bengio, M. Hardt, B. Recht, and O. Vinyals. 2021. "Understanding deep learning (still) requires rethinking generalization." *Commun. ACM* 64 (3): 107–115. https://doi.org/10.1145/3446776.

Zhao, D., Y. Guo, and Y. J. Jia. 2017. "Trafficnet: An open naturalistic driving scenario library." In *Proc., IEEE 20th Int. Conf. on Intelligent Transportation Systems (ITSC)*, 1–8. New York: IEEE.

Zheng, Z., and D. Su. 2016. "Traffic state estimation through compressed sensing and markov random field." *Transp. Res. Part B Methodol.* 91 (Sep): 525–554. https://doi.org/10.1016/j.trb.2016.06.009.

Zhou, Y., A. Ravey, and M.-C. Péra. 2019. "A survey on driving prediction techniques for predictive energy management of plug-in hybrid electric vehicles." *J. Power Sources* 412 (Feb): 480–495. https://doi.org/10.1016/j.jpowsour.2018.11.085.