Research Article

Stacking Ensemble Learning Process to Predict Rural Road Traffic Flow

Arash Rasaizadi and Seyedehsan Seyedabrishami

Department of Civil and Environmental Engineering, Tarbiat Modares University, Tehran, Iran

Correspondence should be addressed to Seyedehsan Seyedabrishami; seyedabrishami@modares.ac.ir

Received 23 January 2022; Revised 24 April 2022; Accepted 12 May 2022; Published 1 June 2022

1. Introduction

Intelligent transportation systems are one of the leading efficient tools for transportation network traffic management. The result of using these systems is achieving or maintaining the balance between transportation supply and demand with low cost [1]. Intelligent transportation systems include various subsystems which one of the most important of them is the advanced traveler information system. By this system, available information about the transportation network is given to travelers to plan their travels with more awareness. This information can be informed for the current state of the network, but its effectiveness becomes more if it is predicted and informed for the future of the transportation network [2]. In such circumstances, the traveler is more prepared to choose the appropriate route and departure time and even to choose to have a trip or cancel it. Generally, traffic parameters such as traffic volume [3], average speed [4], and travel time [5] are predicted and informed by intelligent systems. As the time horizon of these predictions is limited to the near future compared to the time horizon of classical 4-step transportation planning prediction, they are short-term predictions.

Prediction of traffic parameters is made by analyzing the past observations and discovering effective features on the variation of traffic parameters. For this purpose, the use of time-series models as a tool based on statistics and probability has more antiquity in previous studies. In time-series models, each traffic parameter’s variation is a function of that parameter’s previously observed values, independent effective features, and random term. For example, Kumar and Vanajakshi [6] have predicted the traffic flow using the seasonal autoregressive integrated moving average (SARIMA). Results show that the model is more accurate than the historical average models. In Yan et al.’s study [7], autoregressive integrated moving average (ARIMA) has been used to predict subway passengers’ flow. Time-series models are only capable of considering linear relationships between independent and dependent variables. On the other hand, by increasing the number of observations and features, traffic...
data are converted to big data. These models are not compatible with big data characteristics, including volume, velocity, and variety [8].

Another approach to predict traffic parameters is the machine learning (ML) approach. ML models are compatible with big data characteristics and can depict linear and nonlinear relationships. Lack of interpretability and disability in discovering causal relationships are the main weaknesses of ML models, and time-series models are superior to ML models in this regard [9]. ML models are diverse and artificial neural network (ANN) [10], support vector machine [11], and decision tree [12] are some of the widely used ML models. To predict the traffic flow, Ma et al. [13] use ANN optimized by genetic algorithms and exponential smoothing. The results show that the optimization of the artificial neural network improves prediction accuracy. Simple ANN considers consequential observations independently. To capture the relationship between successive observations, Lu et al. [14] have used a recurrent neural network (RNN) model. The RNN model emphasizes the importance of the time-series nature of data by forming neural network blocks at different time intervals. Each block’s input is the output of another block related to past times and predictive features. Also, long short-term memory (LSTM) model is another type of RNN model that considers the dependency of observations for both short-term (near past observations) and long-term (far past observations) pasts. This algorithm is used in Farahani et al. [15] and Chen [16] studies. Wang et al. [17] focus their research on ANN models’ weakness in interpreting results. After training a deep neural network model for traffic flow prediction, the proposed model is interpreted in two different ways: first, justifying the number of layers and nodes; second, explaining the causality between historical data and future state of traffic.

The ML models used to predict traffic variables are not limited to the neural network-based modes and traffic flow prediction problem. As an example of other ML models and other traffic parameters, Xu et al. [18] predict the nominal traffic state by using the Kalman filter, Zheng et al. [19] predict traffic speed by K-nearest neighbours (KNNs), Liu et al. [20] predict traffic congestion by random forest (RF), and Yang et al. [21] predict travel time by Markov chain method.

Variety of short-term prediction methods, and on the other hand, lack of a technique that has the highest accuracy for all situations has led researchers to the use of ensemble learning process. In this process, the base models’ output is used and to provide one unique final prediction. In general, the ensemble learning process is divided into three categories: bagging, boosting, and stacking. In the bagging process, the base models are trained with the same training dataset, and by averaging or voting, the final prediction is determined. In the boosting process, the base models are trained sequentially to improve the old model’s prediction accuracy in the current model. In the stacking process, predictions of base models are introduced as inputs of a supermodel that can be an ML model, the supermodel’s output is the final prediction [22]. By using bagging ensemble modeling, Moretti et al. [23] combine predictions of statistical and neural network models to predict traffic flow. Yenru and Haghani [24] use a gradient boosting regression tree model to predict travel time. Ma et al. [25] use a contextual convolutional recurrent neural network to recognize inter- and intra-day traffic patterns. Lin et al. [26] propose a stacking ensemble learning process to predict public bicycle traffic flow. In all of these three studies, using ensemble learning modeling leads to more accuracy of predictions than base models.

In this study, hourly traffic flow is predicted using three ML base methods, including KNN, RF, and RNN. Outputs of these models are given to XGboost as a stacking supermodel and bagged averaging to predict the final output in the ensemble learning process. The predictive models are divided into two categories: short-term and mid-term. In the short-term models, in addition to the external predictive features including cyclical temporal features, holidays, and weather conditions, the observed traffic flow in the previous 3 to 8 hours has also been used, and these models can only predict the traffic flow only for one and two hours of the future. In mid-term models, only use external predictive features, and there is no time horizon limitation. Finally, the accuracy of these two sets of models is evaluated and compared. The data used in this study are related to traffic data of Tehran-Saveh, a rural road in Iran, for both directions. In general, identifying the dominant pattern of traffic parameters in rural roads is more complicated than the urban roads because in contrast to urban trips, a significant part of rural trips is nonroutine.

This study’s contribution is to propose a stacking and bagging ensemble learning process consisting of three base ML algorithms, including KNN, RF, and RNN, alongside the XGboost as a supermodel that puts predictions of base models together. Although previous studies use ensemble learning process for traffic parameter prediction, but designed architecture used in this study is unique. XGboost is a significant part of this structure which is recommended to be used as a stacking supermodel which is not used in the architecture of previous studies related to traffic parameter prediction. Also, short-term and mid-term models with different time horizons and different predictive features are trained and evaluated in this paper for rural road that less investigated before. Finally, employing cyclical feature which are related to temporal features is another novel idea for traffic flow prediction.

2. Data

This study’s traffic data is collected for one section of the Saveh-Tehran rural road for both directions by loop detectors. Data collection has been carried out for about three years, from 21 March 2017 to 10 March 2020. Data are divided into three sections: first, two years of observations are used to train base models, the next six months, and related predictions of base models are used to train the stacking model. The last six months are used to test the base models and stacking model performances. We called these datasets train 1, train 2, and test. Also, total observations for the ensemble learning process, including train 1 and train 2 datasets, are named train datasets. The raw data includes hourly traffic flow and date. After exploring the relationship between hourly traffic flow and calendar attributes such as
holidays and their type, new features related to the calendar are added to the dataset. Since holidays in Iran are based on two lunar and solar calendars, and as these two calendars are not fixed together, both of them are considered. Also, many passengers start their trips before the holiday and continue it until after the holiday, so it is necessary to consider the effects of holidays on the traffic flow of the days before and after it. Weather condition is another important factor affecting the traffic flow, which is extracted and added to the features. Table 1 describes the candidate features to predict traffic flow in the dataset.

In Table 1, season, solar and lunar months, day of solar and lunar months, day of the week, and time of day (temporal features) are essentially cyclical and varied in particular intervals. For instance, hour 23 and hour 0 are close to each other. This also refers to the spring and winter, the first month of the year and the last month of the year and the first day of the week, and the last day of the week. The biggest problem is letting the algorithms know that these features varied in cycles. Calculating the components of the sinus and cosine and introducing cyclical characteristics is the best way to deal with this problem. For this purpose, the following sinus and cosine transformations are used [27].

\[
x_{\sin} = \sin \left( \frac{2 \pi \cdot x}{\text{max}(x)} \right),
\]

\[
x_{\cos} = \cos \left( \frac{2 \pi \cdot x}{\text{max}(x)} \right).
\]

The scatter graph of temporal features after these transformations is shown in Figure 1.

Season, solar and lunar months, day of solar and lunar months, day of the week, and time of day are used cyclically in this study.

The introduced features in Table 1 are used to train the mid-term models with the unlimited prediction time horizon. In short-term models, in addition to the features in Table 1, the traffic flow observed at intervals 3 to 8 hours ago is also used as predictor features and these models are only able to predict one and two hours of the future.

Figure 2 depicts the traffic flow histogram for the ensemble learning train dataset (train 1 + train 2) and the test dataset. Table 2 presents a statistical summary description of traffic flow.

In current study, to prepare and select predictive features, cyclical features have been used. There are several input data selection methods for this purpose. For example, genetic algorithm, forward or backward feature selection, and recursive feature elimination [28]. In the rest of this paper, the effect of using cyclical features have been presented.

3. Methods

This study proposes a stacking and bagging ensemble learning process consisting of three base ML algorithms, including KNN, RF, and RNN, alongside the XGboost as a supermodel. We choose base models based on their accuracy and selected base model outperforms other models. For example, we tried to employ LSTM algorithm as a deep learning base model but the resulted predictions have not enough accuracy to consider LSTM in ensemble learning process.

3.1. K-Nearest Neighbors. The KNN model is an ML method used for both classification and regression problems. The main objective of the KNN is to find some labeled observations in the training dataset which have the smallest distance with nonlabeled observations in the test data. Using the averaging or voting, the new label is assigned to new data [29]. The four main steps of this approach are as following:

Step 1: the train dataset is given in an n-dimensional coordinate system (n is the number of features).

Step 2: Euclidean distance between any new observation and training data observations is calculated.

Step 3: k is the number of observations that have the smallest distance from any new observation.

Step 4: the average of K observation labels is selected as the new observation label.

Euclidean distance between observations \(p\) and \(q\) is defined according to equation (2) [30].

\[
d(p, q) = \sqrt((q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2)
\]

\[
= \sqrt{n \sum_{i=1}^{n} (q_i - p_i)^2}.
\]

3.2. Random Forest. Similar to the KNN, the RF is an ML model used for regression and classification problems. The RF consists of a large number of decision trees. In this model, the training data are divided between decision tree models, and after training them, predictions are made for each decision tree. The average of predictions is determined as the RF’s final prediction [31]. The following steps indicate how the algorithm works.

Step 1: start with the select random samples from the training dataset

Step 2: using each sample to train a decision tree.

Step 3: the prediction of each decision tree model is made for the test data.

Step 4: the average of predictions is selected as the final prediction.

RF starts with a node and branches to another node. This paper uses the entropy formula to determine how the dataset branches from each node. Equation (3) presents the entropy formula [31].

\[
\text{Entropy} = \sum_{i=1}^{c} -p_i \cdot \log_2 (p_i),\]

where \(p_i\) is the relative frequency of label \(i\), \(i\) is the index of labels, and \(c\) is the total number of labels.
3.3. Recurrent Neural Network. RNN is a kind of deep neural network. Since the successive observations are dependent on each other, the use of the RNN can help improve the accuracy of predictions. These ANNs are particularly useful for time-series analysis, where each neuron can maintain internal information of the connected nodes. This attribute of maintaining the internal state or the memory capability helps the network to understand and discover the link between different successive observations [32].

Let denote the input time series with \( D \) variables of length \( T \) as \( X = (X_1, X_2, \ldots, X_T) \), where \( X_t \) is the \( t \)-th observation. \( c_t \) is a memory cell, contains information at time step \( t \), and is controlled by three gates. These gates control whether to forgot the current cell value (forget gate \( f_t \)) to read its input (input gate \( i_t \)) and to output the new cell value (output gate \( o_t \)) [33]. Also, \( \tilde{c}_t \) is an input modulation gates. All these gates, cell update, and output are computed in the following formulas [34]:

### Table 1: Description of candidate features to predict traffic flow.

<table>
<thead>
<tr>
<th>Feature name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Season</td>
<td>Including spring, summer, fall, and winter</td>
</tr>
<tr>
<td>Solar month</td>
<td>Including 12 solar months</td>
</tr>
<tr>
<td>Lunar month</td>
<td>Including 12 lunar months</td>
</tr>
<tr>
<td>Day of a solar month</td>
<td>Including 29–31 days of a solar month</td>
</tr>
<tr>
<td>Day of a lunar month</td>
<td>Including 29–30 days of a lunar month</td>
</tr>
<tr>
<td>Time of day</td>
<td>Including 24 hours a day</td>
</tr>
<tr>
<td>6 hour before the holidays</td>
<td>Equal to 1 if it is 1 to 6 hour before holidays</td>
</tr>
<tr>
<td>6 hour after the holidays</td>
<td>Equal to 1 if it is 1 to 6 hour after holidays</td>
</tr>
<tr>
<td>Day or night</td>
<td>Including day and night</td>
</tr>
<tr>
<td>Number of holidays</td>
<td>The number of sequential holidays</td>
</tr>
<tr>
<td>Holidays</td>
<td>Includes 1 for holidays and 0 for other days</td>
</tr>
<tr>
<td>Holiday type</td>
<td>Type of holidays</td>
</tr>
<tr>
<td>Holiday in three days later</td>
<td>Equal to 1 if three days later is a holiday</td>
</tr>
<tr>
<td>Type of holidays in three days later</td>
<td>Including the holiday type of three days later if it is a holiday, otherwise equals 0.</td>
</tr>
<tr>
<td>Holiday in three days ago</td>
<td>Equal to 1 if three days ago is a holiday</td>
</tr>
<tr>
<td>Type of holidays in three days ago</td>
<td>Including the holiday type of three days ago if it is a holiday, otherwise equals 0.</td>
</tr>
<tr>
<td>Holiday in two days later</td>
<td>Equal to 1 if two days later is a holiday</td>
</tr>
<tr>
<td>Type of holidays in two days later</td>
<td>Including the holiday type of two days later if it is a holiday, otherwise equals 0.</td>
</tr>
<tr>
<td>Holiday in two days ago</td>
<td>Equal to 1 if two days ago is a holiday</td>
</tr>
<tr>
<td>Type of holidays in two days ago</td>
<td>Including the holiday type of two days ago if it is a holiday, otherwise equals 0.</td>
</tr>
<tr>
<td>Holiday in a day later</td>
<td>Equal to 1 if a day later is a holiday</td>
</tr>
<tr>
<td>Type of holidays in a day later</td>
<td>Including the holiday type of a day later if it is a holiday, otherwise equals 0.</td>
</tr>
<tr>
<td>Holiday in a day ago</td>
<td>Equal to 1 if a day ago is a holiday</td>
</tr>
<tr>
<td>Type of holidays in a day ago</td>
<td>Including the holiday type of a day ago if it is a holiday, otherwise equals 0.</td>
</tr>
</tbody>
</table>
it = \sigma(W_xiX_t + W_{i0}h_{t-1}),
\bar{c_t} = \phi(W_xcX_t + W_{c0}h_{t-1}),
\bar{c_t} = f_t \odot c_{t-1} + h_t \odot \bar{c_t},
h_t = o_t \odot \phi(c_t),
\label{eq:i_ct_ht}

where \odot indicates scalar product, W s are the network parameters matrices, h_t is the hidden state, \phi is the hyperbolic tangent function, and \sigma denotes the standard logistics sigmoid transfer function.

3.4. Bagged Averaging. After training KNN, RF, and RNN, the predicted traffic flow is given to the ensemble learning algorithms to determine the final prediction. Bagged averaging is one of these algorithms that can be done weighted or simple. In the weighted method, each model’s prediction weight is inversely related to the model’s root mean square error (RMSE). Equation (5) shows how weights in bagged averaging are calculated.

\begin{equation}
W_i = \frac{1}{\sum_{i=1}^{I} RMSE_i}
\end{equation}

where \(W_i\) is prediction weight of model \(i\), \(I\) is the total number of models, and \(RMSE_i\) is the root mean square error of model \(i\).

3.5. Stacking XGBoost. XGBoost is an optimized variant of the ensemble learning model that has improved and expanded from the tree model of gradient boosting. Under the
The gradient boosting paradigm, it applies ML algorithms. XGBoost offers a parallel tree boost that easily and reliably addresses several data science issues [35]. The boosting tree is defined as follows:

$$\hat{y}_i = \sum_{k=1}^{n} f_k(x_i), f_k \in F,$$

where $F$ is the set of decision trees, $\hat{y}_i$ is model prediction, $x_i$ is a set of predictor features, and $n$ is the number of trees. The loss function of the model is as follows:

$$L(t) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k),$$

where $\gamma$ and $\lambda$ are penalty coefficients. XGBoost aims to minimize the differentiable function. By rewriting the differentiable function and Taylor expansion, the formula is as follows:

$$L(t) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k),$$

$$\approx \sum_{i=1}^{n} \left[ (y_i - \hat{y}_i^{(t-1)}) + g_i f_i(x_i) + \frac{1}{2} h_i f_i(x_i)^2 \right] + \sum_{k=1}^{K} \Omega(f_k),$$

where $g_i$ and $h_i$ are the first and second derivatives of the loss function, respectively [36].

4. Results and Discussion

4.1. Base Models Results. In the first step to train the KNN, RF, and RNN, selecting proper values for model parameters has a significant effect on the final accuracy of prediction. These parameters include the number of neighbors ($K$) in the
KNN, the number of trees (NT), and the number of variables randomly sampled as candidates at each split (NV) in RF, and the number of hidden layers (N) in the neural network model.

To find the optimal value of these parameters after assigning different values to them, models are trained. Accuracy for the test dataset is evaluated based on the RMSE. Equation (10) represents how to calculate the RMSE. Figures 3 and 4 show

<table>
<thead>
<tr>
<th>Table 3: Selected optimal values for models parameters.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time horizon</td>
</tr>
<tr>
<td>---------------</td>
</tr>
<tr>
<td>Mid-term</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Short-term</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Time horizon</th>
<th>Model</th>
<th>Road</th>
<th>RMSE</th>
<th>MAPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mid-term</td>
<td>RF</td>
<td>Saveh to Tehran</td>
<td>775</td>
<td>27.14</td>
</tr>
<tr>
<td></td>
<td>KNN</td>
<td>Tehran to Saveh</td>
<td>729</td>
<td>21.23</td>
</tr>
<tr>
<td></td>
<td>RNN</td>
<td>Tehran to Saveh</td>
<td>1060</td>
<td>33.58</td>
</tr>
<tr>
<td>Short-term</td>
<td>RF</td>
<td>Tehran to Saveh</td>
<td>453</td>
<td>16.61</td>
</tr>
<tr>
<td></td>
<td>KNN</td>
<td>Saveh to Tehran</td>
<td>535</td>
<td>15.25</td>
</tr>
<tr>
<td></td>
<td>RNN</td>
<td>Tehran to Saveh</td>
<td>914</td>
<td>34.59</td>
</tr>
</tbody>
</table>

KNN, the number of trees (NT), and the number of variables randomly sampled as candidates at each split (NV) in RF, and the number of hidden layers (N) in the neural network model. Figures 3 and 4 show the results of these experiments.
the sensitivity analysis performed to find optimal values of the short-term and mid-term models’ parameters.

\[
RMSE = \sqrt{\frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{n}}, \tag{10}
\]

where \(\hat{y}_i\) and \(y_i\) are predicted and actual values, and \(n\) is the number of observations.

Table 3 shows selected optimal values for final models. After training the final models to assess the accuracy of predictions on the test dataset, in addition to the RMSE, the mean absolute percentage error (MAPE) is used. Equation (11) shows how MAPE is calculated. Table 4 presents the obtained values of error metrics for the final models.

\[
MAPE = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{\hat{y}_i - y_i}{y_i} \right|. \tag{11}
\]

Results in Table 4 shows that for both short-and mid-term models and both directions of Saveh–Tehran road, the lowest error prediction is achieved by the RF, and then the KNN has the highest prediction error. The MAPE of the mid-term RF for the Saveh to Tehran and Tehran to Saveh is 21.23 and 27.14, respectively. Also, in the short-term model, the MAPE of RF for the Saveh to Tehran and Tehran to Saveh is 15.25 and 16.61.

Figure 5 shows the difference between the RMSE of the short-term and mid-term models. The accuracy of the short-term models is higher than the mid-term models, and using previously observed traffic flows had increased the accuracy.
of the prediction. The limited-time horizon of these models is considered as their weakness.

4.2. Bagging and Stacking Ensemble Models Results. After receiving the base models’ predictions, the ensemble learning process is performed by using the bagging and stacking methods, and the final results are obtained. In addition to the bagging and stacking methods, the maximum and minimum predicted traffic flow values are analyzed as the final prediction. Like the base models, the ensemble learning process has also been examined for short-term and mid-term predictions that their inputs are the short-term and mid-term output of base models. Table 5 shows the results obtained by ensemble learning and the RF as the most accurate base model.

Table 5 indicates that for both the short-term and mid-term models and both directions of Tehran–Saveh road, based on the RMSE and the MAPE, using the XGBoost decreases the prediction error and stacking ensemble learning by using XGBoost has the lowest prediction error. Based on the RMSE, in the mid-term model, the predictions through maximum and minimum values of the predicted traffic flow values have higher and lower accuracy compared to the RF, respectively. It can be concluded that the base models underestimate traffic flow. Bagged averaging only increases the accuracy of predicting for Tehran to Saveh. In the short-term models, only the XGBoost model has reduced the traffic volume prediction error, and other methods have no positive effects on the accuracy of traffic flow prediction.

Another critical point in the traffic flow prediction is predicting maximum and minimum traffic flow values that indicate rare traffic events. Generally, informing hours with high and low traffic flow is more worthwhile for users and system operators than normal traffic flows. To determine the models’ performance in predicting rare events, the RMSE has been calculated separately for the first and fourth quartiles of the observed traffic flow and presented in Figures 6 and 7.

Figure 6: RMSE of base, bagging, and stacking models for the first quartiles of the observed traffic flow. (a) Mid-term models of Tehran to Saveh. (b) Mid-term models of Saveh to Tehran. (c) Short-term models of Tehran to Saveh. (d) Short-term models of Saveh to Tehran.
Figures 6 and 7 show the lowest RMSE for the first and fourth quartiles are achieved by XGBoost and Max methods, respectively. The exciting point is less prediction error of the XGBoost than the Min method in predicting the first quartile. The XGBoost could predict both the first and fourth quartile more accurately than the base models, whereas the Max method only predicts the fourth quarter more accurately than the base models. Among the base models, the RF model predicts the traffic flow for two quadrants more accurately than the two other base models.

5. Conclusion

One of the applications of intelligent transportation systems is predicting the future state of traffic while the traveler will have more proper planning to choose travel, departure time, and route choice. Also, the transportation network operator will be more prepared to deal with traffic congestion. In this study, traffic flow as a parameter shows the state of traffic is predicted using three base methods based on ML, including KNN, RF, and RNN for a rural road in Iran for both directions. Then, using the bagging and stacking methods, the most important of them is the XGBoost, and the final traffic flow is predicted. Pre-processing is performing by adding predictor features related to cyclical temporal features, holidays, types of holidays, and weather in the first step. In the second step, to find optimal values of the parameters of short-term and mid-term models, models are trained by different values of parameters, and optimal values are selected based on the accuracy of prediction on the test data. After training the base models with optimal values of parameters, the initial predictions are evaluated and compared. In the next step, by using base models’ predictions, the ensemble learning process is applied to make the final prediction, which is expected to be more accurate than base models predictions. The results show that the highest accuracy of prediction for both short-term and mid-term is achieved...
using the XGBoost model in the stacking learning process. This model predicts the first and fourth quartiles of the observed traffic flow more accurately than the base models. In general, the prediction error of short-term models is lower than the mid-term models. However, these models can only predict the traffic flow of one and two hours of the future.

In the end, the predicted traffic flow by short-term and mid-term models can be informed to passengers via advanced traveler information systems. To use the prediction accuracy of the short-term models and have the prediction time horizon of mid-term models, future one and two hours will be predicted by short-term models, and for the next hours, prediction by mid-term models can be used.

**Data Availability**

The traffic data used in this study are available from the corresponding author upon reasonable request.

**Conflicts of Interest**

The authors declare no conflicts of interest.

**Authors’ Contributions**

A. R. contributed to software analysis, validation, formal analysis, data curation, original draft preparation, and visualization. S. S. contributed to conceptualization, methodology, supervision, and review and editing.

**References**


