Deep tree neural network for multiple-time-step prediction of short-term speed and confidence estimation

Yanyun Tao1 | Xiang Wang1 | Jianying Zheng1 | Wenjuan E1 | Po Zhao1 | Shiwei Meng2

1 School of Rail Transportation, Soochow University, No.8 Jixue Road, Suzhou, Jiangsu 215000, People’s Republic of China
2 Fu Foundation School of Engineering & Applied Science, Columbia University, 500 W 120th St#510, New York 10027, United States

Abstract

To solve the multiple-time-step prediction of traffic speed and confidence for segment types of expressway, a deep tree neural network (DTNN) with multitask learning is proposed. DTNN contains a classification network, regression networks and a confidence network. These sub backbone networks accomplish the tasks of distinguishing segment types, fitting the speed of segments and the confidence estimation on the predicted speed, respectively. Through multitask learning, the sub networks in DTNN share feature representation and complement each other. To further improve the accuracy of speed prediction on congestion, the mean absolute percentage error loss function (MAPE-loss) is applied in DTNN. It makes the learning and extracted features biased to low speed samples. The traffic speed dataset of the Shanghai Expressway is used to test the DTNN and 12 comparison methods. Results show that the proposed DTNN with MAPE-loss can efficiently improve the predictive accuracy of low-speed samples over the other methods. The trained DTNN also gave highly accurate low speed prediction on the dataset of Suzhou expressway. In addition, the smallest reduction in the R-squared value from the training stage to the testing stage illustrates the best generalization of DTNN model.

1 | INTRODUCTION

As a way to relieve traffic congestion and improve the efficiency of expressway networks, intelligent transportation systems and advanced traffic management systems have been actively developed. A necessary component of such systems is short-term traffic speed prediction, which aims to forecast and evaluate traffic patterns in a short period. If an accurate short-term prediction of congested segments could be derived, then corresponding traffic management could ease congestion and thereby improve the operation efficiency of expressway networks. The early research about short-term traffic speed prediction mainly focused on time series models and statistical models, including the autoregressive integrated moving average (ARIMA) model [1], exponential smoothing model [2], Kalman filter model [3] and Bayesian model [4]. These models can predict stable traffic conditions by using small traffic datasets, but their prediction tends to be inferior given non-linear variations with large-scale and high-dimensional traffic condition data. As traffic data volume increases, short-term prediction based on data-driven approaches offers a potential solution. Many experts have tried implementing machine learning methods to capture non-linear spatiotemporal relationships for traffic state prediction; these methods include k-nearest neighbour [5,6], support vector machine [7], artificial neural network [8], decision tree [9] and Gaussian maximum likelihood [10].

Meanwhile, deep learning shows promise in traffic condition prediction with the advancement of computational power. Numerous studies have applied neural network-based models, such as deep belief network, convolutional neural network (CNN), recurrent neural network (RNN), long short-term memory (LSTM) and gated recurrent units GRUs [11–13], to capture the spatiotemporal correlations of traffic conditions and obtain favourable prediction accuracy. Gu Y. et al. [14]...
combined a Bayesian model with deep learning for traffic flow prediction. Zhao Z. et al. [15] established a cascaded LSTM network that combines the interactions among road networks in the time and spatial domains. Yu H. et al. [16] proposed a spatiotemporal recurrent convolutional network for traffic forecasting; the proposed method inherits the advantages of deep CNNs and LSTM neural networks. Guo S. et al. [17] proposed a novel attention-based spatiotemporal graph convolutional network that considers recent, daily–periodic and weekly–periodic dependencies. Fukuda S. et al. [18] focused on traffic prediction under incident conditions through a graph convolutional RNN and reduced the effects of unusual conditions on forecasts. Chen X. M. et al. [19] integrated gradient boosting regression trees and the least absolute shrinkage and selection operator to present a multimodel ensemble for short-term traffic flow prediction under normal and abnormal conditions.

However, most existing research has focused on single-time-step prediction or one particular segment type (e.g., normal segment, merge segment, diverge segment, weaving segment and ramp). Another issue is that deep learning can always generate a prediction regardless of the divergence between current situations and training datasets. In most cases, prediction confidence cannot be achieved at the current time. By contrast, multi-time-step prediction for all expressway segment types is serviceable but sophisticated. Such prediction also helps traffic managers to not only predict traffic conditions but also achieve prediction confidence. To address existing issues, we aim to propose a novel approach that is applicable to all expressway segment types. This approach is designed to predict multi-time-step speeds and achieve prediction confidence. The deep tree neural network (DTNN) with multitask learning (MTL) is chosen to achieve the aforementioned objectives. MTL. [20–22] is an approach to learning several related problems simultaneously through a shared representation. Continuous short-term predictions can be seen as multiple related tasks.

Fortunately, several attempts have been made to apply MTL to short-term prediction. Jia H. et al. [23] established an attention-based deep spatiotemporal network with MTL, which can forecast steady trends and sudden changes in passenger flow. Kuang L. et al. [24] constructed an MTL component and captured the correlation between taxi pick-up and drop-off tasks with 3D ResNet for accurate prediction. Luo H. et al. [25] proposed an MTDL model to predict short-term taxi demand in multizone levels. Rago A. et al. [26] proposed an MTL model to evaluate the impacts of the observation windows of traffic profiles on classification accuracy prediction loss, complexity, and convergence. These MTL-based models have already shown that learning related tasks jointly results in a more accurate prediction than single task learning. However, the research into the application of MTL to multi-time-step prediction for all expressway segment types remains limited.

Confident forecasting is essential for the quantification of potential risks and uncertainties for delicate management [27], and existing research has explored the quantification of short-term prediction confidence. Lin et al. [28] provided a prediction interval with upper and lower bounds to quantify the uncertainty in short-term traffic volume prediction. Xing et al. [29] proposed a probabilistic forecasting model of traffic flow on the basis of a multikernel extreme learning machine and utilized a quantum behaved particle swarm optimization algorithm to optimize the output weights. Ghosh et al. [30] established Bayesian support vector regression to obtain error bars as the measurement of uncertainty, along with the predicted duration of incidents. According to existing research, an interval is normally given to represent confidence. Meanwhile, the confidence of a point traffic speed value in multi-time-step prediction has rarely been given attention.

Given the complex representations of optimized functions and the characteristics of multiple segment types, simultaneously predicting traffic speed and confidence for multiple time steps is a challenge. Dealing with this problem requires solutions to two issues. One is to make use of the correlation between the information at multiple terms, and the other is to fully extract the spatiotemporal features for different segment types. We observe that forecasting tasks can be decomposed into classification tasks, several groups of related regression tasks and confidence tasks. 1) Classification tasks. In this type of task, the category label is predicted for each segment type instance. We expect this subtask to be simpler than regression tasks on original data. It is also easily addressed by one convolution network. 2) Regression tasks. These subtasks learn from and improve one another by sharing informative representation. In one group of regression tasks, instances are predicted for corresponding segment types. 3) Confidence tasks. These subtasks estimate the confidence of speed prediction by calculating the probability of predicted speed belonging to each level of service.

To promote the training and testing of the proposed model, as well as its prediction evaluation, we utilize the loop detector data of the Shanghai urban expressway. The short-term prediction of speed for congested segments every 5–15 min is then realized. The experimental results show that the prediction results of the proposed DTNN model are significantly better than those of the other 10 methods compared herein. We also demonstrate the effectiveness of DTNN for prediction tasks. The contributions of this work are highlighted as follows:

1. A classification network can automatically recognize segment types, that is, normal, merging, diverging, weaving and ramp segments. This network provides the features of specific segments for speed prediction;
2. A confidence network is designed to simultaneously give the confidence value of the predicted speed. The result proves the credibility of the predicted speed;
3. MTL is adopted to train the DTNN for multistep speed prediction (i.e., next 5, 10 and 15 min). MTL improves the generalization of the model by sharing the feature expressions of multiple tasks;
4. The mean absolute percentage error (MAPE) loss function is used in training the proposed DTNN to improve model’s predictive accuracy in low-speed cases.

The paper is organized as follows. Section 2 presents the problem formulation for the multi-time-step speed prediction of expressway segments. Section 3 describes the structure of
2 PROBLEM FORMULATION

2.1 Speed dataset and feature preselection

The subject investigated in this study is the Shanghai urban expressway. The loop detector data from 1 March 2014 to 14 March 2014 are collected. A total of 3304 in-path loops are located on the expressway, and related data are aggregated every 5 min; the daily data volume can reach 1 million. Each piece of data contains a loop code, a recording time, section speed, section flow rate, section occupancy rate and other related information. The distribution of the loop detectors on Shanghai urban expressway is shown in Figure 1. The distance between adjacent detectors ranges from 150 to 500 m.

There are five segment types mentioned in this study, including normal segment, merging segment, diverging segment, weaving segment and ramp segment, denoted as S1–S5 in Figure 2.

Herein, a triangular fundamental diagram [31,32] is chosen to fit the relationship between speed and density. The fundamental diagram shows the relationship of the three traffic flow elements (i.e. flow, speed and density). The triangular fundamental diagram, which is well-known and often used in academia [33], represents the flow–density plane in a triangular shape. This diagram can describe the realistic situation in which drivers maintain the same speed with a range of low densities possibly because of the current speed limit. The equation is as follows:

\[
q = \begin{cases} 
  v_f \times k & \text{if } k < k_c \\
  q_c - \frac{k - k_c}{k_j - k_c} \times q_c & \text{if } k \geq k_c
\end{cases}
\]

\[
v = \frac{q}{k}.
\]

where \(q\) is the flow rate; \(k\) is the density; \(v\) is the speed; \(k_c\) is the critical density; \(v_f\) is the free flow speed and \(k_j\) is the jam density, which is always set to 200 pcu km\(^{-1}\) ln\(^{-1}\). On the basis of the loop detector data, the flow rate, speed and density in each time interval (i.e. 5 min in this study) can be obtained. Then, the parameters \(k_c\) and \(v_f\) of the triangular fundamental diagram can be calibrated by the least squares method.

Table 1 shows the calibration results of the traffic speed data derived from the five segment types.

![Figure 1](image1.png) The location of loop detectors on Shanghai urban expressway

![Figure 2](image2.png) Five categories of segment types in this study
As shown in Table 1, the traffic flow characteristic parameters, that is, critical density and free flow speed, show diversity among different segment types. For instance, diverge segment has both lowest critical density and free flow speed; normal segment has low critical density while the free flow speed is the highest; the traffic speeds of expressway are generally related to segment types and the traffic flow characteristics varies from different segment types [34,35]. The existing research and the above test indicate the training model should consider the impacts of different segment types.

To choose the appropriate spatiotemporal feature group, we compare the combinations of features with different numbers of adjacent segments and recent previous time intervals. Ten congested segments of the Shanghai expressway are randomly selected, and the 5 min speed predictions for the segments are obtained by training the BPNN for each feature group. BPNN is a simple neural network, which is easy to train. To ensure the fairness among different segments, we apply the same training number that can guarantee that the BPNN achieves convergence (i.e. 50 times in this research). Table 2 shows the average MAPE and root-mean-squared error (RMSE) values of the 10 congested segments for speed prediction.

As shown in Table 2, different numbers of adjacent upstream/downstream segments and previous time intervals are chosen as the combinations of features. The traffic condition of the target segment reflects the spatial dependency of the adjacent segments and the temporal dependency of the recent time intervals. However, the use of more features cannot significantly improve the predictive accuracy and may even bring noise and reduce training efficiency. The experiment indicates that the third combination of features performs best in terms of MAPE and RMSE. Therefore, the speed of the target segment, the five adjacent upstream and downstream segments, and the five previous intervals of each segment are selected. These 55 variables are grouped into a feature vector for training and testing in this study.

### 2.2 Problem formulation

Let \( t \) be the current time, and let \( t + \xi_l \) (\( i = 1, 2, \ldots, m \)) be the next time step to be predicted. In this study, the interval between time steps is 5 min. Let \( D \) be the dataset of traffic speed for the whole time period, including training and testing dataset. Given an input vector

\[
X = \{d_{t-1}, d_{t-2}, \ldots, d_{t-p}, r_{t-1}, r_{t-2}, \ldots, r_{t-j}\}. \quad (2)
\]

where \( t \) indicates the current time, \( d_{t}, \in D \) denotes the traffic speed of the target segment, and \( r_{t-j} \) indicates the speed of related segments at \( t-j \) respectively. The optimization problem for short-term speed prediction on time steps \((t+\xi_1, t+\xi_2, \ldots, t+\xi_m)\) is defined as follows:

\[
\min Z(W) = \{Z_1(W), Z_2(W), \ldots, Z_m(W)\}.
\]

s.t. \( W \in R^N \) \quad (3)

where

\[
W = [w_1, w_2, \ldots, w_N]^T.
\]

indicates the parameter vector of the speed predictive model \( g \) and \( N \) is the number of parameters in \( g \).

\[
Z_i(X) = \|g(X) + \xi - d_{t+\xi_i}\|^2.
\]

indicates the sub-objective function on the \( i \)-th time step and is calculated by the mean squared error (MSE) between \( g(X) + \xi_i \) and \( d_{t+\xi_i} \). \( d_{t+\xi_i} \) are the prediction speed of the model \( f \) and the observed speed of the target segment on the \( \xi \)-time step, respectively.

To solve Equation (2), the multi-objective optimization problem can be converted to a single objective optimization problem. This can be implemented by reducing the sum of \( Z_m \) over \( m \) during the training on a machine learning model. Here, \( Z_m \) is a sub loss function of the model.

### 2.3 Input data

As discussed in Section 2.1, a sample contains 55 variables, including the 5 latest time steps of the target segment and the first 5 adjacent upstream and downstream segments. The spatiotemporal relationship between the future and previous speeds is included in the sample. In this study, a sample \( X \) with 55 features is arranged as \([t, f] \) input matrix of the DTNN (i.e. \( t = 5 \) and \( f = 11 \)). Figure 2 shows the \([t, f] \) input data, in which column \( X_t \) indicates the speed of traffic node \( / \) and row \( X \), indicates the time step \( t \).

### 3 DECISION TREE NEURAL NETWORK

We propose the DTNN with MTL to predict speeds on multiple time steps. As shown Figure 4, the architecture of the DTNN contains three subnetworks, namely, classification, regression and confidence networks. The classification network aims to identify the segment type automatically. The regression network
predicts speed on time steps. The confidence network outputs the reliability of the predicted speed. These networks share the same feature extraction module.

The input to the DTNN is the \([t, I]\) matrix \(X\) shown in Figure 3. As observed in Figure 4, three sub-models share the same feature extraction module, which extracts spatiotemporal features from the input \(X, s^f_1, s^f_2\) are the outputs of the generalist and expert networks, respectively. \(s\) is the output of the classification network. \(b\) is the dot product of the expert and classification networks. \(\hat{s}\) indicates the convolution kernel. The activation function used is ReLU.

\[ y^f = \text{conv}(y^{f-1}, I), \]
\[ y_j^f = f(y_j^f). \]  
\[ (6) \]

where \(y_j^f\) indicates the sensitivity matrix at \(l\)-layer, \(y_j^{f-1}\) is the \(j\)-th feature map at \((l-1)\)-layer and \(y_j^f\) is the \(j\)-th output feature map at \(l\)-layer. \(I\) indicates the convolution kernel. The activation function \(f(\cdot)\) used is ReLU.

The FC layer combines feature maps to implement non-linear mapping. Let \(s = [s_1, s_2, \ldots, s_k]\) be the output vector of the classification network; it is the probability of samples belonging to the \(i\)-th category. The functions in the FC and output are in the following forms:

\[ u^f_i = \sum_j W^{f}_{ij}y^{f-1}_j + b^f_i, \]
\[ y_j^f = \text{dropout}[f(u_j^f), \rho]. \]
\[ \hat{s}_i = \rho \cdot \sum_j W^{f}_{ij}y_j^f. \]  
\[ (7) \]

Here, \(u^f_i\) is the sensitive factor, and \(y_j^f\) is the \(j\)-th hidden node at \(l\)-layer. \(W^{f}_{ij}\) denotes the weight of \(y_j^{f-1}\) connected to \(u^f_i\) and it indicates the importance of feature maps in generating different feature combinations. \(b^f_i\) denotes the bias to the node. \(\rho\) is the dropout probability.

\subsection{3.2 Regression network}

Regression networks contain expert and generalist networks, which predict speed on multiple time steps. Generalist network is trained without distinguishing segment types and it is good at extracting common features among segment types, such as the correlations with the traffic speed of previous time intervals. Whereas, each expert network is a specialist on one segment type and can extract typical features, such as the merge segment has more attention on the adjacent upstream segments, while the diverge segment has more attention on the adjacent downstream segments.

The input \(X\) to the regression network is the same as that to the classification network. In this study, the residual structure is adopted because it can be easily fit into a residual function, for example, \(\text{Res}(X) = f(X) - X\), relative to the underlying mapping \(f(X) = X\). The residual structure is as follows:

\[ y = f\{\text{conv}[f(\text{conv}(X, I^1)), I^2]\} + W^e_1X. \]  
\[ (8) \]

Here, \(I^1\) and \(I^2\) indicate the kernels of convolution layers, and \(W^e_1X\) denotes the shortcut across convolution layers in the DTNN.

The output \(\hat{s}_k\) of the generalist network is as follows:

\[ \hat{s}_k = \sum_{j=1}^{k} W_{jk}y_j. \]  
\[ (9) \]

Here, \(y_j\) is the \(j\)-th element at \(l\)-layer; \(W_{jk}\) is the weight of connecting the FC layer to the output and indicates the contribution of feature combinations to the output.

The expert network predicts the speed with the output vector of the classification network. \(b\) is the dot product of the output vectors of the expert and classification networks, as shown in Equation (7).

\[ \hat{b} = \hat{r} \cdot \hat{s}_k. \]  
\[ (10) \]

Here, \(\hat{r} = [\hat{r}_1, \hat{r}_2, \ldots, \hat{r}_k]\) is the output vector of the expert network, and \(\hat{s}_k\) indicates the \(k\)-th output; \(\hat{s} = [\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_k]\) denotes the output of the classification network. For instance, the classification network outputs \(\hat{s} = [0,0.7,0.3,0,0]\), and the
expert network outputs $\hat{y}_e = [19, 21, 15, 18, 22]$. Thus, the dot product is $\hat{b} = 19.2$.

The predicted speed $\hat{H}$ is the weighted sum of $\hat{y}_g$ and $\hat{b}$ and is written as

$$\hat{H} = w \times \hat{y}_g + (1 - w) \times \hat{b}. \quad (11)$$

Here, $w$ is the weight of the generalist and expert networks.

To observe the difference between expert and generalist networks, we use BPNN in training a “Generalist” model on mixed segment data and training “Expert-S1,” “Expert-S2,” “Expert-S3,” “Expert-S4” and “Expert-S5” models on the data of five segment types ($S1, S2, S3, S4, S5$, respectively). Figure 5 presents the MSEs of these models on the $S1, S2, S3, S4, S5$ and mixed segment data. The Expert-S1 model outperforms the other models on the $S1$ data, and the Generalist model provides more
FIGURE 6 Multitask learning of deep tree neural network

accurate prediction on mixed segment data than on S1 data. The experts and generalist networks have their own specific knowledge and improve each other.

3.3 Confidence network

Confidence is estimated by calculating the probability of the predicted speed belonging to each speed level of service. The five speed ranges for expressway is based on (Code for design of urban road engineering) (CJJ37-2012) [36]. In each level of service, the threshold speed is determined in consideration of both traffic efficiency and safety. The five levels are defined as: Level 1, forced flow, (0–40 km h$^{-1}$); Level 2, saturation flow, (40, 55 km h$^{-1}$); Level 3, steady flow, (55, 64 km h$^{-1}$); Level 4, steady flow-upper part, (64, 72 km h$^{-1}$); Level 5, free flow, (72, 80 km h$^{-1}$).

In the confidence network, two convolution layers, a max pooling layer, an FC layer with dropout and a logistic output layer are used to accomplish the probability estimation. Herein, max pooling reduces a feature map into half. The probabilities of the predicted speed belonging to different speed levels should be independent and non-competitive. Softmax usually gives conflicting probabilities of multiple outputs, thereby yielding an extremely large probability for a large input and a minimal probability for a small input. Hence, we employ logistics, and not softmax, to calculate the independent probabilities of multiple outputs.

The output vector of confidence network is \( \hat{c} \) = \( \{\hat{c}_1, \hat{c}_2, \ldots, \hat{c}_j, \ldots, \hat{c}_n \} \). If the predicted speed \( \hat{H} \) falls at speed level \( j \), the confidence is \( C = \hat{c}_j \), which is denoted as follow.

\[
\hat{c}_j = \frac{1}{1 + e^{-y_j}}. 
\]  

where \( \hat{c}_j \) is the \( j \)-th output of confidence vector \( \hat{c} \)-th service level), and \( y_j \) denotes the output of FC layer.

4 | TRAINING PROCESS

4.1 Multitask learning

Through the joint learning of multiple tasks, more informative representations can be achieved. Figure 6 shows the schematic of MTL in the DTNN. During MTL, subtasks belonging to one group are performed in parallel on multiple time steps. In each time step, three types of subtasks run in parallel, and they share feature expressions and promote one another. They are described as follows.

- The \( k+1 \) group of regression tasks are carried out to learn the data of \( k \) segment types and the mixed segment data. Each group of subtasks targets to predict speed on multiple segment data.
- The classification task classifies samples into \( k \) categories of segments;
- The confidence estimation task provides the confidence of the predicted speed and reduces the probability of inauthentic predicted speed.

For the classification and confidence estimation tasks, one hot code, representing \( k \) segment types and speed levels, is used as the label of samples. For regression tasks, the speed value \([v_5, v_{10}, v_{15}, \ldots]\) of the target segment on multiple steps \((t+5, t+10, t+15, \ldots)\).

The inductive bias, provided by different tasks, makes the model adaptive to different segment types and even unknown ones. For instance, the classification of segment types shares the segment features with the speed prediction task. These features probably improve the accuracy of speed prediction on a specific segment; the speed prediction task on a 5 min step provides short-term information for prediction on 10 and 15 min steps. The DTNN with MTL acquires knowledge of the segment type, segment speed, and the correlation between multiple time steps. Once the speed prediction on a new segment is required, the
DTNN with adequate knowledge is likely to discover the relationship between new segments and trained segments so as to improve speed prediction.

4.2 Loss function

For the classification task, cross entropy loss function is used. Multiple tasks are realized by multiple loss functions in the training process. The loss function $E_{\text{net}}$ of DTNN, which contains many sub loss terms, is as the form:

$$ E_{\text{net}} = [E_{\text{net}}^{i+5}, E_{\text{net}}^{i+10}, E_{\text{net}}^{i+15}], $$ (13)

where

$$ E_{\text{net}}^{i+\xi} = \left( L_{\text{softmax}} + L_{\text{logistic}} + \sum L_j + \lambda \| W \|^2 \right)_{+\xi}. $$ (14)

that is, $\xi = 5, 10, 15, \text{...}$.

Here, the first loss term

$$ L_{\text{softmax}} = -\sum_k (s_j \log \hat{y}_j). $$ (15)

indicates the cross-entropy loss of classification network (k segment types). $\hat{y}_j$ and $s_j$ are the output and label of the $j$-th segment class, respectively; the second loss term

$$ L_{\text{logistic}} = -\sum_j [c_j \log \hat{y}_j]. $$ (16)

indicates the cross-entropy loss of confidence network ($i$ speed levels), and $\hat{y}_j$ and $c_j$ are the output and probability of the $j$-th speed level, respectively. $W$ denotes the parameters in DTNN.

- **Square loss ($L_2$)**

  Square loss ($L_2$) is a conventional loss function, which is sensitive to outline data. $L_2$ loss function is used to train the regression networks.

  The third term

  $$ L_j = \| \hat{d} - d \|^2. $$ (17)

is a $L_2$ loss of the $j$-th regression network ($j = 1, 2, \ldots, k+1$), where $\hat{d}$ is the output of regression networks, and $d$ is an observed sample in $D$.

The fourth term, $\lambda \| W \|^2$, is the $L_2$ regulation on all the parameters. This aims to obtain a low complex network.

- **MAPE loss**

  $L_2$ loss can reduce the error with respect to most samples, but it cannot solve the problem of data imbalance. In short-term prediction, low-speed samples are more difficult to predict accurately than high- and middle-speed samples. In actual settings, the accurate prediction for low traffic speed is important because traffic management always pays close attention to the congestion state (i.e. low speed). The input traffic data usually contain a larger volume of high- and middle-speed samples than low-speed samples. These easy samples with high and middle speeds produce a large loss in total to make the learning process unaware of the valuable low-speed samples.

  To solve this problem, we employ MAPE loss. It penalizes the loss from samples of high speed while keeping the loss of low-speed samples unchanged. The third term can be changed into the following form:

$$ L = \mu \times \| 1 - \hat{y} / d \|. $$ (18)

Here, $\mu$ is set to 10 that enlarges the weight of the loss term.

MAPE loss only penalizes the importance of high-speed samples and increases the loss of low-speed samples when compared with $L_2$ loss. The focal loss ($L_3$) penalizes the easy and hard samples.

By combining square loss ($L_2$) and MAPE loss, we apply gradient descent to train the classification, regression and confidence networks.

4.3 Gradient descent in deep tree neural network

The parameters $W$ and $b$ are optimized by minimizing the loss function with iterative gradient descent:

$$ W_j = W_{j-1} - \eta \frac{\partial E_{\text{net}}}{\partial W_j}. $$
$$ b_j = b_{j-1} - \eta \frac{\partial E_{\text{net}}}{\partial b_j}. $$ (19)

$\eta$ is the learning rate that is set to 0.01.

We discuss the gradient descent on the parameters of the DTNN. Some parts are similar to the gradient descent of CNNs. Herein, we only present the distinguishing parts of the DTNN. The gradient on $W_s$ in the residual structure has the following form:

$$ \frac{\partial E_{\text{net}}}{\partial W_s} = \frac{\partial (f(\text{conv}(\text{conv}(\hat{x}, k^1), k^2)) + W(x))}{\partial W_s} = x. $$ (20)

The second convolution layer is shared by the regression, confidence and classification networks. Here, the feedback of loss terms on the kernel parameters is in the following form:

$$ \frac{\partial E_{\text{net}}}{\partial u_j} = \frac{\partial C_{\text{logistic}}}{\partial u_j} + \frac{\partial C_{\text{softmax}}}{\partial u_j} + \sum \frac{\partial L}{\partial u_j}. $$ (21)

The FC layers of the different components in the DTNN are independent on tasks. For the generalist network, the gradients
on the weights of the FC layers are in the following form:

$$\frac{\partial E_{\text{net}}}{\partial W_{ij}^{\text{fc}}} = \frac{\partial E_{\text{net}}}{\partial h_{ij}} \frac{\partial h_{ij}}{\partial W_{ij}^{\text{fc}}} + 2\lambda W_{ij}^{\text{fc}},$$

$$\frac{\partial E_{\text{net}}}{\partial b_{ij}} = \frac{\partial E_{\text{net}}}{\partial h_{ij}} \frac{\partial h_{ij}}{\partial b_{ij}} = \frac{\partial E_{\text{net}}}{\partial h_{ij}}. \quad (22)$$

For the expert network, the gradients on the FC layers are as follows:

$$\frac{\partial E_{\text{net}}}{\partial W_{ij}^{\text{exp}}} = \frac{\partial E_{\text{net}}}{\partial h_{ij}} \frac{\partial h_{ij}}{\partial W_{ij}^{\text{exp}}} + 2\lambda W_{ij}^{\text{exp}}. \quad (23)$$

In the regression network, the expert network has $y = s_n$, and the generalist network has $y = h_n$. In the classification network, the gradient on the FC layer is as follows:

$$\frac{\partial E_{\text{net}}}{\partial W_{ij}^{\text{cls}}} = \frac{\partial C_{\text{softmax}}}{\partial h_{ij}} \frac{\partial h_{ij}}{\partial W_{ij}^{\text{cls}}} = \frac{\partial C_{\text{softmax}}}{\partial h_{ij}} \sum_j y_j^{-1}. \quad (24)$$

For the confidence network, the gradient is as follows:

$$\frac{\partial E_{\text{net}}}{\partial W_{ij}^{\text{cfd}}} = \frac{\partial C_{\text{logistic}}}{\partial h_{ij}} \frac{\partial h_{ij}}{\partial W_{ij}^{\text{cfd}}} = \frac{\partial C_{\text{logistic}}}{\partial h_{ij}} \sum_j y_j^{-1}. \quad (25)$$

## 5 | EXPERIMENT SETTING

### 5.1 Data description

Loop detector data from Shanghai urban expressway during 1 March–4 April 2014 are utilized in this study. There are 3304 in-path loops located on the expressway. All the locations of the loop detectors are known and the space between adjacent detectors varies from 150 to 500 m. The data collection is aggregated in every 5 min. Each data includes a loop code, recording time, speed, flow rate and occupancy rate. A total of 406 congested segments are collected, containing 171 segments of S1, 22 segments of S2, 21 segments of S3, 61 segments of S4 and 131 segments of S5. A target segment is randomly selected, and the other 12 days as the training set and data for 3 days as test set. This operation is conducted five times for five-cross validation. The data amount for training is around 0.468 million records, and the data for testing is about 0.117 million records.

### 5.2 Running algorithm

In this study, we perform three short-term predictions on traffic speeds (i.e. the current time is 8:00 AM, and the predictions are 8:05 AM, 8:10 AM and 8:15 AM). We implement the DTNN in TensorFlow. The parameters of the DTNN are given in Table 3. All experiments are performed on a PC with an Intel i5 4.0 GHz CPU and an NVIDIA 1060i GPU.

### 5.3 Evaluation index

MSE, RMSE, and mean absolute error (MAE) are three typical error measures. MAPE puts a penalty on errors with respect
to low speed. Therefore, a small MAPE values equates to an accurate prediction of the congestion state. This measure is a commonly used evaluation tool in short-term traffic speed prediction.

R-squared (i.e. $R^2$) evaluates the goodness of fit, which represents the fitness of a model with respect to observed samples. The larger $R^2$ is, the better the fitness of the model is. All methods use the same number of variables to build predictive models. Hence, one can reasonably compare different predictive models in terms of $R^2$. The decline of $R^2$ from a training set to a test set reflects the generalization of the built model. A small decline of $R^2$ indicates good model generalization.

6 Model result validation and discussion

6.1 Predicted speed and confidence

The predicted speed curve, along with its confidence and absolute error (AE), is shown in Figure 8. The confidence network...
achieves 85% accuracy on the training set and 82% accuracy on the test set. For most of the samples, the confidence of the predicted speed of the DTNN ranges from 0.9 to 1, and the AEs are small. We select five samples (lines 1, 2, 4, 5 and 7) of large AEs as instances. Confidence less than 0.8 is given on three samples denoted by lines 2, 5 and 7. The result illustrates that the predicted speeds for these samples are questionable. Through the confidence calculation, false speed prediction can be raised.

A large AE will not generate a low confidence. When observed and predicted speeds belong to level 1 or 5, the confidence may be high despite the large AE. Specifically, the speed interval of levels 1 and 5 is longer than that of the other levels. When the observed speed belongs to level 2, 3 or 4, even a small AE of the predicted speed could lead to a low confidence. For example, the DTNN yields low confidence for a sample (line 5) but a small AE. In a few cases, confidence misleads the speed prediction. For a given sample (line 6), the DTNN outputs a low confidence but an accurate prediction.

Residuals, that is, the difference between observed data and predicted values, can assess the accuracy of predictive models. The best residuals reveal whether a model has made use of available information in the data despite it not being the most accurate model in the prediction. Figure 9 shows three models in a 5 min step accounting for available information. Autocorrelations exist in the residuals. As observed from the autocorrelation plots, the residuals of the DTNN are not highly correlated. The time plots show that the DTNN residuals have fewer patterns and more uniform and smaller variances than the LSTM and ARIMA models. The histograms demonstrate that the DTNN's residuals follow a normal distribution with small variance. Three models are biased; the mean residual of the DTNN is −5.5 while those for the LSTM and ARIMA models are −5 and −7, respectively.

6.2 Comparison between predictive models

Figure 10 shows the predicted curves of the models for the types of ramp, diverging, and merging. DTNN-S builds three individual models in three steps, and DTNN-M trains only one model in three steps. As shown in Figure 10, at the 5 min step, DTNN-S yields a correct predictive trend under high speeds despite its low accuracy for low speeds. DTNN-M yields an accurate speed prediction between 9:30 AM and 10:00 AM and a correct trend at 8:30 AM (e.g., the congestion state). This result is denoted as a circle on the graph. At the 10 min step, DTNN-M yields a similar predicted curve to those of the other models. At the 15 min step, the curve of DTNN-M fits the observed speed accurately at 12:00 AM. DTNN-S only concentrates on speed prediction in one step without considering the other steps. DTNN-M can improve the accuracy by sharing information between multiple steps. Hence, DTNN-M performs the best at the 15 min step but performs relatively poorly at the 5 min step.
FIGURE 9  Residual (a) time plot, (b) histogram and (c) autocorrelation

FIGURE 10  Predicted curve on ramp in 5 min step, diverging in 10 min step and merging in 15 min step (working hours)
Tables S1, S2 and S3 present the MAPE and MAE values of the 15 models in realizing speed predictions at the 5, 10 and 15 min steps for the normal (S1), merging (S2), diverging (S3), weaving area (S4) and ramp segments (S5). In Tables S1, S2 and S3, DTNN-S obtains the smallest MAPE for at least two segment types at the 5 and 15 min steps. DTNN-M obtains an incremental improvement in MAE from 5 to 15 min steps. DTNN-S yields a smaller MAPE than most of the other models, but its MAPE is slightly larger than those of LSTM, RNN and GRU.

LSTM, RNN and GRU perform well in most cases. The LSTM curve is the most accurate for the 10 and 15 min steps, and RNN achieves an accurate prediction for the 5 min step. At the 5 min step, three networks correctly predict the congestion state during rush hour. At the 15 min step, three networks generate the wrong prediction at 12:00 AM. They gradually yield a small increase in MAE from the 5 min step to the 15 min step. Their small MAPEs at the three steps suggest the high accuracy of low-speed prediction. In the graphs, DCNN, a six-layer network, yields a much lower speed at 8:45 AM than the observed speed. In terms of MAPE and MAE, the six-layer DCNN is better than the conventional models but is poorer than DTNN, LSTM, RNN and GRU.

BBRT performs well at the 10 and 15 min steps but performs poorly at the 5 min step. RF, CART, SVR, BPNN and RBF-NN do not perform well at these three steps. BPNN yields a correct predictive trend for the traffic congestion state, but its MAEs and MAPEs for the five segment types are large. RBF-NN obtains a perfect fitting curve on the training samples but performs poorly in the test set. It incorrectly predicts the speed at the 5 min step during rush hour (2:00 PM). Three ARIMA models with parameters \((p, d, q) = (1, 1, 0), (2, 1, 2)\) and \((2, 2, 2)\) are tested. They provide well-fitting predicted curves at the 5 min step and poor-fitting ones at the 10 and 15 min steps. Relative to DTNN, the ARIMA models cannot effectively deal with traffic emergencies. Regardless of segment type, the poor prediction of the ARIMA models can also be observed from the large MAPEs and MAEs in some cases.

### 6.3 Improvement in low-speed prediction

The accurate prediction of the traffic congestion state is more important than that of the smooth traffic state. Such accurate prediction can greatly benefit traffic management. Herein, we observe five loss functions, namely, \(L_1\) loss, \(L_2\) loss, \(L_3\) loss, MAPE loss and shrink loss, in low-speed prediction.

Observed from Table 4, compared to the other loss functions, MAPE loss yielded only 0.2% reduction on MAPE on the samples of low and high speed. In some cases, MAPE loss has not much advantage in prediction on high speed samples. Owing to the sensitivity to low speed (i.e. speed \(< 40 \text{ km h}^{-1}\)), MAPE loss function can accurately predict on low speed samples. It obtained at least 12.36% smaller MAPE than that of the other loss functions except \(L_1\) loss. The small increase of MAE is in exchange for the decrease of MAPE. On S2 segment, \(L_1\) loss achieved a smallest MAPE on low speed samples. Meanwhile, MAPE loss function yielded only 2.97% increase of MAE on different segment types.

Figure 11 shows the scatter plots of DTNN-M using the \(L_2\) loss and MAPE loss relative to LSTM, RNN, and GRU at the 5 and 15 min steps. The isoline represents the predicted speed equal to the observed speed. The low speed samples are shown by circles in the pictures. The predicted points of DTNN-M using the MAPE loss at low speeds are nearer to the isoline than those using the \(L_2\) loss. Relative to LSTM, RNN and GRU, DTNN-M with MAPE loss generally predicts similar speeds at the 5 min step; at the 15 min step, DTNN-M with MAPE loss

### Table 4 MAPE Obtained using different losses

|         | S1 | S2 | S3 | S4 | S5 | S1 | S2 | S3 | S4 | S5 | S1 | S2 | S3 | S4 | S5 |
|---------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| MAPE-   |    |    |    |    |    | MAPE- |    |    |    |    | MAPE- |    |    |    |    |
| loss    |    |    |    |    |    | loss    |    |    |    |    | loss    |    |    |    |    |
| MAPE    | 6.3 | 11.22 | 11.71 | 10.25 | 15.62 | 11.87 | 13.62 | 13.55 | 12.72 | 18.00 | 13.85 | 15.84 | 15.24 | 14.51 | 19.56 |
| \(L_1\) | 11.88 | 14.52 | 14.62 | 14.38 | 21.03 | 14.13 | 16.30 | 16.87 | 16.36 | 23.45 | 15.99 | 17.98 | 19.04 | 18.38 | 25.12 |
| \(L_2\) | 11.21 | 11.44 | 12.39 | 11.36 | 16.92 | 13.13 | 13.97 | 14.06 | 13.00 | 19.04 | 14.99 | 16.24 | 15.72 | 14.39 | 21.32 |
| \(L_3\) | 8.84 | 11.41 | 11.65 | 10.39 | 15.45 | 11.90 | 13.69 | 15.03 | 12.77 | 17.64 | 13.81 | 15.84 | 18.08 | 14.26 | 19.72 |
| Shrink | 12.11 | 13.24 | 13.66 | 12.41 | 17.00 | 13.95 | 14.52 | 13.70 | 13.26 | 21.29 | 15.25 | 16.03 | 15.88 | 15.26 | 22.67 |
| MAPE    | 9.54 | 32.95 | 6.83 | 20.36 | 30.75 | 12.57 | 40.67 | 11.59 | 30.74 | 39.42 | 18.62 | 48.69 | 14.88 | 38.27 | 48.25 |
| \(L_1\) | 25.05 | 49.52 | 41.26 | 48.31 | 82.92 | 31.82 | 55.24 | 50.45 | 55.06 | 92.65 | 37.75 | 60.76 | 62.09 | 61.67 | 102.4 |
| \(L_2\) | 20.12 | 35.87 | 12.23 | 39.91 | 54.67 | 19.07 | 44.21 | 15.94 | 45.65 | 67.55 | 23.93 | 51.88 | 22.22 | 51.91 | 79.30 |
| \(L_3\) | 14.23 | 30.71 | 25.99 | 26.96 | 48.34 | 18.56 | 38.19 | 26.47 | 38.64 | 60.11 | 24.32 | 45.21 | 30.97 | 43.67 | 69.93 |
| Shrink | 28.21 | 46.23 | 36.33 | 49.55 | 61.75 | 32.11 | 52.49 | 45.65 | 54.96 | 72.02 | 39.30 | 58.22 | 60.32 | 62.77 | 83.39 |
is superior to LSTM, RNN and GRU in terms of the approximation with the isoline.

6.4 Generalization of predictive models

We compare the generalization of two DTNNs (using $L_2$ loss) with that of the other 10 models. Five datasets comprising samples derived from different segment types are generated for testing the models, denoted as data1, 2, 3, 4 and 5. Five measures, namely, MAPE, MSE, MAE, $R^2$, and the decline of $R^2$ from training to testing, are used to evaluate the models. $R^2$ is used to evaluate model fitting quality. Machine learning models usually achieve high quality in fitting training sets, but trained models did not fit equally well in new datasets. The $R^2$ value from the training set to a new dataset is bound to decrease. A small decline in $R^2$ from the training set to a new dataset suggests that the model built on the training set can effectively adapt to the new dataset. Hence, this model can be considered to have high generalization.

Figure 12 shows the evaluation bars of the models at three-time steps. All the results show the mean values of five runs on the datasets. As shown in the figure, two DTNNs, BBRT, RNN, LSTM and GRU obtain small MAPEs and large $R^2$ values a three-time steps. The large $R^2$ and the small decline of $R^2$ illustrated the satisfied generalization of DTNN. Among the other models, BBRT, DCNN, RNN, LSTM and GRU show a small decline in $R^2$ for different datasets. This result highlights their good generalization. BPNN and RF obtain small MSEs and MAEs and a high $R^2$, but the decline of $R^2$ is unstable. The generalization of these models is uncertain. By contrast, CART, SVR and RBF-NN obtain large MSEs and MAEs and small $R^2$. Their generalization is certainly poor.

7 DISCUSSION

Training set size. In theory, training set size is larger, the higher accuracy of the predictive models. But this is not absolute. Actually, highly diverse data is more valuable for training a generalized model. If the diversity is not enough and the similarity of samples is too high, no matter how much training set cannot generate a good model.

In this study, we used 12 days data to train the DTNN and the predictive accuracy in terms of MAPE and MAE was satisfied. This illustrated that 12 days data was enough to train a DTNN.
TABLE 5  MAPE and MAE of different training set size on 5 min and 15 min steps ($L_2$ loss)

| Training size | MAPE (5 min) S1 | MAPE (5 min) S5 | MAE (5 min) S1 | MAE (5 min) S5 |
|---------------|----------------|----------------|----------------|----------------|
| 7 days        | 10.74          | 29.11          | 5.02           | 12.58          |
| 8 days        | 9.96           | 28.81          | 4.66           | 12.27          |
| 9 days        | 10.92          | 28.64          | 5.05           | 10.76          |
| 10 days       | 10.49          | 24.69          | 4.40           | 10.40          |
| 11 days       | 10.71          | 19.90          | 4.41           | 9.05           |
| 12 days       | 11.21          | 16.92          | 4.42           | 8.45           |

| Training size | MAPE (15 min) S1 | MAPE (15 min) S5 | MAE (15 min) S1 | MAE (15 min) S5 |
|---------------|----------------|----------------|----------------|----------------|
| 7 days        | 14.66          | 31.82          | 7.98           | 13.2           |
| 8 days        | 14.20          | 31.65          | 7.65           | 13.07          |
| 9 days        | 15.19          | 32.05          | 7.09           | 11.64          |
| 10 days       | 14.56          | 29.57          | 6.79           | 11.55          |
| 11 days       | 14.74          | 25.22          | 6.01           | 10.25          |
| 12 days       | 14.99          | 21.32          | 5.68           | 10.01          |

It is noted that there are total 15 days data. So, the proportion of 4:1, 12 days for training and 3 days for test, is reasonable. To understand the requirement of training set size in training a DTNN, we use 7, 8, 9, 10 and 11 days data for training, respectively. Through the tests on 3 days data, the results in Table 5 show that 7, 8, 9, 10 and 11 days lead to an obvious accuracy decrease of DTNN on S5 segment, but they obtained a little or no accuracy decrease on S1 segment. This illustrated that it is still necessary to use more data for training DTNN in some cases.

**Knowledge acquisition.** DTNNs can build a highly generalized model and reduce the structural risk to overcome the overfitting. The abundant knowledge acquisition of DTNN is attributed to the MTL. The recognition of segment types provides the knowledge of segment speed to regression networks. In DTNN, the joint of expert networks with the segment knowledge can perform accurately on the segment types, which have been trained by DTNN. For the new or unknown segment types, the generalist network trained on mixed segment types can always predict a speed close to observed value.

Tree-based ensemble regression models could perform well in some cases. This is because they have the diversity of sub models and generate different feature combinations in the models. RNN, LSTM and GRU are the models, which will deal with the sequential data. They have the potential in predicting short-term speed if integrated with spatial features networks. However, these comparison models only can acquire limited relationship information from the input data because of the single learning task. This is why they cannot outperform DTNN in most cases.

**Stable prediction.** It must be noted that there is no model can perform perfectly or outperform the other models in every case. As the Chinese proverb “specialist only master his own field,” DTNN with different loss functions can play a role in different fields.

- DTNN with MAPE loss perform well on low speed prediction. This can be observed from Table 4.
- $L_2$ loss cannot well deal with traffic emergencies because of its sensitivity to outliers. It reduced the error with respect to outliers by sacrificing the predictive accuracy on the other samples. Usually, on expressway, low speed samples are minority and high and middle speed samples are majority. So, $L_2$ loss can still achieve error reduction on the whole dataset.

**FIGURE 12**  Predictive accuracy of 12 models on traffic speed in expressway

- **Knowledge acquisition.** DTNNs can build a highly generalized model and reduce the structural risk to overcome the overfitting.
- **Stable prediction.** It must be noted that there is no model can perform perfectly or outperform the other models in every case. As the Chinese proverb “specialist only master his own field,” DTNN with different loss functions can play a role in different fields.
It can be observed from Table S1, S2, S3 that DTNN cannot give stable accurate prediction on different segments. This is because the actual traffic speed of different segments are various. Some segments are dominated by low speed state, while some ones keep middle and high speed state for a long time. DTNN with $L_2$ loss can high predicted accuracy if the segments are middle and high speed state, while DTNN with MAPE loss can achieve high predicted accuracy on low speed segments.

**External factors and improvement.** Many studies have indicated that weather and weekdays/weekends are important factors that influence the traffic prediction accuracy. The weather factors include the weather status, temperature and wind. These four factors can be added to the input as new features if DTNN is applied for prediction on more time steps. Meanwhile, the building works would also have impact on the state of segments.

In addition to the external factors, the missing and incomplete data is also an important factor affecting the accuracy. Currently, ground loop detectors sometimes fail to detect the traffic flow. So, the new detector equipment can be employed to help detect the traffic flow more accurately. With the accurate dataset, the trained model is bound to be more accurate in short-term speed prediction.

8 | CONCLUSION

To solve multiple short-term (e.g. 5, 10 and 15 min) predictions of traffic speeds for different segment types, we propose DTNN with MTL. As a result of the different features of segment types and the strong correlation between traffic speeds at continuous time periods, we assign the DTNN with a classification task for distinguishing the segment types of samples and multiple groups of regression tasks for simultaneously fitting samples of continuous times on each segment type. These classification and regression tasks can improve and learn from one another.

In the DTNN, the classification network for the classification task shares informative representation with the generalist and expert networks, which carry out groups of regression tasks. In further improving the forecast of traffic congestion state, we propose MAPE loss to make MTL biased to low speeds. The traffic speed dataset of the Shanghai Expressway is used to test our method and 10 other methods. The results show that DTNN with $L_2$ loss obtains the smallest MAE in most cases and that the DTNN with MAPE loss can efficiently improve the predictive accuracy for low-speed samples. In terms of $R^2$, the DTNN also achieves the largest value in most cases. In addition, the smallest reduction of $R^2$ from training to testing illustrates the best generalization of the DTNN model.

We identify three future research directions. The first is to analyze the sensitivity to different prediction time intervals (5 min in this research) to evaluate the time dimension generalization of the proposed model. The second is to apply the classification and regression processes to urban road segments. The final prospect is to further improve the accuracy and confidence, especially in low-speed predictions.

**ACKNOWLEDGEMENTS**

The National Natural Science Foundation of China under Grant No. 52002262 and the National Key Research and Development Program of China under Grant No. 2018YFB1600500 supported this work.

**CONFLICT OF INTEREST**

None.

**ORCID**

Xiang Wang https://orcid.org/0000-0001-6113-0923

**REFERENCES**

1. Yao, R., Zhang, W., Zhang, L.: Hybrid methods for short-term traffic flow prediction based on ARIMA-GARCH model and wavelet neural network. J. Transp. Eng., Part A: Systems 146(8), 04020086 (2020)
2. Chan, K.Y., Dillon, T.S., Singh, J., et al.: Traffic flow forecasting neural networks based on exponential smoothing method. 2011 6th IEEE Conference on Industrial Electronics and Applications. IEEE, Beijing, pp. 376–381 (2011)
3. Guo, J., Williams, B.M.: Real-time short-term traffic speed level forecasting and uncertainty quantification using layered Kalman filters. Transp. Res. Record 2175(1), 28–37 (2010)
4. Zhu, Z., et al.: Short-term traffic flow prediction with linear conditional Gaussian Bayesian network. J. Adv. Transp. 36(6), 1111–1123 (2016)
5. Djennouri, Y., et al.: Adapted k-nearest neighbors for detecting anomalies on spatio–temporal traffic flow. IEEE Access 7, 10015–10027 (2019)
6. Kuang, L., et al.: Predicting duration of traffic accidents based on cost-sensitive Bayesian network and weighted K-nearest neighbor. Journal of Intelligent Transportation Systems 23(2), 161–174 (2019)
7. Castro-Neto, M., et al.: Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions. Expert Syst. Appl. 36(3), 6164–6173 (2009)
8. Cheng, J., Li, G., Chen, X.: Research on travel time prediction model of freeway based on gradient boosting decision tree. IEEE Access 7, 7466–7480 (2018)
9. Wu, J., Xu, H.: Annual average daily traffic prediction model for minor roads at intersections. J. Transp. Eng., Part A: Systems 145(10), 04019041 (2019)
10. Bayati, A., Asghari, V., Nguyen, K., et al.: Gaussian process regression based on spatio–temporal traffic flow. IEEE Global Communications Conference (GLOBECOM). IEEE, Washington D.C., pp. 1–7 (2016)
11. Chen, W., et al.: A novel fuzzy deep-learning approach to traffic flow prediction with uncertain spatial–temporal data features. Future Gener. Comput. Syst. 89, 78–88 (2018)
12. Dai, G., Ma, C., Xu, X.: Short-term traffic flow prediction method for urban road sections based on spatio–time analysis and GRU. IEEE Access 7, 143025–143035 (2019)
13. Wang, J., Chen, R., He, Z.: Traffic speed prediction for urban transportation network: A path based deep learning approach. Transp. Res. Part C: Emerging Technologies 100, 372–385 (2019)
14. Gu, Y., et al.: An improved Bayesian combination model for short-term traffic prediction with deep learning. IEEE Trans. Intell. Transp. Syst. 21(3), 1332–1342 (2019)
15. Zhao, Z., et al.: LSTM network: A deep learning approach for short-term traffic forecast. IET Intel. Transport Syst. 11(2), 68–75 (2017)
16. Yu, H., Wu, Z., Wang, S., et al.: Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks. Sensors 17(5), 1501 (2017)
17. Guo, S., et al.: Attenion based spatial-temporal graph convolutional networks for traffic flow forecasting. Proceedings of the AAAI Conference on Artif. Intell. 33, 922–929 (2019)
18. Fukuda, S., et al.: Short-term prediction of traffic flow under incident conditions using graph convolutional recurrent neural network and traffic simulation. IET Intel. Transport Syst. 14(8), 936–946 (2020)
19. Chen, X.M., Zhang, S., Li, L.: Multi-model ensemble for short-term traffic flow prediction under normal and abnormal conditions. IET Intel. Transport Syst. 13(2), 260–268 (2018)
20. Ou, J., et al.: Short-term traffic flow forecasting for urban roads using data-driven feature selection strategy and Bias-corrected random forests. Transp. Res. Record 2645(1), 157–167 (2017)
21. Zhang, K., Wu, L., Zhu, Z., et al. A multitask learning model for traffic flow and speed forecasting. IEEE Access 8, 80707–80715 (2020)
22. Gong, T., et al.: A comparison of loss weighting strategies for multi task learning in deep neural networks. IEEE Access 7, 141627–141632 (2019)
23. Jia, H., et al. ADST: Forecasting metro flow using attention-based deep spatial-temporal networks with multi-task learning. Sensors 20(16), 4574 (2020).
24. Zhang, K., Liu, Z., Zheng, L.: Short-term prediction of passenger demand in multi-zone level: Temporal convolutional neural network with multi-task learning. IEEE Transactions on Intelligent Transportation Systems 21(4), 1480–1490 (2019)
25. Kuang, L., et al.: Predicting taxi demand based on 3D convolutional neural network and multi-task learning. Remote Sensing 11(11), 1265 (2019)
26. Luo, H., Cai, J., Zhang, K., et al.: A multi-task deep learning model for short-term taxi demand forecasting considering spatiotemporal dependencies. Journal of Traffic and Transp. Eng. (English Edition) 7(2), 1–12 (2020)
27. Rago, A., et al.: Multi-task learning at the mobile edge: An effective way to combine traffic classification and prediction. IEEE Trans. Veh. Technol. 69, 10362–10374 (2020)
28. Voronov, S., Frisk, E., Krysander, M.: Data-driven battery lifetime prediction and confidence estimation for heavy-duty trucks. IEEE Trans. Reliab. 67(2), 623–639 (2018)
29. Liu, L., et al.: Quantifying uncertainty in short-term traffic prediction and its application to optimal staffing plan development. Transp. Res. Part C: Emerging Technologies 92, 323–348 (2018)
30. Xing, Y., Ban, X., Guo, C., et al.: Probabilistic forecasting of traffic flow using kernel based extreme learning maching and quantum-behaved particle swarm optimization. 2016 4th International Conference on Cloud Computing and Intelligence Systems (CCIS). IEEE, Beijing, pp. 205–209 (2016)
31. Ghosh, B., Asif, M.T., Dauwels, J.: Bayesian prediction of the duration of non-recurring road incidents. 2016 IEEE Region 10 Conference (TENCON). IEEE, Singapore, pp. 87–90 (2016)
32. Daganzo, C.F.: A behavioral theory of multi-lane traffic flow. Part I: Long homogeneous freeway sections. Transp. Res. Part B: Methodological 36(2), 131–158 (2002)
33. Daganzo, C.F., Geroliminis, N.: An analytical approximation for the macroscopic fundamental diagram of urban traffic. Transp. Res. Part B: Methodological 42(9), 771–781 (2008)
34. Qu, X., Zhang, J., Wang, S.: On the stochastic fundamental diagram for freeway traffic: Model development, analytical properties, validation, and extensive applications. Transp. Res. Part B: Methodological 104(10), 256–271 (2017)
35. Gao, S., et al.: Understanding urban traffic-flow characteristics: A rethinking of betweenness centrality. Environment and Planning B: Planning and Design 40(1), 135–153 (2013)
36. Zhao, N., et al.: Analysis of traffic flow characteristics on ring road expressways in Beijing: Using floating car data and remote traffic microwave sensor data. Transp. Res. Record 2124(1), 178–185 (2009)
37. He, K., Zhu, Z.F., Code for Design of Urban Road Engineering. China Architecture & Building Press, Beijing, China (2012)

SUPPORTING INFORMATION
Additional supporting information may be found online in the Supporting Information section at the end of the article.

How to cite this article: Tao Y, Wang X, Zheng J, E W, Zhao P, Meng S. Deep tree neural network for multiple-time-step prediction of short-term speed and confidence estimation. IET Intell Transp Syst. 2021;15:446–462. https://doi.org/10.1049/itr2.12037