Hindawi
Journal of Advanced Transportation
Volume 2022, Article ID 6888115, 9 pages
https://doi.org/10.1155/2022/6888115

Research Article

Freeway Traffic Speed Prediction under the Intelligent Driving Environment: A Deep Learning Approach

Chengying Hua and Wei (David) Fan

USDOT Center for Advanced Multimodal Mobility Solutions and Education (CAMMSE), Department of Civil and Environmental Engineering, University of North Carolina, EPIC Building, Room 3366, 9201 University City Boulevard, Charlotte, NC 28223-0001, USA

Correspondence should be addressed to Wei (David) Fan; wfan7@uncc.edu

Received 27 July 2022; Revised 2 September 2022; Accepted 13 October 2022; Published 16 November 2022

Academic Editor: Wen Liu

The intelligent transportation system (ITS) has been proven capable of effectively addressing traffic congestion issues. For vehicles to perform effectively and improve mobility under the intelligent driving environment, real-time prediction of traffic speed is undoubtedly essential. Considering the complex spatiotemporal dependency inherent in traffic data, conventional prediction models encounter many limitations. To improve the prediction performance and investigate the temporal features, this study focuses on emerging deep neural networks (DNNs) using the Caltrans Performance Measurement System (PeMS) data. This research also establishes an intelligent driving environment in the simulation and compares the traditional car-following model with deep learning methods in terms of multiple performance metrics. The results indicate that both supervised learning and unsupervised learning are superior to the simulation-based model on the freeway, and the two deep learning networks are almost identical to one another. Besides, the result reveals that all models have their latent features for different time dimensions under the low traffic loads, transition states, and heavy traffic loads. This is critical in the application of prediction technologies in ITS. The findings can assist transportation researchers and traffic engineers in both traffic operation and management, such as bottleneck identification, platooning control, and route planning.

1. Introduction

The continuous growth in the number of vehicles brings many mobility challenges to the current transportation system, such as traffic congestion and extended commuting time. Benefiting from the development of intelligent transportation systems (ITS) and deployment of artificial intelligence (AI) technology, intelligent vehicles (e.g., connected and autonomous vehicles) are expected to greatly help alleviate traffic congestion. Accompanying this is the real-time prediction of traffic speed issues, which is essential for intelligent vehicles to be fully leveraged. Accurate speed prediction can help efficiently control traffic in advance and short-term forecasting has gained popularity due to its adaptability [1].

An overview of existing literature indicates that traffic prediction tasks have shifted from statistical models to adaptive machine learning (ML) methods [2]. Theoretically, nonparametric ML methods can handle stochastic and nonlinear problems better than parametric methods. In practice, more historical data can be converted into useful information by developing data-driven models with enhanced data storage capacities. However, considering the high-dimensional and spatiotemporal traffic data collected from the sensors, shallow ML techniques may be unsatisfactory in the intelligent driving environment, especially as the forecasting horizon size increases [3]. Deep learning (DL) methods, given their ability to mine deep relationships between data [4], greatly inspire researchers to address time series traffic prediction to achieve improved results [5, 6].

In contrast to human-driven cars, where driving behavior is usually uncertain and can only be estimated via massive data from roadside units (RSUs), the control algorithms under the intelligent driving environment may be predictable [7]. However, the requirement for specialized infrastructure and robust algorithms during execution makes traffic prediction costly in a real intelligent driving environment. Furthermore, the vehicle-road synergy is still in its initial phase, with fewer scenario-based, large-scale tests, and comprehensive frameworks in place [8]. Fortunately, simulation-based methods can solve the aforementioned problems. The Intelligent Driver Model (IDM) is a widely used car-following model, which can be developed and implemented in the simulated environment. It can also forecast the vehicle status in an intelligent collision-free manner and modify its behavior as desired [9].

Given the complex and dynamic spatiotemporal dependency inherent in traffic speed data, which is difficult to solve with traditional prediction methods, this study focuses on undertaking the traffic speed prediction task based on emerging deep neural networks (DNNs) using ground truth data. To model intelligent driving behaviors, this research also establishes a simulation environment. Besides, this study compares different methods in terms of multiple evaluation metrics and reveals temporal features under various traffic loads. The findings can help researchers and traffic engineers improve dynamic traffic management. Platooning control, route planning, and signal optimization are some of the potential applications with improved traffic speed prediction.

The remainder of this study is organized as follows: Section 2 reviews previous literature on traffic prediction. Section 3 introduces two DNNs with supervised and unsupervised learning separately and builds an intelligent driving environment in simulation. Section 4 describes the experimental settings. Section 5 compares the performances of different models. Summary and future research directions are given in Section 6.

2. Literature Review

The traffic prediction is affected by various factors such as horizon size, aggregation rate, algorithms, selected area, and database. Horizon size is referred to as the time span when tasks are conducted. Too large or small values will affect the accuracy and complexity of models [10]. Short-term forecasts of five to ten minutes have been widely used in traffic prediction due to the nearly real-time feature [11]. For aggregation rate, the higher the sampling frequency of recent observations, the lower the error compared to historical data [12]. Concerning algorithms, Kamarianakis and Prastacos [13] illustrated that multivariate algorithms can extract more spatiotemporal information, which outperforms univariate algorithms. Another vital element is the selected area which denotes the region of data collected for prediction, such as freeways or urban arterials. Apart from the periodic nature, the freeway area is simple to implement without signal restrictions [14]. The database represents the dataset being whether real-world or simulated, and the data source such as loop detectors, probe vehicles, and GPS.

Traffic prediction methods can be classified as parametric, nonparametric, and simulation. Parametric (model-based) methods have definite parameters and are based on hypotheses, which are effective when the traffic pattern is a linear process with stable fluctuation. Time series analyses such as historical averaging (HA), and autoregressive integrated moving average (ARIMA) are the earliest methods that were applied in traffic prediction. As an extension of ARMA, ARIMA was first used to predict short-term traffic flow on freeways [15]. Several variants such as Kohenen ARIMA [16], ARIMA with explanatory variable [17], and seasonal ARIMA [18] were utilized to improve accuracy. Another method is Kalman filtering (KF) which is based on the Gaussian assumption through kernel function and can be used in nonstationary stochastic processes with updated variables [19].

However, parametric methods may produce biased outcomes under noise and unstable environments. Nonparametric methods with flexible parameters and no assumptions, on the other hand, outperform in nonlinear and uncertain situations. Among nonparametric methods, k-nearest neighbors (k-NN) is a shallow ML technique used by Davis and Nihan [20] for predicting short-term traffic flow on freeways. But one disadvantage of k-NN is that it cannot reveal spatiotemporal correlations simultaneously. Support vector regression (SVR) is another typical algorithm that is referred to as supervised ML. It can manage unstructured data, scales well to high-dimensional data, and ensures global minima localization. Castro-Neto et al. [21] used an online SVR to test the traffic prediction accuracy under different situations. Similarly, random forest (RF) is an ensemble technique that is capable of capturing nonlinearity, particularly when combined with other algorithms [22]. Bayesian networks were also applied by Sun et al. [23] as they provided density function with adaptive variabilities but failed to fit high-dimensional data.

Compared to shallow ML methods, deep learning (DL) methods use multiple layers to extract features and can explore deep correlations embedded in traffic data. Also, DL techniques can deal with the curse of dimensionality and the network is trained end-to-end. Hua and Faghri [24] introduced the concept of traffic forecasting using ANNs to predict travel time. Since then various NNs for traffic prediction came into being. To investigate the situation of data loss, Parmula [25] found that multilayer perceptron (MLP) outperforms auto-encoder (AE). Convolutional NNs have been used in vision-based traffic prediction tasks. Chung and Sohn [26] used CNNs which regard historical data as an image and reflect topological locality. Furthermore, traffic grid data can be transformed into graph form. Graph convolution network (GCN) was developed in this background [27]. To address the vanishing and exploding gradients problem in Recurrent NN (RNN), variants such as Long Short-Term Memory (LSTM), gated recurrent unit (GRU), and time delay NN have been commonly employed [6, 28, 29]. Due to the slow convergence issue, Feng et al. [30] proposed a traffic prediction algorithm using wavelet analysis and extreme ML. Other unsupervised DL methods such as stacked AEs have also been used. Huang et al. [31] utilized Deep Boltzmann Machine (DBM) to support multitask learning, and the model was created collaboratively. Hybrid modeling has been proposed in recent years to increase forecast accuracy. Wu and Tan [32] used LSTM and CNN to integrate spatial and temporal dependency. Li et al. [33] introduced a state space model to compensate for DL techniques' poor interpretability. Recently, deep learning methods have also been used in large-scale network predictions [34], enabling spatiotemporal information to be effectively ensembled in dense urban areas [35].

Traffic parameters can also be predicted by using simulation methods. Given that intelligent vehicles may be technically challenging to implement in ITS, particularly in complicated interactive circumstances, there has been a necessity to simulate approximate actual traffic situations. The intelligent driver model (IDM) is a former state-of-the-practice that has been broadly employed in the micro-simulation of vehicle motions. It generates more realism than most deterministic car-following models [36]. Although a growing number of methods have been developed, the best-performing model for traffic prediction still remains unknown. The accuracy of different models depends on the distinct dataset selected and features inherent in traffic data [4]. Hence, this study develops three models, supervised and unsupervised deep neural networks, and a traditional car-following model using real-world data on freeways and compares the performance.

3. Method

3.1. Definition of Traffic Speed Prediction. Traffic speed prediction is a regression issue related to time series data that can be stated as follows: let X_i^t represent the observed traffic speed at the i th point during the t th time interval on a freeway. Providing a sequence $\{X_i^t\}$ of observed speed, $i = 1, 2, \ldots, N$, $t = 1, 2, \ldots, T$, the task is to predict the traffic speed at time $(t + \Delta)$ for horizon size Δ. Without any assumptions, deep neural networks (DNNs) are a type of ANNs inspired by human neurons. It can mine traffic data by extracting features generated by hierarchical and distributed architecture.

3.2. Supervised Deep Learning Method. Given the sequential features of traffic speed data, recurrent neural networks (RNNs) are particularly suitable for remembering long-term dependencies in this data type. However, it encounters the problem of vanishing gradient when timesteps increases. To solve it, the variant Long Short-Term Memory (LSTM) was put forward. LSTM was first introduced by Hochreiter and Schmidhuber [37] for language processing and used in traffic flow prediction by Ma et al. [28]. Different from RNNs, LSTM regards the hidden layer as a memory cell, which makes it outperforms RNNs due to its ability to flexibly memorize patterns for longer durations. To make the training process more effective and concise, gated recurrent unit (GRU) was introduced by Chung et al. [38]. It removed the separate memory unit without reducing the performance compared to LSTM. Meanwhile, GRU has a smaller number of parameters, which also reduces the risk of overfitting. Figure 1 shows the structure of GRU.

In GRU, the memory unit comprises of two gates, namely, the reset gate and the update gate, which decide what information should be sent to the output layer. It merges the input gate and the update gate into the reset gate, which performs similarly to the LSTM forget gate, and it selects whether to integrate previous and present information, while the update gate determines how much previous information to retain. Equations are given as follows:

$$\begin{aligned} r &= \sigma\left(X_t U_r + S_{t-1} W_r\right), \\ z &= \sigma\left(X_t U_z + S_{t-1} W_z\right), \\ h &= \tanh\left(X_t U_h + \left(S_{t-1} * r\right) W_h\right), \\ S_t &= \left(\left(1 - z\right) * S_{t-1} + z * h\right), \end{aligned} \tag{1}$$

where X_t is input, r is reset gate, z is update gate, h is hidden state output, S_t is output, and all of them are vectors. U and W are corresponding weight parameter matrices for them. GRU uses the sigmoid function σ to activate reset and update gate. It outputs a value from 0 to 1, where 0 denotes no information going through while 1 denotes all information going through the cell state. The tanh function is used to activate the hidden state and outputs a number from −1 to 1.

After the hyperparameter tuning by a manual search, this study designs a 2 hidden layers architecture GRU with 32 neuron units. To avoid the overfitting problem, dropout regularization [39] is set as 0.2. RMSprop [40] is selected as the optimizer, which is a modification of stochastic gradient descend with adaptive learning rates and is used in RNNs to prevent local minimum. Mean square error is utilized as the loss function and the goal is to minimize it. Datasets are classified with 128 batch sizes and trained with 100 epochs.

3.3. Unsupervised Deep Learning Method. Auto-Encoders (AEs) are the typical unsupervised learning method that use unlabeled training [41]. AEs are made up of two basic parts: encoder and decoder, where the encoder compresses the input x whereas the decoder reconstructs the input x'. Similar to the neural network, it also owns one or more hidden layers, and the numbers of units in the input layer and output layer are the same. They can be used for data compression and fusion since they generate comparable input at the output layer. Backpropagation (BP) algorithms are also used to minimize the error function by adjusting the weight parameters and return a target value that is equal to the input.

Stacked AEs (SAEs) are the most prevalent AEs variants. The SAEs can effectively extract data features by stacking numerous AEs into hidden layers using greedy layer-wise training [42]. However, the SAEs have poor generalization and are not suitable for data with network fluctuations. Each AE receives bottleneck activation vector output from lower layers as input. The mechanism of it is to encode the feature vector extracted from the input via an encoder layer, and then, the feature from the previous layer is sent to the following layer until the training process finishes. Last, the

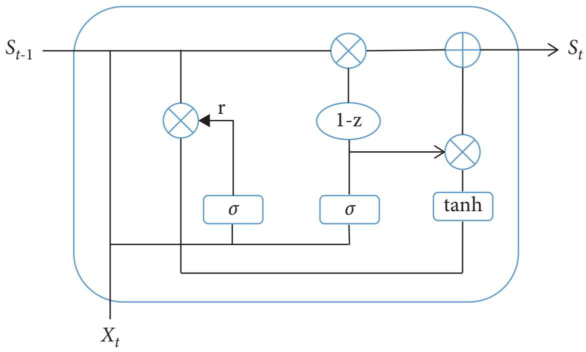

Figure 1: Structure of GRU.

input is reconstructed in the decoder layer. Equations are given as follows:

$$\begin{aligned} y &= f(Wx+b),\\ x' &= g(W'y+b'),\\ \theta &= \arg\min \frac{1}{2}\sum_{i==1}^{N}\|x-x\|^2, \end{aligned} \tag{2}$$

where f and g are sigmoid functions used to activate the encoder and decoder layer, b and b' are the encoder and decoder bias vector, respectively, and W and W' are weight matrices for encoding and decoding, respectively. The parameters are trained by minimizing the error between reconstructed and actual input, which are defined as θ.

This research first designs 3 independent AEs and SAEs that utilize the same hidden layer with 128 neuron units. Dropout regularization is set as 0.2. Adam [43] is selected as the optimizer, which is a combination of RMSprop with Momentum and is used for backpropagation through time. Mean square error is utilized as the loss function. To ensure the same iterations, datasets are also classified with 128 batch sizes and trained with 100 epochs.

3.4. Simulated Car-Following Model. The Intelligent Driver Model (IDM) is a conventional car-following model based on the present state of the object vehicle. Compared to most deterministic car-following models, it produces better realism and can be implemented to model the intelligent driving environment in the simulation. Although the IDM model has few parameters, it can use a unified model to describe different states from free flow to fully congested flow, and it lacks random terms, which is different from the actual vehicle behavior. The core principle of it involves comparing the object vehicle's desired velocity to its real velocity collected from the sensors, as well as comparing its desired headway to its true headway to determine the vehicle's acceleration rate. Equations are given as follows:

$$\begin{aligned} a &= a_m\left[1-\left(\frac{v}{v_0}\right)^{\delta}-\left(\frac{s^*(v,\Delta v)}{s}\right)^2\right],\\ s^*(v,\Delta v) &= s_0+s_1\sqrt{\frac{v}{v_0}}+vT+\frac{v.\Delta v}{2\sqrt{a_m b}}, \end{aligned} \tag{3}$$

where the values of all the parameters in this study are adapted from [36, 44]. a is the acceleration rate of the object vehicle, a_m is the maximum acceleration rate and equals 0.73 m/s^2, v is the current speed of the object vehicle, v is the desired velocity and equals the speed limit m/s, δ is the acceleration exponent and equals 4, s^* $(v,\Delta v)$ is the desired minimum headway, Δv is the velocity difference between the object and the leading vehicle, s is the current headway between the object and the leading vehicle, s_0 is the linear jam gap and equals 2 m, s_1 is the nonlinear jam gap and equals 3 m, T is the desired headway and equals 1.0 s, and b is the comfortable deceleration rate and equals 1.67 m/s^2. It is worth mentioning that there are five parameters, including v_0 desired velocity, a_m maximum acceleration rate, b comfortable deceleration rate, T desired headway, and s_0 linear jam gap that can be calibrated in the simulation according to various scenarios.

The IDM car-following model is applied in the microscopic "Simulation of Urban Mobility" (SUMO) to predict the traffic speed, which is an open access platform developed by the German Institute of Transportation Systems. It provides a Traffic Control Interface (TraCI) to acquire the attributes of traffic parameters. Since this study is mainly devoted to longitudinal traffic speed prediction, the lane-changing model uses the default LC2013. This study first establishes a simulated freeway segment in SUMO, using the traffic flow data provided by the PeMS database as input. Then, let the simulation run by adopting the IDM parameters as discussed above according to a specific time interval, and output the speeds during the corresponding next time period to calculate the average value.

4. Experimental Settings

4.1. Data Collection and Processing. The data is derived from the Caltrans Performance Measurement System (PeMS), which contains data from about 40,000 inductive loop detectors across the highway network in California. Each vehicle detector station collects data every 30 seconds and is aggregated into 5-minute time intervals. Due to the unique patterns of various sequential traffic speed data and that no single pattern can match all-time series data, this study uses the information gathered by a unitary detector.

The experimental scenario is a mainline segment of the I-80 freeway eastbound, Berkeley. The global view of the study area is shown in Figure 2. It is a two-way road with five lanes in each direction, and the average traffic speed from south to north is selected. Since the traffic speed data is periodic and its pattern can differ between weekdays and weekends. This study collects data from March 1st to April 29th on the weekdays of 2022. According to Chen et al. [45], 5-minute traffic is more suitable and predictable. In this experiment, the past 1 hour which is a time sequence of 12 data points is used to predict the coming average traffic speed in the next 5 minutes. Incorporating the periodicity of traffic data over weeks, the whole dataset is divided into

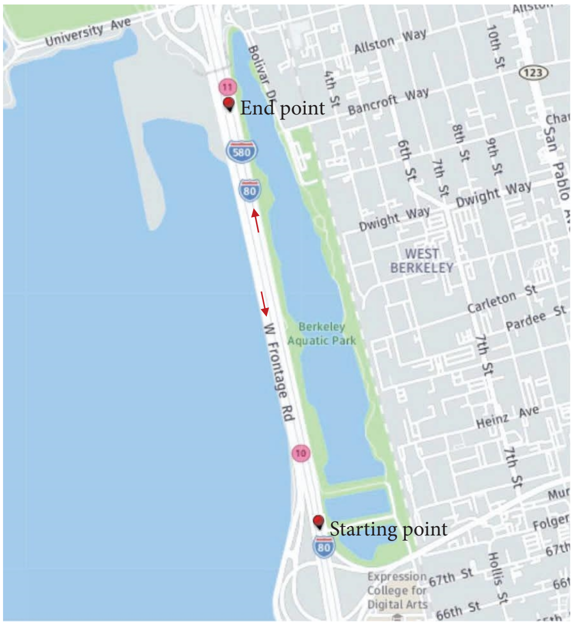

Figure 2: The global view of the study area (*Source.* PeMS).

training and testing sets. The first 33 days (75%) are used as the training set, and the last 11 days (25%) are used as the testing set.

Before training the dataset, normalization is a necessary step to accelerate the gradient descent speed [46]. This study first implements a feature scaler by the training set, then uses the MinMaxScaler to normalize the training set and test set separately. After scaling, data are normalized from 0 to α, where α is a standardized factor that is set as 1 for simpleness. The equation is given as follows:

$$s = \alpha \times \frac{x - \min(x)}{[\max(x) - \min(x)]}. \tag{4}$$

Considering the size of the dataset and the number of hyperparameters, 90% of data is used as training and 10% as validation. Since the sequential traffic prediction needs to use the historical speed to predict the incoming speed, the time lag is utilized to divide the dataset. Since the divided dataset still has a time series feature, this study samples the dataset in order and then shuffles it. Given the modularity and user-friendly interface, the Keras framework which is released in 2015 is used to train the deep learning models and it can run over the popular TensorFlow and Theano.

4.2. Performance Evaluation. To test the prediction accuracy of different models from a comprehensive perspective, there are five metrics mean absolute error (MAE), mean absolute percentage error (MAPE), mean square error (MSE), root mean square error (RMSE), and R^2 are applied to evaluate the performance. Equations are given as follows:

$$\begin{aligned} \mathrm{MAE} &= \frac{1}{N}\sum_{i=1}^{N}\left|x_i - \hat{x}_i\right|, \\ \mathrm{MAPE} &= \frac{1}{N}\sum_{i=1}^{N}\frac{\left|x_i - \hat{x}_i\right|}{x_i}, \\ \mathrm{MSE} &= \frac{1}{N}\sum_{i=1}^{N}\left(x_i - \hat{x}_i\right)^2, \\ \mathrm{RMSE} &= \frac{1}{N}\sqrt{\sum_{i=1}^{N}\left(x_i - \hat{x}_i\right)^2}, \end{aligned} \tag{5}$$

where x_i is the actual average traffic speed and $\hat{x}_i$ is the predicted average traffic speed. The lower these metrics, the better the performance.

5. Results and Discussion

Figure 3 shows the changes in loss function of GRU and SAEs. The loss function is used to measure the degree of consistency between the estimated value of the model and the real value. It is a non-negative real-valued function. The smaller the loss function, the better the robustness of the model. The loss rates of the training set with black line drop rapidly at the beginning before 20 epochs for both GRU and SAEs. With the increase of time, the loss rate of the GRU training set tends to remain flat at the relative minimum value and is infinitely close to 0. For the GRU validation set, there is a small oscillation at the beginning. As the epoch increases, the loss rate continues to decrease, which indicates that the network is still learning. It eventually stabilizes and the validation set converges well, avoiding underfitting and overfitting problems. For the validation set of SAEs, the volatility is significantly larger than that of the supervised learning algorithm. However, it finally stabilizes and fits the training set as the epoch increases. From the performance of the loss function, both deep learning networks are well trained.

Table 1 illustrates the performance of each model based on different statistical metrics. It can be seen that for the MAE, MSE, and RMSE that describe the absolute error, the unsupervised deep learning represented by SAEs is modestly higher than the supervised deep learning represented by GRU, and the performances of both are better than the traditional IDM model. For MAPE describing a relative error, GRU also performs modestly better (3.410%) than SAEs (3.478%), and both outperform the IDM model (5.240%). For the degree of fitness, the R^2 of them are similar (floating around 0.986), demonstrating a relatively good fitting result. Overall, both supervised learning and unsupervised learning methods are superior to the traditional simulation-based car-following model in the prediction of traffic speed. While the difference between the two different deep learning is small, GRU is slightly better than SAEs in

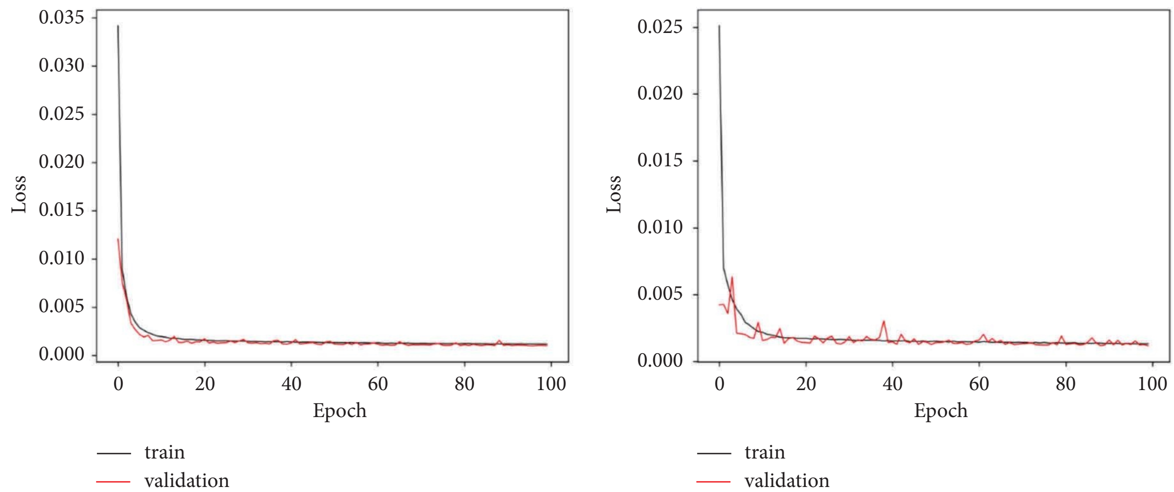

Figure 3: The loss rates of GRU (a) and SAEs (b).

Table 1: Performance comparison of different models.

Model	MAE	MAPE (%)	MSE	RMSE	R^2
GRU	1.352	3.410	4.496	2.120	0.987
SAEs	1.398	3.478	4.950	2.225	0.985
IDM	2.486	5.240	8.896	2.983	0.986

time series prediction. This plays an important role in the application of prediction technology in ITS.

Figure 4 demonstrates the prediction of average speed for different models by the time of the day. The actual value is selected as a baseline with a solid red line. To account for the different traffic states, it is divided into three intervals according to the size of the traffic flow (with dash blue line), low traffic loads, transition state, and heavy traffic loads.

For low traffic loads, it can be classified into two time periods, before congestion (0:00–7:00) and after congestion (19:00–0:00). It can be seen that before congestion, both GRU and SAEs match well with real value. Although IDM model changes more softly, the response at high speed is not timely enough. After congestion, the IDM model cannot revert to the previous accuracy, there is a small gap compared to the original value, but both GRU and SAEs can maintain high accuracy. This shows that the deep learning network can reduce cumulative error propagation over time. Given that the IDM model is collision-free when the distance between the front and rear vehicles decreases sharply, the IDM model will produce strong braking on the target vehicle, which is unrealistic in reality. This is also the problem with the simulation-based car-following model. Transition state is classified into buildup of congestion (7:00–10:00 and 12:00–15:00) and dissipation of congestion (11:00–12:00 and 18:00–19:00). For the buildup of congestion, IDM's performance is inferior to deep learning networks. In addition, IDM still cannot rebound to the previous accuracy in dissipation of congestion. According to the length of the congestion time, heavy traffic loads are classified into short-term full congestion (10:00–11:00) and long-term full congestion (15:00–18:00). In short-term full congestion, all models have different degrees of bias, and the most obvious one goes to the IDM. For long-term full congestion (15:00–18:00), the situation is similar to the before congestion state under the low traffic loads. The three models perform almost the same, but IDM is smoother and with less fluctuation.

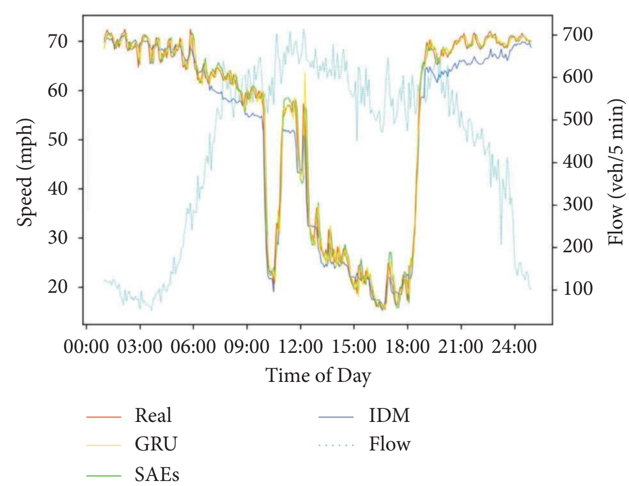

Figure 4: Prediction of the average speed of different models by the time of day.

This study also investigates the speed distribution for different models by time of day with a heatmap, which is displayed in Figure 5. There are two points worth noting. Firstly, for a short period from 10:00 to 10:05, there is a certain prediction delay for both GRU and SAEs, and this phenomenon can continue until the congestion dissipates at 18:00. However, this situation does not exist in the IDM model, which suggests that for short-term slowdowns, IDM can detect the buildup of congestion earlier than deep learning networks. Another finding is that after congestion at 18:30, all models have a prediction lag of about five minutes. However, from the dark blue area afterward, the accuracy of deep learning networks recovers faster than IDM. The above analysis reveals that deep learning networks and simulation-based car-following models have their latent performance features for different time dimensions.

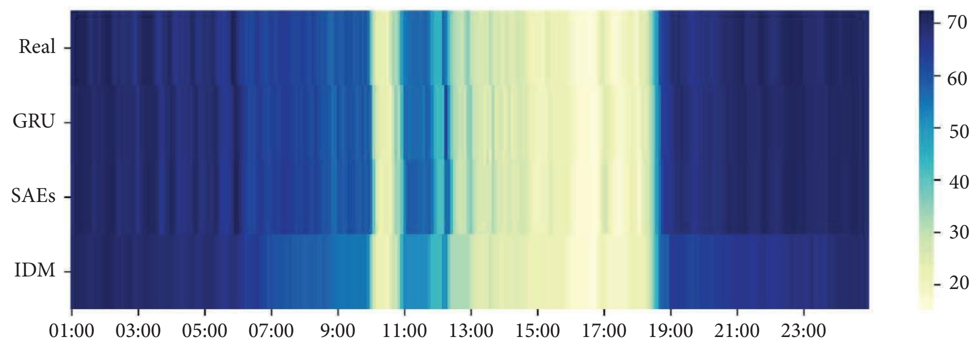

Figure 5: Speed distribution for different models by time of day.

6. Conclusion

The development of intelligent transportation systems has given impetus to intelligent vehicles, which have the potential to address the traffic congestion problem. Meanwhile, it also brings real-time traffic prediction issues. Given the complex and dynamic spatiotemporal dependency embedded in traffic data, traditional prediction models have many drawbacks.

In order to improve the accuracy of traffic speed prediction, this study focuses on emerging deep neural networks using real-world traffic data. Additionally, a simulation-based model is built for intelligent vehicles in SUMO. A series of statistical evaluation metrics, MAE, MAPE, MSE, RMSE, and R^2 are employed to assess the prediction accuracy of the supervised learning method, unsupervised learning method, and simulation-based model. The PeMS dataset is used to train and evaluate the constructed DNNs, and the results suggest that both GRU and SAEs outperform the traditional IDM model in the prediction of traffic speed on the freeway. In addition, there is no difference between the deep learning networks, and GRU outperforms SAEs slightly in time series prediction. It also demonstrates that car-following simulation-based models and deep learning networks both contain latent performance attributes for various time dimensions under low, transition state, and heavy traffic loads. This has a significant impact on how prediction technology is applied in ITS. The outcomes can assist researchers and traffic engineers to improve dynamic traffic control, such as highway operation, bottleneck detection, and level of service assessment. The predicted traffic speed can also be used for further research on variable speed limit control, platooning management, and route guidance, etc.

This study mainly uses traffic speed as the input for prediction. Future research work can introduce hand-engineering factors, such as weather, events, and other traffic parameters. Moreover, more spatiotemporal dependency can be captured by more advanced deep learning networks. In addition, attention mechanism can be combined to model the long sequence data [47]. For the simulation environment, it can focus on improving the car-following model [48]. The lane-changing model can also be considered to better simulate intelligent driving behaviors. Lastly, the transferability issue that all adaptive frameworks face could be addressed, especially in metropolitan areas.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors want to express their deepest gratitude to the financial support by the United States Department of Transportation, University Transportation Center, through the Center for Advanced Multimodal Mobility Solutions and Education at the University of North Carolina at Charlotte (Grant Number: 69A3551747133).

References

[1] J. Liu, N. Wu, Y. Qiao, and Z. Li, "A scientometric review of research on traffic forecasting in transportation," *IET Intelligent Transport Systems*, vol. 15, no. 1, pp. 1–16, 2021.

[2] A. Miglani and N. Kumar, "Deep learning models for traffic flow prediction in autonomous vehicles: a review, solutions, and challenges," *Vehicular Communications*, vol. 20, Article ID 100184, 2019.

[3] R. Yu, Y. Li, C. Shahabi, U. Demiryurek, and Y. Liu, "Deep learning: a generic approach for extreme condition traffic forecasting," in *Proceedings of the 2017 SIAM International Conference on Data Mining*, pp. 777–785, Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, USA, 2017, June.

[4] Y. Lv, Y. Duan, W. Kang, Z. Li, and F. Y. Wang, "Traffic flow prediction with big data: a deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 1–9, 2014.

[5] Y. Tian and L. Pan, "Predicting short-term traffic flow by long short-term memory recurrent neural network," in *Proceedings of the In 2015 IEEE International Conference on Smart city/SocialCom/SustainCom (SmartCity)*, pp. 153–158, IEEE, Chengdu, China, 2015, December.

[6] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," in *Proceedings of the 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, pp. 324–328, IEEE, Wuhan, China, 2016, November.

[7] P. Gora, C. Katrakazas, A. Drabicki, F. Islam, and P. Ostaszewski, "Microscopic traffic simulation models for connected and automated vehicles (CAVs)–state-of-the-art," *Procedia Computer Science*, vol. 170, pp. 474–481, 2020.

[8] W. Do, O. M. Rouhani, and L. Miranda-Moreno, "Simulation-based connected and automated vehicle models on highway sections: a literature review," *Journal of Advanced Transportation*, vol. 2019, pp. 1–14, 2019.

[9] D. Helbing, M. Treiber, A. Kesting, and M. Schönhof, "Theoretical vs. empirical classification and prediction of congested traffic states," *The European Physical Journal B*, vol. 69, no. 4, pp. 583–598, 2009.

[10] S. Ishak and H. Al-Deek, "Performance evaluation of short-term time-series traffic prediction model," *Journal of Transportation Engineering*, vol. 128, no. 6, pp. 490–498, 2002.

[11] J. Li, L. Gao, W. Song, L. Wei, and Y. Shi, "Short term traffic flow prediction based on LSTM," in *Proceedings of the Ninth International Conference on Intelligent Control and Information Processing (ICICIP)*, pp. 251–255, IEEE, Wanzhou, China, 2018, November.

[12] N. G. Polson and V. O. Sokolov, "Deep learning for short-term traffic flow prediction," *Transportation Research Part C: Emerging Technologies*, vol. 79, pp. 1–17, 2017.

[13] Y. Kamarianakis and P. Prastacos, "Forecasting traffic flow conditions in an urban network: comparison of multivariate and univariate approaches," *Transportation Research Record*, vol. 1857, no. 1, pp. 74–84, 2003.

[14] E. I. Vlahogianni, J. C. Golias, and M. G. Karlaftis, "Short-term traffic forecasting: overview of objectives and methods," *Transport Reviews*, vol. 24, no. 5, pp. 533–557, 2004.

[15] M. S. Ahmed and A. R. Cook, *Analysis of Freeway Traffic Time-Series Data by Using Box-Jenkins Techniques*, 1979.

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

[17] B. M. Williams, "Multivariate vehicular traffic flow prediction: evaluation of ARIMAX modeling," *Transportation Research Record*, vol. 1776, no. 1, pp. 194–200, 2001.

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

[19] I. Okutani and Y. J. Stephanedes, "Dynamic prediction of traffic volume through Kalman filtering theory," *Transportation Research Part B: Methodological*, vol. 18, no. 1, pp. 1–11, 1984.

[20] G. A. Davis and N. L. Nihan, "Nonparametric regression and short-term freeway traffic forecasting," *Journal of Transportation Engineering*, vol. 117, no. 2, pp. 178–188, 1991.

[21] M. Castro-Neto, Y. S. Jeong, M. K. Jeong, and L. D. Han, "Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions," *Expert Systems with Applications*, vol. 36, no. 3, pp. 6164–6173, 2009.

[22] G. Leshem and Y. A. Ritov, "Traffic flow prediction using adaboost algorithm with random forests as a weak learner," *International Journal of Mathematics and Computer Science*, vol. 1, no. 1, pp. 1–6, 2007.

[23] S. Sun, C. Zhang, and Y. Zhang, "Traffic flow forecasting using a spatio-temporal bayesian network predictor," in *Proceedings of the International Conference on Artificial Neural Networks*, pp. 273–278, Springer, Berlin, Heidelberg, 2005, September.

[24] J. Hua and A. Faghri, "Apphcations of artificial neural networks to intelligent vehicle-highway systems," *Transportation Research Record*, vol. 1453, p. 83, 1994.

[25] T. Pamuła, "Impact of data loss for prediction of traffic flow on an urban road using neural networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 3, pp. 1000–1009, 2019.

[26] J. Chung and K. Sohn, "Image-based learning to measure traffic density using a deep convolutional neural network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 5, pp. 1670–1675, 2018.

[27] L. Zhao, Y. Song, C. Zhang et al., "T-gcn: a temporal graph convolutional network for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3848–3858, 2020.

[28] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 187–197, 2015.

[29] B. Abdulhai, H. Porwal, and W. Recker, *Short Term Freeway Traffic Flow Prediction Using Genetically-Optimized Time-Delay-Based Neural Networks*, 1999.

[30] W. Feng, H. Chen, and Z. Zhang, "Short-term traffic flow prediction based on wavelet function and extreme learning machine," in *Proceedings of the 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pp. 531–535, IEEE, Beijing, China, 2017, November.

[31] W. Huang, G. Song, H. Hong, and K. Xie, "Deep architecture for traffic flow prediction: deep belief networks with multitask learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2191–2201, 2014.

[32] Y. Wu and H. Tan, "Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework," 2016, https://arxiv.org/abs/1612.01022.

[33] L. Li, J. Yan, X. Yang, and Y. Jin, "Learning Interpretable Deep State Space Model for Probabilistic Time Series Forecasting," 2021, https://arxiv.org/abs/2102.00397.

[34] Y. Liu, C. Lyu, Y. Zhang, Z. Liu, W. Yu, and X. Qu, "DeepTSP: deep traffic state prediction model based on large-scale empirical data," *Communications in transportation research*, vol. 1, Article ID 100012, 2021.

[35] Y. Liu, Z. Liu, H. L. Vu, and C. Lyu, "A spatio-temporal ensemble method for large-scale traffic state prediction," *Computer-Aided Civil and Infrastructure Engineering*, vol. 35, no. 1, pp. 26–44, 2020.

[36] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical Review*, vol. 62, no. 2, pp. 1805–1824, 2000.

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

[38] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," 2014, https://arxiv.org/abs/1412.3555.

[39] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[40] G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent," *Cited on*, vol. 14, no. 8, p. 2, 2012.

[41] C. Y. Liou, W. C. Cheng, J. W. Liou, and D. R. Liou, "Autoencoder for words," *Neurocomputing*, vol. 139, pp. 84–96, 2014.

[42] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, U. D. Montral, and M. Qubec, *Greedy Layer-wise Training of Deep Networks*, NIPS, Noida, 2007.

[43] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," 2014, https://arxiv.org/abs/1412.6980.

[44] P. Liu and W. Fan, "Extreme gradient boosting (XGBoost) model for vehicle trajectory prediction in connected and autonomous vehicle environment," *Promet - Traffic & Transportation*, vol. 33, no. 5, pp. 767–774, 2021.

[45] C. Chen, Y. Wang, L. Li, J. Hu, and Z. Zhang, "The retrieval of intra-day trend and its influence on traffic prediction," *Transportation Research Part C: Emerging Technologies*, vol. 22, pp. 103–118, 2012.

[46] D. Zhang and M. R. Kabuka, "Combining weather condition data to predict traffic flow: a GRU-based deep learning approach," *IET Intelligent Transport Systems*, vol. 12, no. 7, pp. 578–585, 2018.

[47] C. Zheng, X. Fan, C. Wang, and J. Qi, "Gman: a graph multi-attention network for traffic prediction," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, pp. 1234–1241, 2020, April.

[48] D. Salles, S. Kaufmann, and H. C. Reuss, "Extending the intelligent driver model in SUMO and verifying the drive off trajectories with aerial measurements," in *Proceedings of the SUMO User Conference*, 2020, October.