A New Framework for Regional Traffic Volumes Estimation with Large-Scale Connected Vehicle Data and Deep Learning Method

Swastik Khadka[1]; Peirong "Slade" Wang[2]; Pengfei "Taylor" Li, Ph.D., P.Eng., M.ASCE[3]; and Francisco J. Torres, P.E.[4]

Abstract: Connected vehicle (CV) data in this paper refer to the in-vehicle telematic data, including trajectories and driving events (e.g., hard braking) collected by vehicle manufacturers when vehicles are moving. Recently manufactured vehicles are equipped with cellular modems and Internet of Things (IoT) devices to collect vehicle data. Such data, after removing personal information, are being redistributed to third-party organizations. Compared to other probe vehicle data, the CV data has a higher penetration rate, ubiquitous coverage, and almost lane-level positioning accuracy. These features pave the road for novel transportation applications in transportation planning and traffic operations. In this paper, we represent a novel framework to estimate the regional link volumes based on the CV data and a deep neural network (DNN) model. The training data are generated according to the link volumes (targeted model output) and the corresponding CV counts (input features) at the same locations. The DNN model's performance was compared with other estimation methods like linear regression and random forest and showed superior performance. The trained DNN model takes ubiquitous CV counts from other locations to estimate the corresponding link volumes. As a case study, the proposed DNN model was trained with a large training data set derived from CV data and time-dependent link counts collected at over 1,200 locations on freeways in the Dallas Fort Worth, Texas, area. The results reveal good accuracy and robustness. **DOI: 10.1061/JTEPBS.TEENG-7536.**

Author keywords: Connected vehicle (CV) data; Travel demand estimation; Deep learning; Transportation planning; Deep neural network (DNN).

Introduction

Most recently manufactured vehicles are equipped with cellular modems and global positioning system (GPS) modules. Automobile manufacturers add these devices primarily to collect driving behaviors and the corresponding vehicle behaviors. They investigate these data to provide drivers with real-time service (e.g., roadside assistance) as well as improve their products' competitiveness (e.g., fuel efficiency). These data were confidential in the past, but some manufacturers decide to remove the private information and redistribute the recorded vehicle trajectories and driving events. Such emerging data sets are referred to as (internet-based) connected vehicle data or CVD. Compared with the traditional fixed-spot traffic detectors, the connected vehicle (CV) data almost cover any major roads at any time. For example, our preliminary analysis reveals that the CV data's penetration rate in the Dallas Fort Worth (DFW) area in Texas, the fourth largest metropolitan area in the US, is 2%–6% on average. The CV data's positioning accuracy can mostly reach the level of lanes. Compared to the traditional GPS data of probe vehicles or smartphones, the new CV data has ubiquitous, continuous, and consistent coverage (CV traces were identified on each road segment in the DFW area). It also has a higher data quality and the highest penetration rate (the number of vehicles contributing data versus the total number of vehicles) of a single mobility data source. The spatiotemporal information contained in the CV data makes it possible to accurately estimate and visualize traffic states.

Link volumes and speeds are two important components of traffic states. While the low penetration rate will not affect the speed estimation, the CV data cannot be directly used to estimate link volumes. Nonetheless, the consistency and ubiquitous coverage of the new CV data set provide promises to apply the CV data, in conjunction with the infrastructure data, to link volume estimation. A major contribution of this paper is that we present a new framework to explore the potential of the new CV data in estimating the regional time-dependent link volumes. This framework starts with generating a training data set by coupling over 1,000 locations where link volumes (i.e., 100% counts) were collected using roadside traffic detectors with the corresponding connected vehicle counts at those locations. Then various estimating techniques, from linear regression to deep learning, are used to develop traffic volume estimation models for all road links, especially those links without infrastructure sensors. The proposed method provides an alternative method to estimate regional travel demand (i.e., full-spectrum link traffic volumes) to the traditional travel demand modeling. It is driven by data analytics with few assumptions on traveling behaviors.

[1]Graduate Research Assistant, Dept. of Civil Engineering, Univ. of Texas at Arlington, Arlington, TX 76019. Email: swastik.khadka@mavs.uta.edu

[2]Graduate Research Assistant, Dept. of Civil Engineering, Univ. of Texas at Arlington, Arlington, TX 76019. ORCID: https://orcid.org/0000-0002-9636-7047. Email: peirong.wang@mavs.uta.edu

[3]Assistant Professor, Dept. of Civil Engineering, Univ. of Texas at Arlington, Arlington, TX 76019 (corresponding author). ORCID: https://orcid.org/0000-0002-3833-5354. Email: Taylor.Li@uta.edu

[4]Principal Transportation System Modeler, Dept. of Model and Data Development, North Central Texas Council of Governments, Arlington, TX 76011. Email: ftorres@nctcog.org

Note. This manuscript was submitted on May 18, 2022; approved on November 16, 2022; published online on January 27, 2023. Discussion period open until June 27, 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.

Literature Review

Traffic states, such as volumes, estimation, and prediction are important for proactive congestion management. From the traditional regression models to the more recent deep neural networks, there is a rich body of related literature. The methods described in the previous literature can be categorized into two types in principle: parametric and nonparametric approaches. The literature is reviewed and summarized accordingly.

Parametric Approach

Parametric approaches mostly depend upon certain physics and/or mathematical models along with assumptions, such as prior distributions or linearities, and so forth. Parametric models are based on certain assumptions (e.g., the Gaussian process), thus the model's performance may not always be satisfying because such prior settings may violate the truth. One of the most popular parametric estimation methods is the autoregressive integrated moving average (ARIMA) method, including a few variants. Hamed et al. (1995) created the simple ARIMA model of order (0,1,1) to estimate the traffic volumes in an urban arterial. The advantages of their ARIMA model are straightforward and implementable, simply requiring storing the latest anticipated error and the latest traffic observation. Later, the same model was adopted in an improved Bayesian combination model for short-term traffic volume prediction with deep learning techniques. Ding et al. (2011) introduced a new space-time ARIMA (STARIMA) model to predict the traffic volumes of the next 5 min across multiple cities. Okutani et al. used the Kalman Filter for traffic flow prediction (Okutani and Stephanedes 1984). In this method, the model parameters were improved using the latest observed prediction errors. Similar techniques were extended with various new forms of Kalman filters like adaptive Kalman filters (Guo et al. 2014) and extended Kalman filters (Wang and Papageorgiou 2005). The accuracy for the extended forms of Kalman filters was reportedly better and suitable for more scenarios compared to the basic Kalman filter. Tak et al. (2016) used a modified K-nearest neighbor (KNN) approach, a data-driven imputation method, to estimate road conditions according to the geographical and temporal data sets. This approach can estimate the missing data through imputation. With a historical data set for 400 days, the proposed KNN approach outperformed the analytical regression models in all scenarios, including those with missing data. It suggests that the KNN model can offer a resilient traffic state estimation with satisfying accuracy. The KNN model can also be integrated with certain distributed computing techniques to further improve the accuracy of traffic state estimation. Yin et al. (2012) proposed a kernel regression spatial method to incorporate the land use percentage in the estimation process. The Bayesian particle filter (BPF) was utilized by Polson and Sokolov (2018) to predict traffic regimes such as free flow, breakdown, and recovery. This model can capture the traffic's nonlinearities and discontinuities present in traffic flow data. The BPF method contains two steps: (1) resample the current particles with a mixed predictive distribution, and (2) use the conditional posterior distribution to propagate traffic states (Polson and Sokolov 2018). This model is flexible because it can minimize assumptions with respect to the sensor locations for data collection.

Nonparametric Approach

The nonparametric approach is a statistical method that does not assume the sample's characteristics. No prior assumptions exclude subjective bias about the data sets. Examples of nonparametric approaches include machine learning models, deep neural networks, random forest search, K-nearest neural networks, and so forth. Lv et al. (2014) used a deep architecture model trained with a big set of traffic data to predict traffic flow in each link of a road network. They used the stacked autoencoders (SAE) model. This model considers the spatial and temporal correlation inherently, which means the model can discover the latent traffic flow feature that is nonlinear with higher estimation accuracy. Xu et al. (2020) proposed a novel deep learning framework, referred to as the graph-embedding generative adversarial network (GE-GAN) model, to estimate the incomplete traffic state of a road based on data on its adjacent link. Initially, a graph-embedding (GE) based model of a roadway network is created. Then, using a generative adversarial network (GAN), the represented graph is used to create real-time information on road traffic conditions. GAN is applied to learn the traffic state distribution. The outputs from the novel GE-GAN deep learning network reveal high accuracies in estimating the traffic state compared to other state-of-the-art road traffic estimation methods. Lu et al. (2021) proposed a combined method for short-term traffic flow prediction that is based on a recurrent neural network (RNN). The model consists of a simple ARIMA model combined with a long short-term memory (LSTM) neural network. The ARIMA model enables capture of the linear regression feature of the traffic data and then, using the backward propagation LSTM network, the nonlinear features of traffic data were captured. Finally, using dynamic weights of a sliding window, the estimated values of these models were combined to obtain the output. Since both linear and nonlinear aspects of the traffic flow data were studied during the model running process this makes this model much more versatile compared to others. Sekuła et al. (2018) proposed a method to estimate the historical hourly traffic volumes using a feedforward neural network and vehicle probe data. They studied the application of neural networks, vehicle probe data, and automatic traffic recorder (ATR) counts to estimate the hourly volumes. Several features like vehicle probe speeds, weather data, infrastructure data, and so forth were considered while calibrating the neural network. Additionally, they combined the model with the existing profiling method, which on average yields highly accurate data compared to the profiling method by itself. Later, Zahedian et al. (2020) added a few features to the same model called selected ATR counts as an additional input. In this model they selected a subset of available ATR counts and used it as an input variable in a neural network model. By selecting ATR stations according to traffic message channel (TMC) and training an artificial neural network with their input ensures that there is a significant improvement in the output estimated by the model. Duan et al. (2016) proposed a deep learning model named denoising stacked autoencoders (DSAE) for imputing the missing and corrupt data from the traffic counts. This model has two basic blocks: autoencoders (AEs) and denoising autoencoders (DAEs). The AE component of the model helps to extract the features from the input data and DAE has the ability of cleaning and denoising data. They trained the deep neural network hierarchically using data from the vehicle detector station. The results show better performance while comparing their model output with the output from other models like ARIMA and back propagation (BP) neural network. Similar to this technique, Markov chain Monte Carlo multiple imputations had been used to estimate the missing data in intelligent transportation system data (Ni and Leonard 2005). The method employs a Bayesian network to learn from raw data and a Markov chain Monte Carlo methodology to sample from the Bayesian network's probability distributions. This method of estimation deals with time series models. The Bayesian model includes ARIMA as a regression model. The partial data problem is then handled by solving complete problems iteratively and gradually until the method converges. Pun et al. (2019)

proposed a multiregression technique by merging five tropical measures and road length to estimate the volume counts of the traffic. The advantage of this technique is that it integrates the topological and geometrical properties of the roadway segment, which helps in estimating traffic flow more accurately. Six different measures like degree, betweenness, closeness, page rank, clustering coefficient, and road length were used for estimation purposes. Linear regression and random forest were considered during the formulation of the regression model. Their findings imply that the combination of topological and geometrical measurement outperforms in estimating traffic flow compared to a single method. The study is particularly useful for those who are trying to estimate the traffic flow based on correlations but has limited flow data with road network features. Yaghoubi et al. (2021) introduced a traffic flow estimation method using a widely available long-term evolution (LTE)/4G radio frequency performance measurement counter. The estimation is based on a regression model using both classical and deep learning methods. LSTM and random forest regression models have been used. Using transfer learning techniques, they estimated the traffic volume for places where traffic sensors were missing. This model captures the traffic characteristics very well along with providing enough information regarding traffic flow estimation. The experiment was performed on a small scale (i.e., only six locations). The author hypothesizes that with larger data and more location there is a high possibility of improvement in the performance of estimating (Yaghoubi et al. 2021). Heshami and Kattan (2021) applied a case-based reasoning algorithm combined with the Kalman filter (KF) to estimate the real-time queue length on a freeway off-ramp. The KF here is used to fine-tune the final estimation obtained from the model. The occupancy was used as the input and was obtained from the roadside loop detectors installed on a ramp. They further used sensitivity analysis to ensure the performance of the algorithm and the results were promising in terms of estimating and predicting the queue length to an accuracy of ± 3.15 vehicles in a 60-s time interval.

Some of the research was also based on real-time traffic estimation. Khan et al. (2017) developed a novel framework, which combines connected vehicle technology (CVT) with artificial intelligence (AI) together to estimate the real-time traffic state. The assumption made for the CVT-AI model is that the vehicle onboard units will transfer the connected vehicle data to the roadside infrastructure. Distance headway, speed, and the number of stops were considered as input for the model to estimate the density of a given network. The result obtained from the experiments highly suggests that the higher penetration rate of CV will yield more accurate outputs and vice versa. Li et al. (2021) used a multimodel machine learning technique and Gaussian process regressor (GPR) for traffic flow estimation. Their work shows how a machine learning approach based on aggregated data can be used to estimate traffic flow based on floating car data (FCD) (i.e., Google maps data just using a learned regressor). The different regressors were trained to fit into the multimodel machine learning methods. In total, 19 regression methods, such as linear regression model, regression trees, support vector machines (SVM), the ensemble of trees, and Gaussian process regression model, were used for estimation purposes. Comparison of results from single-model and multimodel variants suggests that the multimodel outperforms the single-model variant in generating precisely estimated traffic flow data. Antoniou et al. (2013) used data-driven computational approaches for local traffic state estimation and prediction. The technique they established in their paper is ideal to be used in microscale traffic models, rather than typical speed-density correlations. Their research includes two data sets and surveillance data. Clustering and classification techniques were used in their model. The classification technique was performed with a single hidden layer of neural network. The methodology is a two-stage process, with the first step assigning an observation to a traffic state, and the second step using a state-specific function to approximate the related speed. Since the suggested model outperforms the current state-of-the-art model, it may be useful when combined with other existing traffic state estimation models. Liu et al. (2019) proposed a fully convolutional model based on semantic segmentation technology referred to as the spatiotemporal ensemble net. It is one of a few ensemble techniques designed for the spatiotemporal data set. The ensemble technique allows us to integrate multiple traffic state estimation and prediction models to enhance the prediction and estimation accuracy. Five different models were combined to generate a single model. The models included KNN, linear regression, and a gradient boosting model (GBM) called LightGBM with its different variants. The advantage of the ensemble technique is that the multiple outputs of weak learners can be blended using ensemble learning to create a superior learner that can outperform the individual model (Dietterich 2000). Chen et al. (2020) proposed an improved wavelet neural network (WNN) prediction model to predict short-term traffic flow. WNN is a forecasting model that has strong nonlinear processing power, self-organization, and self-adaptation learning ability. To optimize this network, an improved particle swarm optimization has been used in its architecture. The input supplied to this network is the data collected from the roadside detectors. The comparison has been done between the actual wavelet neural network with improved WNN and the results indicate that the improved version of WNN has outperformed the traditional one. Part of the literature is summarized in Table 1.

Compared with other similar work with other GPS trajectory data sets (e.g., Sekuła et al. 2018), the proposed framework needs much fewer input features to achieve better performance [e.g., mean absolute error (MAE) and R^2]. For example, Sekuła et al. (2018) used over 20 input features and the penetration rate of their GPS data set was lower than the CV data set we use in this paper. It was also recognized that the model's performance steadily improved with the increase in the training data size. Therefore, we anticipate that the proposed method will be capable of tackling even larger networks if there are sufficient data for the model training.

CV Data Reduction and Analytics

For an emerging data source, it is necessary to explore the CV data's features and cross-compare it with the existing traffic data sources (e.g., roadside traffic counts). The first challenge of CV data processing is that the size is much bigger than traditional traffic data. For a metropolitan region, it can easily reach several terabytes of text files every month. The data reduction and analytics are to develop new knowledge of spatiotemporal features contained in the CV data sets in conjunction with the existing traffic data. To tackle the CV data at a manageable level, the first step is to reduce the CV data covering the whole region to smaller areas of interest, such as intersections, corridors, or road segments where traffic detectors are installed. Khadka et al. (2022) proposed an efficient method to reduce the regional CV data to local areas of interest. The size of reduced CV data is then suitable for most data packages. Geofence creation depends on the applications. Fig. 1(a) demonstrates how an hour of regional CV data was reduced around the locations with infrastructure detectors. A small geofence along the road segments is generated around each infrastructure detector, and only the waypoints within the geofence are kept for further processing.

Table 1. Summary of selected literature

Author name	Type	Methodology	Data source/type	Objective	Strength
Hamed et al. (1995)	P	ARIMA	Link counts	Predict volume	Model can be easily implemented and is computationally tractable
Guo et al. (2014)	P	Adaptive Kalman filter	Link counts	Predict volume	Improved adaptability when traffic is highly volatile
Polson and Sokolov (2018)	P	Bayesian particle filter (BPF)	Link counts	Traffic state estimation	Capture nonlinearities and discontinuities
Tak et al. (2016)	NP	*K*-nearest neighbor	Dedicated short-range communications (DSRC) detectors for speed data	Traffic state estimation using imputation technique	Outperformed almost all the missing cases; robust and accurate for over 400 days of data
Xu et al. (2020)	NP	GE-GAN	Fixed-spot traffic counts [California Department of Transportation Performance Measurement System (Caltrans PeMS)]	Traffic state estimation	Average error (RMSE & MAE) is lower compared to other models
Sekuła et al. (2018)	NP	Feedforward neural network	GPS probe data Traffic counts	Traffic flow estimation (hourly)	Up to 24% better than the traditional profiling method
Ni and Leonard (2005)	NP	Bayesian Network ARIMA Model	Video-based traffic counts	Traffic state estimation	Graphical comparison and quantitative assessment reveal a very small imputation error
Yaghoubi et al. (2021)	NP	Supervised regression using deep learning	Cellular radio frequency	Traffic flow estimation	This model is not a perfect estimator, yet it still captures the shape of the traffic very well
Heshami and Kattan (2021)	NP	Case-based reasoning combined with the Kalman filter	Traffic counts and occupancies	Queue length estimation	Results show an accuracy of ±3.15 vehicles in the queue in 60-second time intervals
Liu et al. (2019)	NP	Convolutional model based on semantic segmentation technology	GPS trajectory records	Traffic state estimation	Spatiotemporal ensemble net models can be combined to improve prediction accuracy

Note: P = parametric; and NP = nonparametric.

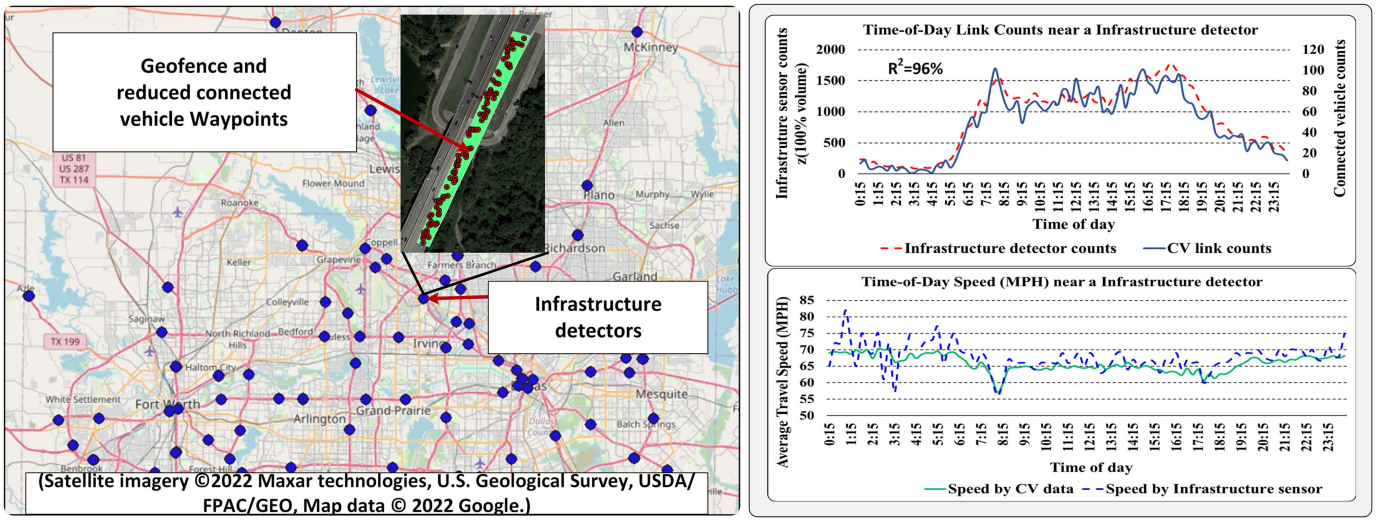

Fig. 1. (a) Demonstration of CV data reduction to the areas of interest. (Imagery ©2022 Maxar technologies, U.S. Geological Survey, USDA/FPAC/GEO, Map data © 2022 Google.); and (b) Comparison between CV data and infrastructural detector data.

Time of Day Connected Vehicle Counts on Selected Links

The reduced CV data are around the infrastructure sensors, which can report nearly 100% of passing vehicles and average speed periodically. We first compared the time of day connected vehicle counts on the corresponding road segments. Fig. 1(b) shows the comparison of 15-min vehicle counts and estimated travel speeds between the CV data and infrastructure sensor data at one location. We compared 168 locations (i.e., geofences), and concluded that the time of day (TOD) trend of CV counts and traffic counts as well as the estimated speeds are highly consistent. This finding paves the road for developing an estimation model to estimate link volumes according to the CV counts.

CV Data Penetration Rates and Reliability

Sekuła et al. (2018) concluded that the penetration rate has a direct correlation with the model's accuracy. The outcome will be better if the penetration rate increases. The experimented penetration rate ranges from 0.78% to 4.56%. As a result, it is advised to have data with greater penetration rates as much as possible instead of

J. Transp. Eng., Part A: Systems, 2023, 149(4): 04023015

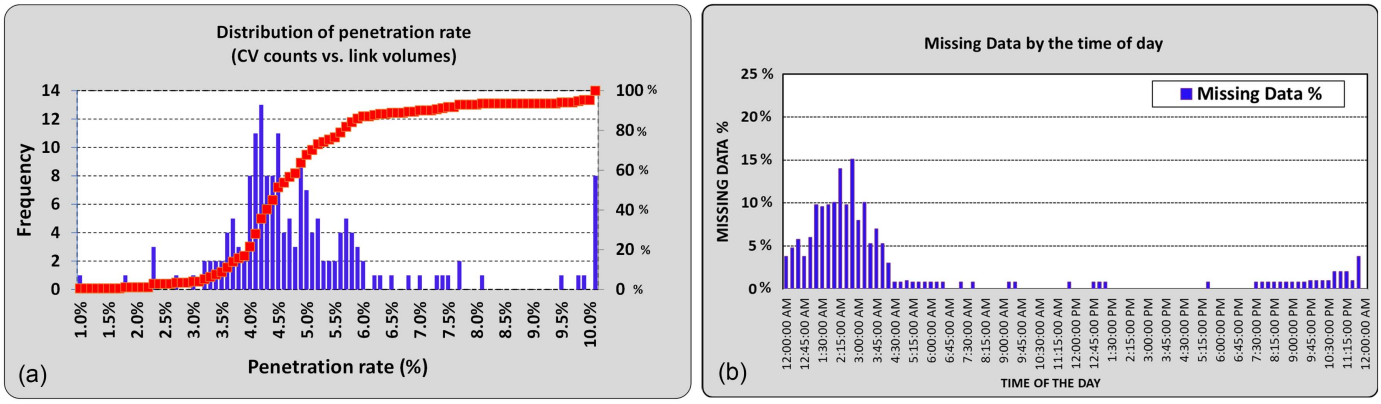

Fig. 2. CV (a) data penetration rate; and (b) missing data percentage.

considering the minimal penetration rate. We further compared the link volumes and the CV counts near the infrastructure sensors at all available locations and found that penetration rates of CV data are mostly between 2% (15 percentile) and 6% (85 percentile) [see Fig. 2(a)]. Given the penetration rate is relatively low, we further analyzed the missing data issue (no captured connected vehicles) during a day within 6 weeks of CV data in the DFW area (August 2021–September 2021). We find that most missing data occurred between 11 p.m. and 5 a.m. the next day. This makes sense because traffic is very light during that period. In particular, the highest missing data rate occurred around 3 a.m. every day.

Methods for Map Matching

Map-Matching Algorithm for the CV Data Set

CV trips are represented with a series of waypoints (latitude, longitude, and time) and they do not contain road information. To make them useful for understanding traffic conditions (e.g., CV link counts), it is necessary to map each waypoint to road links to reveal routes or paths. There are two challenges in this task: (1) a road network may not cover all the CV movements; and (2) various waypoints of a CV trip may be matched on different links. To address these issues, the proposed map-matching algorithm contains three components: (1) the shortest distance to a link in the road network; (2) the difference between a waypoint heading (direction) and road link heading; and (3) matched links of the last few waypoints. To match the waypoints to the nearest link, we use the minimal vertical distance between a waypoint and a road link. The distance between a waypoint and a road link is the minimum length required to move from a waypoint to the road link.

Let P_1, P_2 denote the starting node and ending node of a road link l. The coordinates are $(x_1, y_1), (x_2, y_2)$, respectively, and (x_0, y_0) denote a waypoint's coordinates. Furthermore, let a, b, and c denote the point-to-point distance between (x_0, y_0) and (x_1, y_1), between (x_1, y_1) and (x_2, y_2), and between (x_0, y_0) and (x_2, y_2), respectively; $L = \dfrac{(a+b+c)}{2}$. The distance d (see Fig. 3) from (x_0, y_0) to l can be calculated as

$$d = \begin{cases} b; & \text{if } (b^2 \geq c^2 + a^2) \\ c; & \text{if } (c^2 \geq b^2 + a^2) \\ \dfrac{2\sqrt{L(L-a)\times(L-b)\times(L-c)}}{a}; & \text{otherwise} \end{cases} \quad (1)$$

Nonetheless, it is necessary to address the following issues during the map matching.

1. Most links in a traffic model are curves, represented with a series of short straight lines. It is necessary to break the curves into short straight links to calculate the shortest vertical distances.
2. The CV data's coverage is usually broader than the road networks of studies (e.g., freeways). Therefore, it is necessary to filter out the CV waypoints that are not within the scope. To address this issue, the map-matching algorithm will check each waypoint's distance to its nearest link. If the shortest distance is longer than a threshold (e.g., 3-lane width), then this waypoint is considered out of scope and ignored (the crossed waypoints in Fig. 3). If a CV trip leaves the road network and then reenters from a downstream link later, it is considered two separate shorter CV trips.
3. Other than the vertical distances, the map-matching algorithm will also compare the headings of a waypoint and road links. It will further compare the headings of the adjacent waypoints to make sure a correct link is matched. These considerations are particularly important when vehicles are passing links near multideck interchanges, overpasses, and underpasses. (see Fig. 3).

For additional details on the customized map-matching algorithm readers are recommended to read Khadka et al. (2022).

Performance Evaluation of the Map-Matching Algorithm

To create a benchmark to evaluate the performance of the map-matching algorithm, 80 geofences were created on freeways in the DFW area. Within each geofence, we examined all passing connected vehicles in 24 h and aggregated the CV counts every 15 min. The CV trips in the geofences are manually verified and so they are

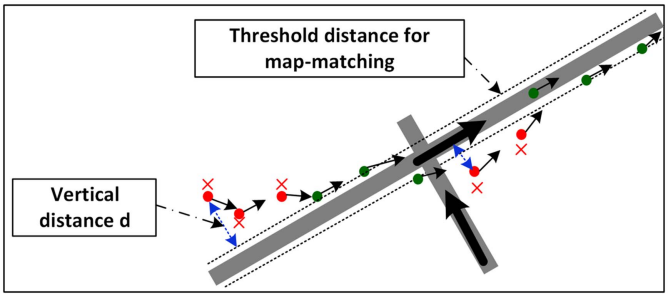

Fig. 3. Illustration of customized map-matching algorithm.

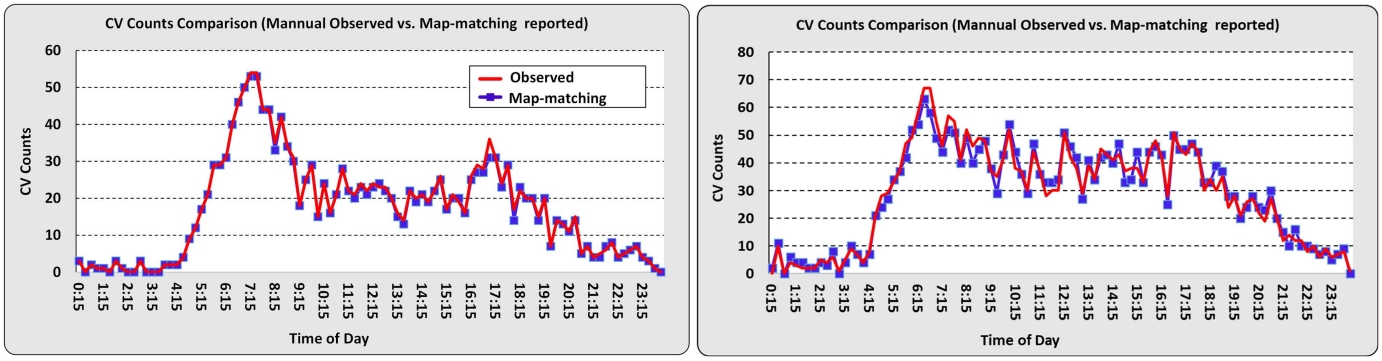

Fig. 4. CV counts comparison between manually observed and map-matching reported.

considered the ground-truth CV counts. The CV counts are then calculated again using the proposed map-matching algorithm on the same link segment and at the same time. The manually observed CV counts and the corresponding CV counts reported by the map-matching algorithm turn out almost the same. Fig. 4 demonstrates strong consistencies at two selected locations. Therefore, it is concluded that the proposed map-matching algorithm can accurately estimate the CV counts on all links in the DFW area.

Estimating Link Volumes with the CV Link Counts

Even though most metropolitan areas in the US have almost deployed infrastructure sensors on all freeways to continuously collect link counts and speeds, the coverage is still very small compared with the overall freeway miles. By contrast, the CV data almost cover all the roadways with a low penetration rate. Given the high TOD consistency between link traffic counts and CV counts, it is possible to develop an effective model(s) to estimate counts on all road links with the CV counts. In this paper, we explore three types of estimating techniques: linear regression model, deep neural network model, and random forest regression model.

Linear Regression Models

The linear regression (LR) model is widely used to build a system that takes an input vector $x \in \mathbb{R}^n$ to estimate and/or predict a scalar value of output $y \in \mathbb{R}$. The linearity is labeled because y is a linear function of in the input. Let $\hat{y}$ denote the value that the linear regression model should output, and the out can be expressed as

$$\hat{y} = w^{\intercal} x \tag{2}$$

where $\omega \in \mathbb{R}^n$ = vector of parameters in response to the input vector x.

The parameters ω are to control the performance of the linear regression model. Specifically, ω_i is the coefficient for the input feature x_i before all the features are added up to get the output scalar y. ω_i determines the importance of feature x_i and its contribution to the output. Therefore, ω is also called weight in other literature. If ω_i is zero, then the corresponding feature x_i does not contribute to the output. We first explored the performance of linear regression modeling for this application because the linear regression models are straightforward and easily interpreted. Thus, we should always prefer the linear regression model if its performance is satisfying. The model development is divided into three sequential steps: (1) data preparation, (2) model calibration, and (3) performance valuation.

Testing Data Preparation

The data source is the connected vehicle data, distributed by Wejo Data Service Inc. A data service company based in United Kingdom whose business is to process and clean networked vehicles' telematric data and re-distribute to the third-party organization. The data include the following relevant features for this context:

Inputs Features (x).

- Time-dependent connected vehicle counts at areas of interest: Each connected vehicle trip has a unique identifier and is composed of a series of waypoints. Each waypoint is composed of three elements: latitude, longitude, and time stamp denoted as (lat, lon, t). If a trip's waypoints are plotted on a map engine, we can see the overall path.

Note that the waypoints do not contain the road information and therefore it is necessary to match the waypoints to the corresponding road links and generate a time-dependent road link sequence for each trip (a.k.a., map matching).

The selected network is the freeway network in the Dallas Fort Worth, Texas, area, skimmed from the travel demand models maintained by the North Center Texas Council of Governments (NCTCOG). The time-dependent link sequences are further aggregated around the areas of interest with small geofences. In this application, the areas of interest are near the infrastructure sensors where 100% link counts and travel speeds are available.

- Time factors: Includes the time of day, day of the week, month of the year, and so forth. These data are included in the delivered CV data set.
- Vehicle's instantaneous speeds and headings: This information is derived from the waypoints. The headings are used to improve the accuracy of the map-matching algorithm and the speeds are used to estimate the link speeds.
- Road information at sensor locations: Includes the number of lanes and free-flow speeds.

Output Scalars (y).

- Infrastructure sensor counts via roadside sensors: Local agencies are continuously collecting vehicle counts and speeds at hundreds of locations. The local agencies have also verified the reported vehicle counts with recorded videos and the infrastructure sensor data are considered to capture 100% of vehicles passing those sensors. In total, we included 146 locations in the DFW area, each of which will report the vehicle counts and average speeds every 15 min. The total number of training records is 14,016.

While the CV counts are naturally suitable for the input vectors of regression models, we had to encode those narrative features (e.g., Monday) as model inputs. For instance, we numbered the day of the week from 1 (Monday) to 7 (Sunday).

Model Calibration

Different weights will change the LR model's performance. One popular way of measuring the performance is to compute the mean squared error (MSE) of the model on the test set, given m testing samples: $(x_i, y_i)\ (i = 1, 2, \ldots m)$

$$\text{MSE} = \frac{1}{m}\sum_i (\widehat{y_i} - y_i)_i^2 \quad \text{or} \quad \text{MSE} = \frac{1}{m}\|\hat{y} - y\|^2 \tag{3}$$

To minimize the mean square errors for the LR models, we can make the first-order differentiation concerning the weights w and make the gradient equal to 0 or $\nabla_w \text{MSE} = 0$

$$\nabla_w \frac{1}{m}\|\hat{y} - y\|^2 = 0 \tag{4}$$

$$\Rightarrow \nabla_w \frac{1}{m}\|\hat{y} - y\|^2 = 0 \tag{5}$$

$$\Rightarrow \nabla_w (Xw - y)^\intercal (Xw - y) = 0 \tag{6}$$

$$\Rightarrow \nabla_w (w^\intercal X^\intercal Xw - 2w^\intercal X^\intercal y + y^\intercal y) = 0 \tag{7}$$

$$\Rightarrow 2X^\intercal Xw - 2X^\intercal y = 0 \tag{8}$$

$$\Rightarrow w = \frac{X^\intercal y}{(X^\intercal X)} \tag{9}$$

Solving the weights for the LR model with Eqs. (4)–(9) is fast and the LR model can be further extended from the standard linear regression model to quadratic or cubic LR models by introducing the intercept as well as x^2 and x^3 in the form of $\hat{y} = b + w_1 x + w_2 x^2 + w_3 x^3$. The polynomial form of the LR model will increase the capacity of estimation but excessive high order may also create a possibility of overfitting.

Performance of Various LR Models

Various combinations of input features and extensions are tested, from over 10 input features to only CV counts, from standard LR models to extended polynomial LR models. We concluded that the CV data alone bring most of the contribution to the LR model based on the analysis of R^2 and covariance matrix. One of the reasons for this phenomenon is that the data sets are time-dependent and have automatically reflected the time factors. The diversity of infrastructure features is also limited because we focus on major freeways where free-flow speeds and the number of lanes are similar. Among all the data records, 80% were used for model calibration and the remaining 20% were used to examine the performance of LR models. Fig. 5 shows two calibrated LR models. R^2 are calculated as $R^2 = \dfrac{1 - \|\hat{\boldsymbol{y}} - \boldsymbol{y}^2\|}{\|\boldsymbol{y} - \bar{y}^2\|}$. The MAE is calculated as $\text{MAE} = \frac{1}{n}\sum_{n=1}^{n} |Y_i - \hat{Y}_i|$, where N is the number of data points, Y_i is the observed output values, and $\hat{Y}_i$ = predicted value.

In summary, the LR models can overly estimate the time of day trend of link counts. At a few locations, the estimation errors are up to 200% with truncations. This may generate misleading information in practice. Since the whole model in linear regression is given a single weight, the model may be unable to accurately predict how traffic flows during peak or off-peak hours. Due to this issue, a single-weight model, such as linear regression, may not be able to support the estimation well. Additionally, it seems ineffective to capture the other input features contained in the CV data, even after those input features are transformed to have better mathematical characteristics. As such, we decided to further explore the potential of deep learning models to further improve the accuracy of the link count estimation. Note that we do not consider the penalty of large weight values because the final LR models only have one input feature, the CV counts.

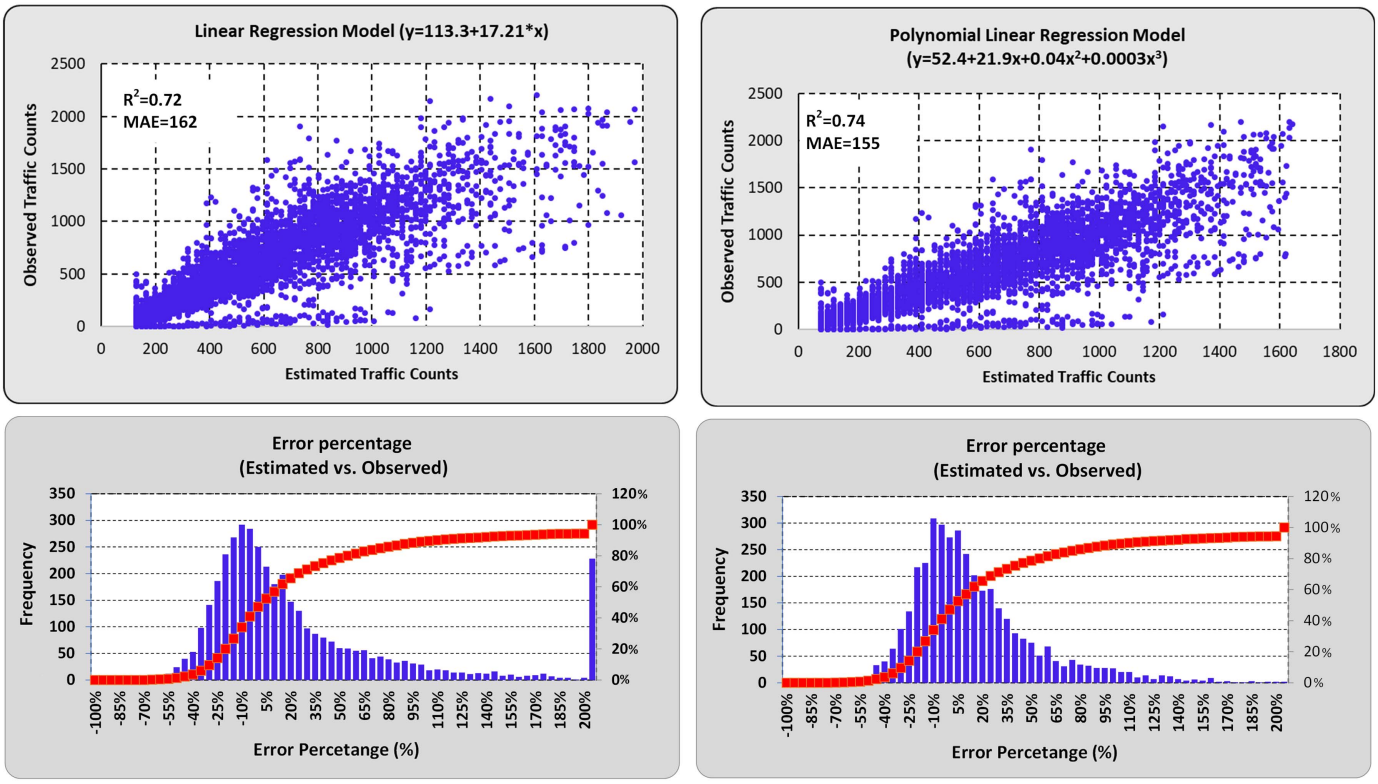

Fig. 5. Performance of the selected linear regression models.

J. Transp. Eng., Part A: Systems, 2023, 149(4): 04023015

Deep Neural Network Models

The objective of using machine learning (ML) models is to further improve the accuracy of link count estimation. In particular, we wish more features other than the CV counts can contribute to the estimation accuracy. We explored developing a deep neural network (DNN) model to approximate the target function with an approximation $y = f(x; w)$. The DNN model's performance is sensitive to the parameters w. To define a DNN model, we need to consider the following elements.

Cost Function

A training data set, which is considered ground truth, is used to evaluate the DNN model with a general cost function, such as the MSE. The cost function must be sensitive to the w values. In most cases, minimizing the cost function value can be achieved by seeking where the gradient of the cost is zero. For estimation problems, the MSE is commonly selected as the cost function.

Output Units

We used linear output units. Given features h, a layer of linear output units produces a vector in the form of $\hat{y} = W^{\intercal}h + b$. In other literature, it is also written as

$$a_i^{(l+1)} = f(w_i^{(l+1)}\boldsymbol{a}^{(l)} + b_i^{(l+1)}) \tag{10}$$

where $a_i^{(l+1)}$ = output from the ith unit (also known as activation function or neuron) in layer $l + 1$; $\boldsymbol{a}^{(l)}$ = vector of unit outputs from the last layer l; and $b_i^{(l+1)}$ = bias associated with each unit in each layer.

For large DNN models, it is necessary to avoid overfitting by dropping out certain intermediate neurons from each step as well as the corresponding weights. This idea is inspired by Hinton et al. (2012). Its roots are in the stochastic optimization to leave a local minimum; so, Eq. (10) can be further modified to

$$a_i^{(l+1)} = f(w_i^{(l+1)}r^{(l)}a^{(l)} + b_i^{(l+1)}) \tag{11}$$

where $r^{(l)} \sim Bernoulli(p)$; and p = dropping rate.

Hidden Layers

Design of hidden layers primarily distinguish one DNN model from another. It includes the selection of activation functions, connection between layers, and the number of neurons in each layer. The hidden layers can be described to compute an affine transformation $z = W^{\intercal}\boldsymbol{x} + b$ and then applying an element-wise activation function $g(z)$.

In this paper, we adopt three fully connected hidden layers. Each hidden layer contains 50 neurons with the rectified linear activation function (ReLU). The ReLU is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero. The ReLU function and its derivative are shown in Eq. (12). The ReLU function gains popularity because of its computing efficiency in practice. The neuron is automatically deactivated and ignored once the input value becomes nonpositive

$$R(z) = \begin{cases} z & z \geq 0 \\ 0 & z \leq 0 \end{cases} \qquad R'(z) = \begin{cases} 1 & z > 0 \\ 0 & z < 0 \end{cases} \tag{12}$$

Training Algorithm

We adopt the adaptive moment estimation (Adam) algorithm, a stochastic gradient-based optimization that can handle high-dimensional search space in nonconvex optimization problems (Kingma and Ba 2014).

In summary, the designed DNN network is illustrated in Fig. 6.

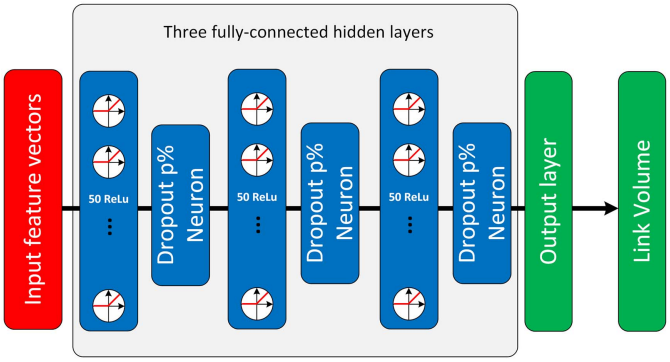

Fig. 6. Structure of the proposed DNN model.

Data Preparation for the DNN Model

Inputs Features (x).

- CV vehicle counts according to the waypoints: Such data covers the entire DFW area. They are reduced to 90 small areas where the infrastructure sensors were installed. The penetration rate of the connected vehicle counts is around 3%–5% of the total vehicle counts by the infrastructure sensors. The CV counts are highly correlated with the infrastructure traffic counts.
- Directions: Using the same data set, the direction of the vehicle can be identified. Direction includes east (E), west (W), north (N), and south (S). With the *one-hot encoding* method, all four directions were converted into four different input features suitable for the DNN model.
- Time factors: Time of day and day of the week have a substantial impact on traffic counts. Therefore, the temporal traffic patterns such as time of day (1–24 h), day of the week [1 (Monday)–7 (Sunday)], and month of the year [1 (January)–12 (December)] are considered for each data point. Since these features are cyclic numbers, we use the one-hot encoding method again to encode. All these time-dependent data are converted into six input features.
- Vehicles' instantaneous speeds: The vehicle's average speed was calculated according to the reported speed samples. Only one feature is used to estimate observed 15 min traffic counts, the average speed.
- Road information at sensor locations: includes the number of lanes and free-flow speeds.

Output Labels (y). Infrastructure sensor counts via the roadside sensors: The local agency provided us with 100% traffic counts at 90 locations, which are considered the ground truth for training and evaluating the DNN model. The DNN model is to output the estimated 15-min vehicle counts at selected locations and then compare them with the ground-truth traffic counts. Note that we only focus on the freeway links because all the infrastructure sensors were deployed on freeways. There are 14,017 training data records and each record contains 12 input features ($\boldsymbol{x}$) and one observed link count (y); 80% of records are used for training and the remaining 20% of records are used for testing. Each point corresponds to the 15-min volume measurements taken at 90 locations, each of which counts both directions of traffic.

One-Hot Encoding Method for Classified Data. Variables containing limited values are referred to as categorical data. For example, a direction variable could have the values north, east, south, and west. Categorical input features must be mapped to integers for the DNN model. To ensure the cost function and optimization algorithm are sensitive to the changes to categorical features, it is necessary to transform them with the one-hot encoding method.

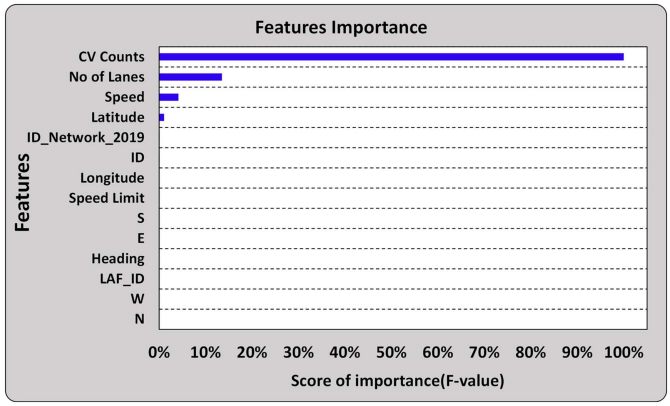

Fig. 7. Feature importance analysis.

This method will transform each categorical value into a new categorical column and give a binary value of 1 or 0. A binary vector is used to represent each integer value. The index is designated with a 1 and all the values are 0. For example, for a direction feature, we can define a quadlet to represent north (1,0,0,0), east (0,1,0,0), south (0,0,1,0), and west (0,0,0,1). As a result, one categorical feature will become four features after the one-hot encoding. One-hot encoding may significantly increase the number of features.

Encoding the Cyclic Data Based on Trigonometry. Day of the month, day of the week, day of the year, and other time-based characteristics have a cyclic nature and various feature values. The day of month feature of one-hot encoding yields a 30-dimensionality vector, while the day of year yields a 366-dimensional vector. One-hot encoding of these attributes is inefficient because it may result in a curse of dimensionality. To address this issue, we encode the cyclic temporal features with the sin and cosine values of the features using the basic concept of trigonometry. Instead of employing one-hot encoding to create a 7-dimensionality feature vector, a 2-dimensional transformed feature vector will now be used to represent the full feature.

Features Importance Analysis. Even if the CV data set contains many properties, not all of the features are important. Some data might not be related to the result that we are attempting to obtain. Therefore, it is necessary to investigate the importance of each feature. The select KBest technique for feature selection (James et al. 2013) was utilized to investigate the feature importance. For this regression model, the F-value between label/feature was calculated using the f regression scoring function. The feature importance study for all the variables shown in our data set is represented in Fig. 7, clearly showing that only CV counts, the number of lanes, measured speed, latitude, and longitude are important. As a result, the remaining features are ignored in the DNN model because they are unlikely to affect the estimating performance.

Performance of the DNN Model

The DNN model is first trained with the training data set, 80% of which are used for training, and 20% are used for testing. Fig. 8 shows the loss function value and MAE value changes over epochs. We can see there is no significant overfitting problem because the training data's loss function and MAE values do not increase over iterations. Fig. 9 shows the estimated link volumes as opposed to the observed link volumes of the testing data sets and the distribution of the estimation error rate $\left(\frac{\text{estimated} - \text{observed}}{\text{observed}}\%\right)$.

Random Forest Regression Model

We also explore the popular random forest (RF) regression model to estimate the link counts. It is a nonparametric machine learning algorithm suitable for the classification and regression of high-dimension data. The RF algorithm is due to Breiman (2001). The standard RF regression model can be estimated as follows:

- Assume the training data set contains M records (rows). Each record contains N input features and one output ($N+1$ columns). Let m denote the number of sample records ($m \le M$), n denote the number of features ($n \le N$) to build subtrees; K denote the number of independent trees; and J represent the maximum depth (branches) of each tree. The branching efforts stop if a leaf node only contains one sample and, therefore, the number of branches of each tree $j \le J$. The estimated value of each node will be the average output scalar.
- Using the bootstrap sampling technique (i.e., putting back selected samples and features each time for resampling), we use m samples (rows) and n features to generate K independent decision trees with j branches. When branching, the RF algorithm will choose one of the remaining features with an appropriate threshold to generate the maximal information gain or the maximal reduction of uncertainty. The information gain is calculated as

$$E_{\text{before}} - E_{\text{after}} = \left(-\left(\sum_i^C p_i \log_2 p_i\right)\right) - \left(-\left(\sum_i^{C1} p_i \log_2 p_i\right) - \left(\sum_i^{C2} p_i \log_2 p_i\right)\right) \tag{13}$$

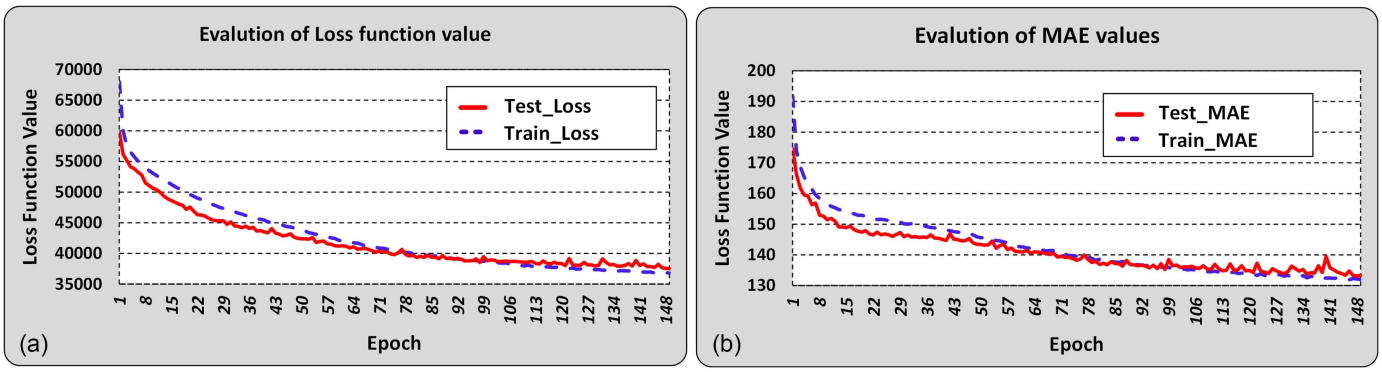

Fig. 8. Evolution of DNN model training: (a) evaluation of loss function value; and (b) evaluation of MAE values.

J. Transp. Eng., Part A: Systems, 2023, 149(4): 04023015

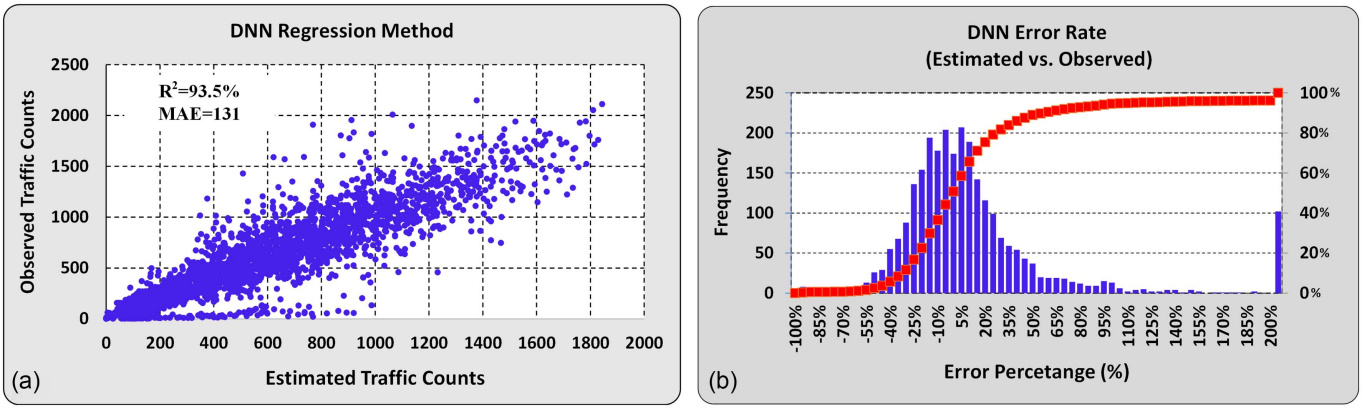

Fig. 9. Performance of the selected linear regression models: (a) DNN regression method; and (b) DNN error rate.

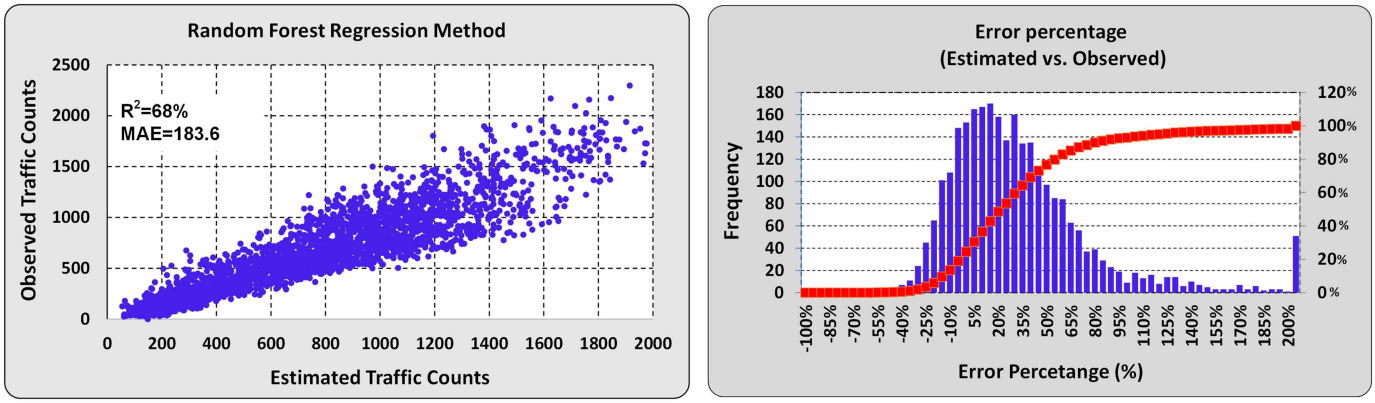

Fig. 10. Performance of random forest regression method.

where C = all the samples in the parent node; and C_1, C_2 = the samples after splitting. So $C = C_1 + C_2$.

- For a new test data sample, it will go through all the independent decision trees until it reaches a leaf node. Then the estimated output scalar of that decision tree will be the average output values of the training data samples in that node. Each decision tree will generate an estimated output and the final estimated output for a test data sample will be the average of all decision trees' output values.

In this experiment, we use the same data set for the DNN estimation model. The training and testing data sets are 80% and 20%, respectively. After evaluating a different number of trees, we decided to generate 50 independent decision trees for the regression model, the minimal number of samples in the node for further splitting is 2.

Fig. 10 shows the performance of the RF regression model on the testing data set. We can tell that the RF regression method can provide decent link count estimation. However, it has a lower R^2 value than the LR regression models. In addition, the RF regression model cannot estimate the output values larger than those in the training data set while the LR regression model can well extrapolate. Therefore, we consider the RF regression model inferior to the aforementioned LR regression models.

Discussion

According to the R^2 and MAE in each model, the DNN model seems to outperform the LR and RF models. Table 2 displays all of the model's results in tabulated form. Table 2 makes it quite evident that DNN outperforms all other models in terms of performance. Among all the models, the DNN model has the lowest MAE and highest R-squared value.

Table 2. Comparison of different models based on MAE and R-squared values

Models	MAE	R-squared value
Linear regression	162	0.72
Polynomial linear regression	155	0.74
Random forest	184	0.68
DNN	131	0.93

The DNN model's performance is expected to be further improved if more infrastructure data (e.g., link volumes) are available. Therefore, we prefer the DNN model over the other two models. Nonetheless, it is noticed that all three models may generate outlier volumes whose error rates are more than 200%. Large error rates could be caused either by the inefficiency of the DNN model or just by the malfunctioned infrastructure sensors. Therefore, it will be complicated to solve only through modifying the DNN model. It is recommended to provide an upper bound for volume estimation. For instance, we assumed a 2% penetration rate is the minimal penetration rate and 6% is the maximal penetration rate according to Fig. 2(a). Then the upper bound of link volumes can be set as the CV counts divided by 2%. Any estimated link volumes beyond the

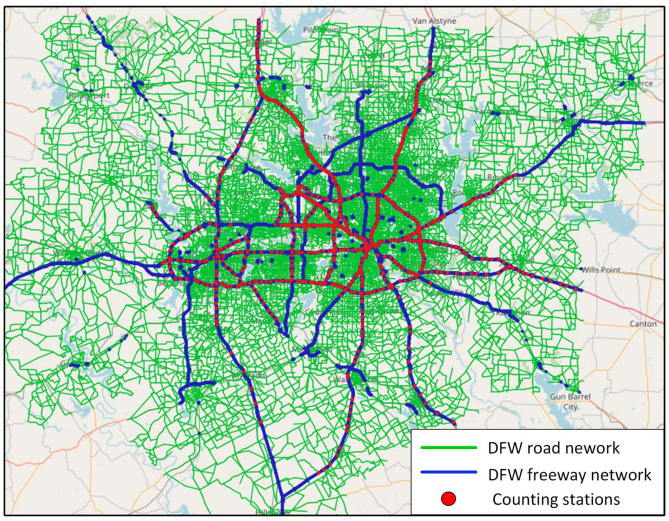

Fig. 11. Dallas Fort Worth Road network and counting stations. (Data from NCTCOG, n.d.)

upper bound value should be either truncated as the upper bound value or ignored.

Case Study: Estimate Time-Dependent Link Volumes with the Connected Vehicle Data in the Dallas Fort Worth Area, Texas

After comparing the three estimation models in the section "Estimating Link Volumes with CV Link Counts," we conclude that the proposed DNN model is the best method to estimate the link traffic counts with the connected vehicle data. Since the CV data are ubiquitous, it is possible to estimate full-network link volumes using a well-trained DNN model and the corresponding connected vehicle counts.

Regional Freeway Link Volumes Estimation Using the DNN Model

The freeway network in the DFW area contains 5,053 freeway segments and ramps. To prepare the training data sets for the DNN model, we collected link counts from 1,063 freeway locations during 20 workdays in September 2021 (See Fig. 11 for infrastructure detector locations). The CV data were first matched to those locations to generate the corresponding CV counts. The link counts and CV counts were archived every 15 min; each location-day (one location per day) data contain 96 data records, including the traffic counts per 15 min as well as the number of lanes and speed limits. After removing outliers, the total number of training data records was 1,913,856. The input features include all those in the section "Deep Neural Network Models" plus three new features: the number of lanes, speed limits, and counting station IDs. The number of lanes and speed limits are geographic features, and the counting station ID is to facilitate the DNN model to distinguish the training data among the counting stations.

The estimating procedure is summarized in Fig. 12. We put aside the link counts on September 29 as independent testing data to validate the performance of the DNN model. The testing data sets contain the TOD link volumes, which are used as the ground truth to compare with the corresponding estimated TOD link volumes according to the CV counts.

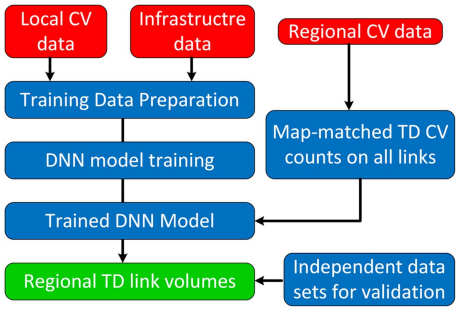

Fig. 12. Framework of TOD link volume estimation with the regional CV data. TD = travel demand.

24-h Estimated Link Volumes Analysis and Validation

To predict a link volume, we first calculate the CV counts on that link with the map-matching algorithm. The CV count, in conjunction with other spatiotemporal features, is input into the calibrated DNN model. The DNN model will then output the estimated value of link volume. At each location, 96 time-dependent link volumes were generated for each day (15-min interval).

The 15-min link counts collected via the infrastructure sensors at 1,063 locations on September 29, 2021, were used as the ground truth to validate the estimated link counts by the DNN model. Fig. 13 reveals strong consistencies between the estimated link volumes and the actual link volumes. According to Fig. 16, about 69% of estimated link volumes are within the 20% range of error rate e, defined as

$$e = \left(\frac{\text{estimated counts} - \text{observed dounts}}{\text{observed counts}}\right)\% \tag{14}$$

In the preliminary experiments, we gradually increased the number of infrastructure sensor locations and the number of days to examine the impact of training data size on the model's performance. It was recognized that the DNN model's performance steadily improved with the increase in training data set size. As such, it is anticipated the model's performance would be further improved if more locations of infrastructure sensors over more days are used for the training data set. Fig. 14 shows a comparison of time of day link volumes (observed versus estimated) at four randomly selected locations. They all show strong consistency.

Peak-Hour Estimated Link Volumes Analysis and Validation

In practice, transportation planners often focus on the link volumes during peak hours. Therefore, we further conducted a time of day model performance analysis. The error rates of the testing data set (September 29, 2021) were averaged every 15 min. Fig. 15(a) shows that the error rates were low (less than 20%) during the daytime but were high during the off-peak hours at late night and early morning. Fig. 15(b) is the time of day average 15-min (observed) link counts from the testing data set. According to Eq. (14), the error rate will tend to be large if the denominator (observed link counts) is small, bringing bias to the overall performance evaluation.

Fig. 16 shows the error rate distribution during morning and evening peak hours compared with the 24-h error rate distribution. We can tell the model's performance is better during peak hours. Specifically, 76% of error rate samples are within the 20% error range during the morning peak hours and 78% of error rate samples are within the 20% error range during the evening peak hours.

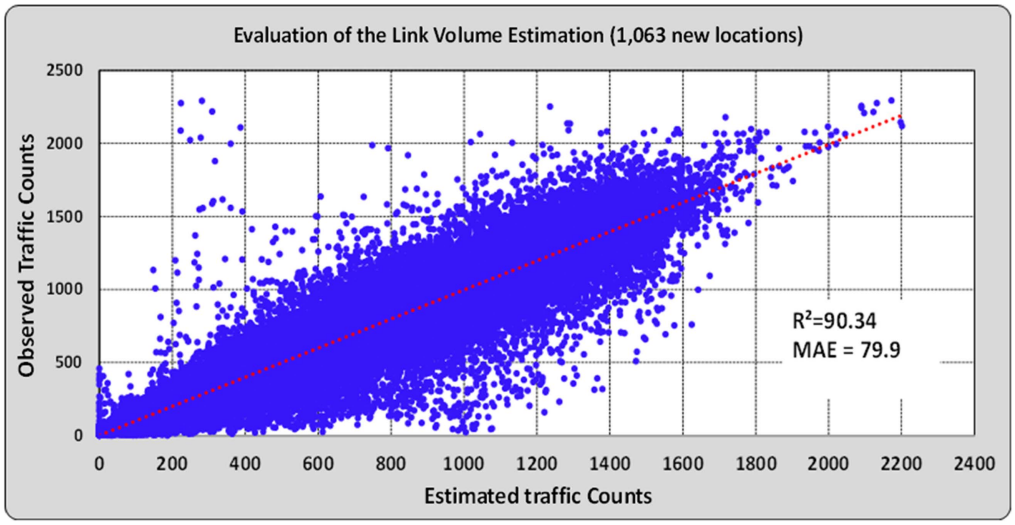

Fig. 13. Performance of link volumes estimation at four new locations.

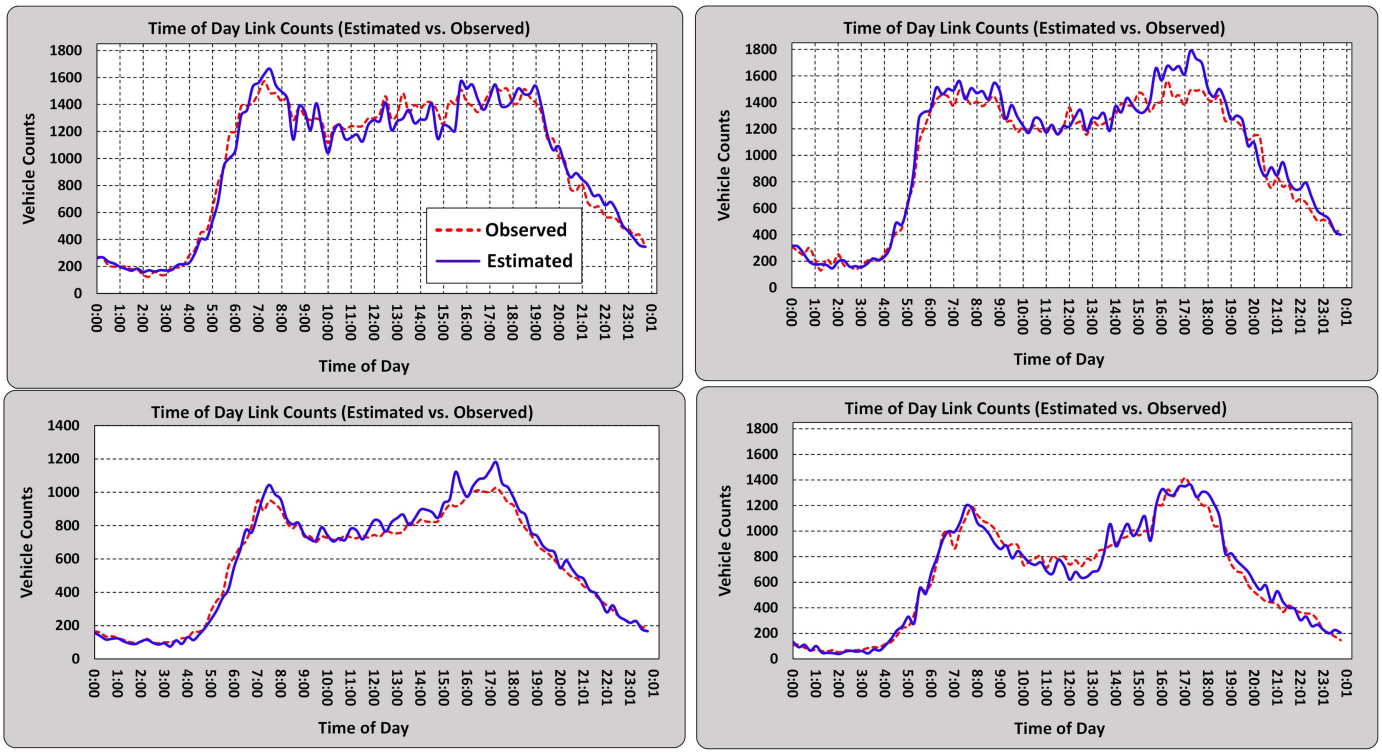

Fig. 14. Comparison of time of day link volumes at four new locations.

By contrast, only 69% of 24-h error rates are within the 20% range of error rates.

Discussion

- It is noticed that including the geolocations (latitude, longitude) of infrastructure sensors does not improve the performance significantly. The rationale is that two adjacent infrastructure sensors are not necessarily consistent in terms of traffic volumes (e.g., freeway mainlines and the frontages roads). Our preliminary experiment results also show that including the geolocation information does not bring benefits.
- It is noticed that the configuration of input features is critical to ensuring the performance of the DNN model. Human experiences are important to designing the input features of the training data set. We compared two options of input features: (1) including the number of lanes as one input feature and using the total link count as the output scalar; and (2) not including the number of lanes and using the link counts per lane as the output scalar. The experiment results reveal that the DNN model under the second option significantly outperforms the DNN model under the first option on the independent testing data set.
- For map matching, it is necessary to break long and/or curvy links into short straight-line links to calculate the vertical distances from waypoints to links. Otherwise, it would be hard

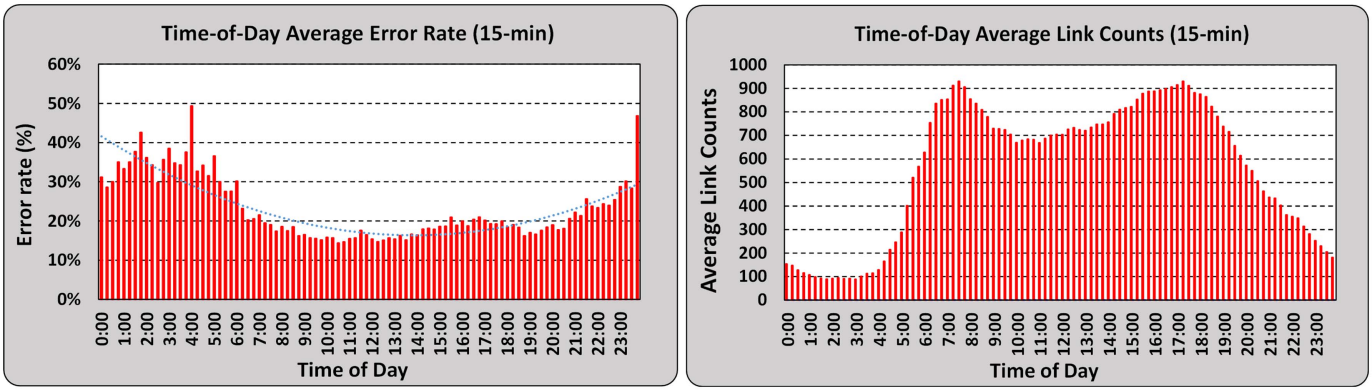

Fig. 15. Time of day model performance analysis.

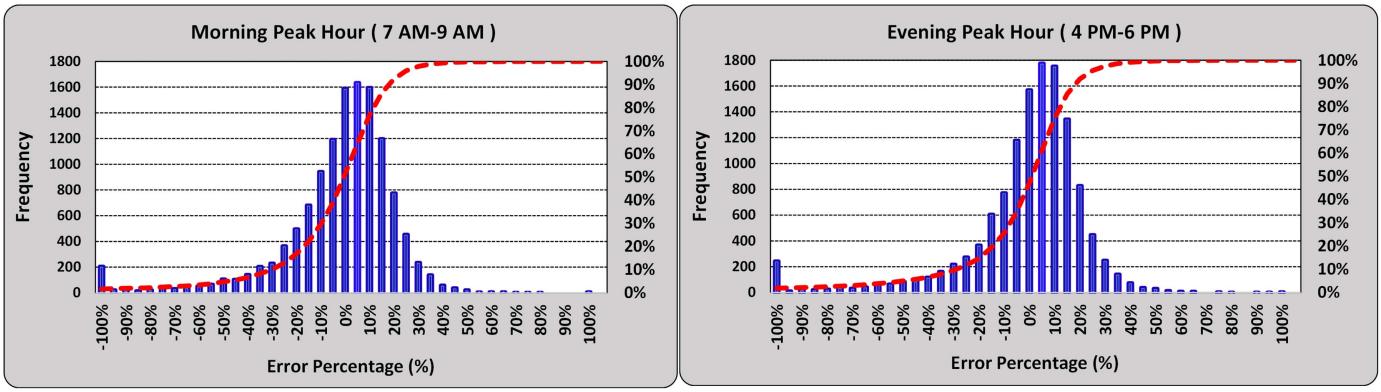

Fig. 16. Peak-hour error rate distribution.

to precisely match CV counts to links in which vehicles merge or diverge.

- A few days of link volumes at some locations were found to have large error rates in the testing data set. To mitigate this issue, we can set upper and lower bounds of link volumes as the connected vehicle counts divided by 2% and 6% penetration rates, respectively.

Conclusions and Future Work

In this paper, we present a new framework for regional travel demand estimation, powered by the merging of connected vehicle data and machine learning techniques. By comparing three similar regression methods, we conclude that the DNN model can best estimate the link volumes according to the captured link CV counts. We also present a customize map-matching algorithm to map each CV trip to the freeway network. This method enables us to estimate the time-dependent CV counts on all road links. Using the CV data in the DFW area for September 2021 and the regional freeway network, we conducted a case study to estimate the regional travel demand on freeways and then validate with independent testing data set. The results are promising.

The proposed framework provides an alternative, data-driven approach for regional travel demand forecast. Compared with the classic four-step travel demand forecast, this new method contains few assumptions about traveling behaviors. It also exploits the potential of the CV data set in transportation planning and travel demand forecast.

This paper focuses on (historical) traffic volume estimation. In the future, we plan to explore the short-term prediction of regional travel demand (e.g., the next 15 min) based on a longer period of CV data set because of its importance to congestion management and reduction of air pollution.

Data Availability Statement

Some or all data, models, or code used during the study were provided by a third party. Direct requests for these materials may be made to the provider as indicated in the Acknowledgments.

Acknowledgments

This research is supported by the project "Embracing emerging traffic big data (connected vehicle data) in smart city applications to improve transportation systems efficiency, safety, and equity," sponsored by Center for Transportation Equity, Decisions, and Dollars (CTEDD), a USDOT university research center at the University of Texas at Arlington. It is supported by the University Partnership Program at the North Central Texas Council of Governments (NCTCOG). The connected vehicle data were distributed by Wejo Data Service. The authors also thank Mr. Arash Mirzaei, Dr. Hong Zheng, and Dr. Gopindra Nair of NCTCOG for their suggestions and comments. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the official views or policies of the aforementioned organizations, nor do the contents constitute a standard, specification, or regulation of these organizations.

References

Antoniou, C., H. N. Koutsopoulos, and G. Yannis. 2013. "Dynamic data-driven local traffic state estimation and prediction." *Transp. Res. Part C Emerging Technol.* 34 (Sep): 89–107. https://doi.org/10.1016/j.trc.2013.05.012.

Breiman, L. 2001. "Random forests." *Mach. Learn.* 45 (1): 5–32. https://doi.org/10.1023/A:1010933404324.

Chen, Q., Y. Song, and J. Zhao. 2020. "Short-term traffic flow prediction based on improved wavelet neural network." *Neural Comput. Appl.* 33 (14): 8181–8190. https://doi.org/10.1007/s00521-020-04932-5.

Dietterich, T. G. 2000. "Ensemble methods in machine learning." In *Proc., Multiple Classifier Systems*, 1–15. Berlin: Springer.

Ding, Q. Y., X. F. Wang, X. Y. Zhang, and Z. Q. Sun. 2011. "Forecasting traffic volume with space-time ARIMA model." *Adv. Mater. Res.* 156–157 (Oct): 979–983. https://doi.org/10.4028/www.scientific.net/AMR.156-157.979.

Duan, Y., Y. Lv, Y.-L. Liu, and F.-Y. Wang. 2016. "An efficient realization of deep learning for traffic data imputation." *Transp. Res. Part C Emerging Technol.* 72 (Nov): 168–181. https://doi.org/10.1016/j.trc.2016.09.015.

Guo, J., W. Huang, and B. M. Williams. 2014. "Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification." *Transp. Res. Part C Emerging Technol.* 43 (Jun): 50–64. https://doi.org/10.1016/j.trc.2014.02.006.

Hamed, M. M., H. R. Al-Masaeid, and Z. M. B. Said. 1995. "Short-term prediction of traffic volume in urban arterials." *J. Transp. Eng.* 121 (3): 249–254. https://doi.org/10.1061/(ASCE)0733-947X(1995)121:3(249).

Heshami, S., and L. Kattan. 2021. "A queue length estimation and prediction model for long freeway off-ramps." *J. Intell. Transp. Syst.* 25 (1): 122–134. https://doi.org/10.1080/15472450.2020.1846125.

Hinton, G., et al. 2012. "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups." *IEEE Signal Process Mag.* 29 (6): 82–97. https://doi.org/10.1109/MSP.2012.2205597.

James, G., D. Witten, T. Hastie, and R. Tibshirani. 2013. *An introduction to statistical learning*. New York: Springer.

Khadka, S., P. T. Li, and Q. Wang. 2022. "Developing novel performance measures for traffic congestion management and operational planning based on connected vehicle data." *J. Urban Plann. Dev.* 148 (2): 04022016. https://doi.org/10.1061/(ASCE)UP.1943-5444.0000835.

Khan, S. M., K. C. Dey, and M. Chowdhury. 2017. "Real-time traffic state estimation with connected vehicles." *IEEE Trans. Intell. Transp. Syst.* 18 (7): 1687–1699. https://doi.org/10.1109/TITS.2017.2658664.

Kingma, D. P., and J. Ba. 2014. "Adam: A method for stochastic optimization." Preprint, submitted July 25, 2019. http://arxiv.org/abs/1412.6980.

Li, J., J. Boonaert, A. Doniec, and G. Lozenguez. 2021. "Multi-models machine learning methods for traffic flow estimation from floating car data." *Transp. Res. Part C Emerging Technol.* 132 (Oct): 103389. https://doi.org/10.1016/j.trc.2021.103389.

Liu, Y., Z. Liu, H. L. Vu, and C. Lyu. 2019. "A spatio-temporal ensemble method for large-scale traffic state prediction." *Comput.-Aided Civ. Infrastruct. Eng.* 35 (1): 26–44. https://doi.org/10.1111/mice.12459.

Lu, S., Q. Zhang, G. Chen, and D. Seng. 2021. "A combined method for short-term traffic flow prediction based on recurrent neural network." *Alexandria Eng. J.* 60 (1): 87–94. https://doi.org/10.1016/j.aej.2020.06.008.

Lv, Y., Y. Duan, W. Kang, Z. Li, and F.-Y. Wang. 2014. "Traffic flow prediction with big data: A deep learning approach." *IEEE Trans. Intell. Transp. Syst.* 16 (2): 865–873. https://doi.org/10.1109/TITS.2014.2345663.

NCTCOG (North Central Texas Council of Governments). n.d. "Voluntary association to assist local governments in planning for common needs and cooperating for mutual benefit for sound regional development." Accessed December 5, 2022. https://www.nctcog.org/about-us.

Ni, D., and J. D. Leonard. 2005. "Markov Chain Monte Carlo multiple imputation using bayesian networks for incomplete intelligent transportation systems data." *Transp. Res. Rec.* 1935 (1): 57–67. https://doi.org/10.1177/0361198105193500107.

Okutani, I., and Y. J. Stephanedes. 1984. "Dynamic prediction of traffic volume through Kalman filtering theory." *Transp. Res. Part B Methodol.* 18 (1): 1–11. https://doi.org/10.1016/0191-2615(84)90002-X.

Polson, N., and V. Sokolov. 2018. "Bayesian particle tracking of traffic flows." *IEEE Trans. Intell. Transp. Syst.* 19 (2): 345–356. https://doi.org/10.1109/TITS.2017.2650947.

Pun, L., P. Zhao, and X. Liu. 2019. "A Multiple regression approach for traffic flow estimation." *IEEE Access* 7 (Mar): 35998–36009. https://doi.org/10.1109/ACCESS.2019.2904645.

Sekuła, P., N. Marković, Z. Vander Laan, and K. F. Sadabadi. 2018. "Estimating historical hourly traffic volumes via machine learning and vehicle probe data: A Maryland case study." *Transp. Res. Part C Emerging Technol.* 97 (Dec): 147–158. https://doi.org/10.1016/j.trc.2018.10.012.

Tak, S., S. Woo, and H. Yeo. 2016. "Data-driven imputation method for traffic data in sectional units of road links." *IEEE Trans. Intell. Transp. Syst.* 17 (6): 1762–1771. https://doi.org/10.1109/TITS.2016.2530312.

Wang, Y., and M. Papageorgiou. 2005. "Real-time freeway traffic state estimation based on extended Kalman filter: A general approach." *Transp. Res. Part B Methodol.* 39 (2): 141–167. https://doi.org/10.1016/j.trb.2004.03.003.

Xu, D., C. Wei, P. Peng, Q. Xuan, and H. Guo. 2020. "GE-GAN: A novel deep learning framework for road traffic state estimation." *Transp. Res. Part C Emerging Technol.* 117 (Aug): 102635. https://doi.org/10.1016/j.trc.2020.102635.

Yaghoubi, F., A. Catovic, A. Gusmao, J. Pieczkowski, and P. Boros. 2021. "Traffic flow estimation using LTE radio frequency counters and machine learning." Preprint, submitted January 22, 2021. https://arxiv.org/abs/2101.09143.

Yin, W., P. Murray-Tuite, and H. Rakha. 2012. "Imputing erroneous data of single-station loop detectors for nonincident conditions: Comparison between temporal and spatial methods." *J. Intell. Transp. Syst.* 16 (3): 159–176. https://doi.org/10.1080/15472450.2012.694788.

Zahedian, S., P. Sekuła, A. Nohekhan, and Z. Vander Laan. 2020. "Estimating hourly traffic volumes using artificial neural network with additional inputs from automatic traffic recorders." *Transp. Res. Rec.* 2674 (3): 272–282. https://doi.org/10.1177/0361198120910737.