

Research Article

Short-Term Prediction of Traffic State for a Rural Road Applying Ensemble Learning Process

Arash Rasaizadi ¹, Seyedehsan Seyedabrishami ¹ and Mohammad Saniee Abadeh ²

¹School of Civil and Environmental Engineering, Tarbiat Modares University, Tehran, Iran

²School of Electrical & Computer Engineering, Tarbiat Modares University, Tehran, Iran

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

Received 8 June 2021; Accepted 17 December 2021; Published 29 December 2021

Academic Editor: Socrates Basbas

Copyright © 2021 Arash Rasaizadi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Short-term prediction of traffic variables aims at providing information for travelers before commencing their trips. In this paper, machine learning methods consisting of long short-term memory (LSTM), random forest (RF), support vector machine (SVM), and K-nearest neighbors (KNN) are employed to predict traffic state, categorized into A to C for segments of a rural road network. Since the temporal variation of rural road traffic is irregular, the performance of applied algorithms varies among different time intervals. To find the most precise prediction for each time interval for segments, several ensemble methods, including voting methods and ordinal logit (OL) model, are utilized to ensemble predictions of four machine learning algorithms. The Karaj-Chalus rural road traffic data was used as a case study to show how to implement it. As there are many influential features on traffic state, the genetic algorithm (GA) has been used to identify 25 of 32 features, which are the most influential on models' fitness. Results show that the OL model as an ensemble learning model outperforms machine learning models, and its accuracy is equal to 80.03 percent. The highest balanced accuracy achieved by OL for predicting traffic states A, B, and C is 89, 73.4, and 58.5 percent, respectively.

1. Introduction

Sustainable transportation networks need to use data obtained from intelligent transportation systems (ITS) to relieve traffic congestion and its consequences, such as air and noise pollution and wasting energy and time. Intelligent traffic congestion alleviation is a vital element of smart mobility and smart transportation systems [1]. One of the intelligent transportation systems is the advanced traveler information system (ATIS). ATISs provide useful information about the current or future traffic conditions to travelers and transportation agencies [2]. These systems' effectiveness is more when predicting the future state of the transportation network and letting users have better plans for their next trips [3]. A group of travelers who plan to travel during traffic peak hours will more likely postpone or cancel their trips. These changes lead to more balanced distributed trips over time and a more sustainable transportation network. Traffic volume and

average speed are well-known continuous traffic variables that can be predicted [4, 5]. Many users are unable to understand the performance condition of the transportation network by knowing these variables. The traffic volume to capacity ratio and average speed to free-flow speed ratio are more informative and meaningful for users [6]. Instead of predicting traffic volume and speed, we can predict the traffic state as a nominal traffic variable. This variable is determined regarding the volume to capacity ratio and the speed to free-flow speed ratio.

Another critical point is employing appropriate models that are compatible with the nominal nature of the traffic state. Predictive methods are diverse, and there is no superior model for every prediction problem [7]. Generally, predictive techniques are divided into naïve, time series, and machine learning [8].

Naïve methods are simple with short computational time. These methods do not react to dynamic changes and

usually are used as a benchmark [8]. Time-series methods (also known as parametric or statistical methods) have a well-established theoretical background and show the importance and effect of independent variables on the dependent variable by estimating coefficients and t states [9]. However, one of these models' deficiencies is the inability to depict nonlinear relationships because of the assumption's limitations [10]. By increasing the volume of the dataset, these methods need more computational processing power. Also, these methods concentrate on means and miss the extremes [11]. The family of time-series methods includes autoregression (AR), moving average (MA), autoregressive moving average (ARMA), autoregressive integrated moving average (ARIMA), and seasonal autoregressive integrated moving average (SARIMA) [12]. Alghamdi et al. [13] leverage ARIMA-based modeling to forecast traffic congestion. Analyzing nonstationary and nonnormally distributed traffic data by ARIMA achieves appropriate performance with a confidence level of 95%. Ding et al. [14] forecast subway ridership by ARIMA-GHARCH. The proposed model has a more accurate prediction compared to the ARIMA-only model.

Machine learning methods (also known as nonparametric methods) are capable of mapping nonlinear relationships. These methods are more suitable for analyzing big data and having no or fewer assumptions [15]. The main disadvantages of machine learning methods are lack of interpretability, needing many observations to train, and working like a black box [16]. Learning models based on neural networks such as long short-term memory (LSTM) [17], SVM [18], K-nearest neighbor (KNN) [19], and random forest (RF) [20] are some machine learning methods to predict traffic variables. Du et al. [21] predict traffic passenger flow for urban areas and propose a deep irregular convolutional residual LSTM network (DST-ICRL). By using both short-term and long-term historical data, the proposed method outperforms traditional machine learning methods. To predict traffic flow, Kang et al. [22] use LSTM recurrent neural network. They conclude that occupancy, speed, and downstream and upstream traffic information as predictor variables can enhance prediction accuracy. A spatiotemporal correlative K-nearest neighbor is proposed by Cai et al. [23]. Gaussian weighted Euclidean finds the nearest neighbors. Also, considering the relationship between road segments improves model performance. Using the RF model, Liu and Wu [24] predict traffic congestion. The weather conditions, time, special road conditions, road quality, and holiday are used in the predictive RF model. Li et al. [25] propose a combined WT-FCBF-LSTM (wavelet transform, fast correlation-based filter, and long short-term memory) model to predict passenger demand by hybrid ridesharing service models. The proposed model has better performance in terms of accuracy compared to single LSTM and single WT-LSTM models.

Diverse prediction methods have different advantages and disadvantages. Ensemble learning is a process that receives model predictions as input and makes a unique final prediction. Moretti et al. [26] predict traffic flow by using a statistical and neural network bagging ensemble hybrid

model, which outperforms the prediction of input methods. Also, Yang et al. [27] show that the gradient boosting decision trees (GBDT) bring more prediction accuracy than the SVM and backpropagation neural networks for their case study.

In the current study, the hourly traffic state consisting of light, semiheavy, and heavy traffic is predicted for one section of Karaj-Chalus, a rural road in Iran. Many features related to traffic state variation are extracted and used as predictor features. One of the essential parts of modeling is feature selection. Features could be selected by try and error, but there are some systematic methods. In this study, by using the genetic algorithm (GA), influential features are selected systematically. The next step is to employ machine learning methods. Several machine learning methods, including LSTM as a deep learning approach for time-series prediction, KNN, SVM, and RF, are trained to predict traffic states. Finally, ensemble methods, including OL and four voting methods, convert LSTM, KNN, SVM, and RF predictions to one final prediction. It is expected that ensemble methods provide more accurate predictions compared to the initial predictive methods.

Compared to traffic volume and speed, nominal traffic state is more informative for travelers. It can be shown in traffic maps by the easily understandable colors. Travelers decide about departure time and trip route by information obtained from advanced traveler information systems. Also, transportation agencies can benefit from accurate traffic state predictions. It is vital to provide accurate predictions at any time. It motivates us to propose an ensemble learning process that is expected to have more stable performance in terms of prediction accuracy than single models.

The first contribution of this paper is related to the data. We add new features related to date, solar and lunar calendars, weather conditions, and road blockage. Also, we predict traffic state as a nominal variable, investigate rural traffic data with nonroutine trips, and use Iran's traffic data as a developing country. Second, the feature selection is made by GA and two datasets train models; the first one includes selected features by GA, and the second one contains all features. The accuracy of the models for each dataset is calculated and compared. The ensemble learning process by OL and voting methods is another contribution of the current study. The performance of single machine learning and ensemble learning algorithms for hourly traffic state prediction is evaluated in different evaluation metrics.

2. Data

Karaj-Chalus is a rural road in the north of Iran, a part of a route from Tehran, the capital of Iran, to the seaside. The length of this road is 170 kilometers. In addition, there are three parallel roads with this road but with different lengths. Many trips to Chalus are recreational and nonmandatory. These nonroutine trips make the prediction more difficult because finding traffic patterns is not simply compared to routine trips. Figure 1 shows the map of Karaj-Chalus road.

The purpose of this paper is to predict the hourly traffic state. Traffic state is a more informative variable for travelers

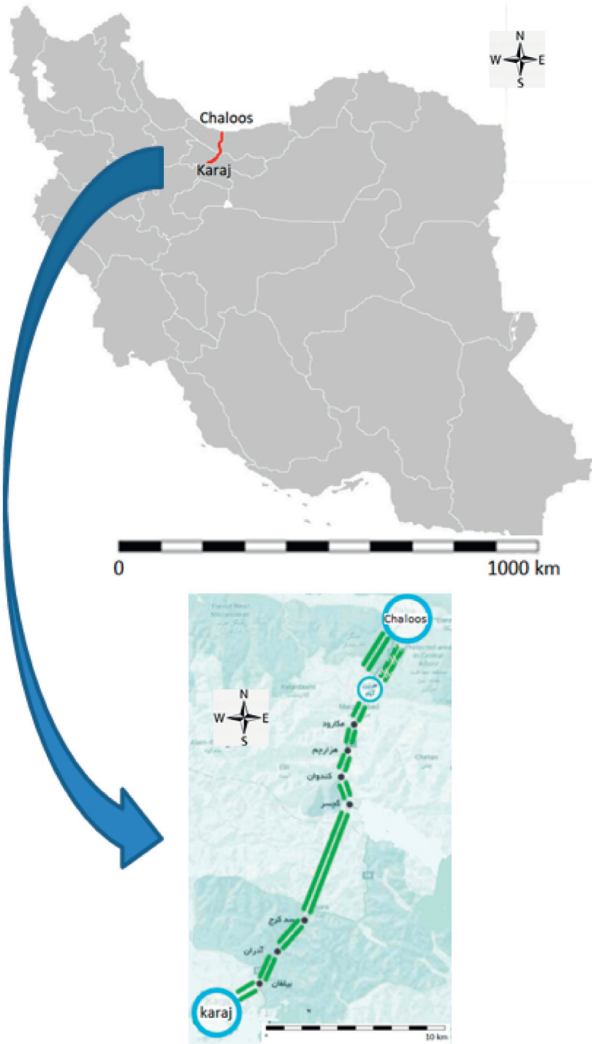


FIGURE 1: Map of Karaj-Chalus road in Iran.

who do not know other characteristics of the road. Loop detectors in one section of this road collect hourly traffic volume and hourly average speed. By calculating the hourly traffic volume over the hourly capacity ratio and hourly average speed over the free-flow speed ratio, the hourly traffic state is determined based on Table 1. A, B, and C represent light, semiheavy, and heavy traffic, respectively. This type of traffic state definition is provided by Iran Road Maintenance and Transportation Organization (RMTO, <http://www.rmto.ir/>).

The raw data only has hourly traffic state, hourly traffic volume, hourly average speed, and date. One of the essential steps before training models is extracting effective features. Traffic patterns of nonroutine trips are affected by holidays and different types of holidays have different effects. Holidays in Iran are based on solar and lunar calendars. Since solar and lunar calendars are not fixed together, both of them are considered simultaneously. So, several features related to holidays and their types are defined based on lunar and solar calendars. Police often blocked this road in each direction or parallel roads for traffic management at peak hours. Therefore,

TABLE 1: Determining hourly traffic state.

S/S_f^*	V/C^*					
	Under 0.1	0.1–0.3	0.3–0.5	0.5–0.7	0.7–0.9	Over 0.9
Over 0.95	A	A	A	B	B	C
0.8–0.95	A	A	B	B	B	C
0.6–0.8	A	B	B	B	C	C
0.45–0.6	B	B	B	C	C	C
Under 0.45	C	C	C	C	C	C

*V and C are hourly traffic volume and the hourly capacity and V/C is their ratio. S and Sf are hourly average speed and the free-flow speed and S/Sf is their ratio. More V/C equals fewer S/Sf and more heavy traffic. Less V/C equals more S/Sf and less congested traffic. Thresholds are defined based on observed traffic patterns and RMTO experts' judgment. For example, S/Sf under 0.3 is rarely observed, so it is not necessary to consider it as a threshold.

blockage of the road, blockage of opposite direction, and blockage of parallel roads are added to the dataset as features. Table 2 shows all the features, which are extracted in this study.

This data is collected for 17 months, from March 2017 to August 2018. The first 12 months of the dataset are used for training single models (train dataset 1), the OL model is calibrated by the next three months (train dataset 2), and the last two months (test dataset) is used to test the predictions of single and ensemble methods. Also, two months of train dataset 1 is used for cross-validation to tune the models' parameters and evaluate the performance of models in a robust manner. For this purpose, another method is using Monte-Carlo simulation [28]. The total number of observations is 11353. Table 3 shows the frequency of traffic states for each part of the dataset. Pie charts in Figure 2 show the characteristics of candidate features.

All features are used to train the models, but some of these features may have less effect on the models' prediction power or even have a negative effect on prediction. In this study, to select effective predictors systematically and include them in models, GA is used. The following procedure is employed for feature selection [29]:

- Step 1: define population size (P) for each generation, mutation probability (pm), and stopping criteria.
- Step 2: randomly generate an initial population of chromosomes.
- Step 3: repeat until the stopping criterion is met:
 - For each chromosome, do
 - Tune and train the classifier model and compute each chromosome's fitness
 - End.
 - For each reproduction 1 to P/2, do
 - Select two chromosomes based on fitness.
 - Crossover: randomly select a locus and exchange genes (a mechanism to form new genes) on either side of the locus to produce two-child chromosomes with mixed genes.
 - Mutate the child chromosomes with probability pm.
 - End.

TABLE 2: Description of extracted candidate features to predict traffic state.

Feature name	Description
Season	Including spring, summer, fall, and winter
Solar month	Including 12 solar months
Lunar month	Including 12 lunar months
Day of a solar month	Including 29–31 days of a solar month
Day of a lunar month	Including 29–30 days of a lunar month
Time of day	Including 24 hours a day
Six hours before holidays	Equal to 1 if it is 1 to 6 hours before holidays
Six hours after holidays	Equal to 1 if it is 1 to 6 hours after holidays
Day or night	Including day and night
Number of holidays	The number of sequential holidays
Holidays	Including 1 for holidays and 0 for other days
Holiday type	Type of holidays
Holiday in three days later	Equal to 1 if three days later is a holiday
Type of holidays in three days later	Including the holiday type of three days later if it is a holiday; otherwise, it equals 0
Holiday in three days ago	Equal to 1 if three days ago is a holiday
Type of holidays in three days ago	Including the holiday type of three days ago if it is a holiday; otherwise, it equals 0
Holiday in two days later	Equal to 1 if two days later is a holiday
Type of holidays in two days later	Including the holiday type of two days later if it is a holiday; otherwise, it equals 0
Holiday in two days ago	Equal to 1 if two days ago is a holiday
Type of holidays in two days ago	Including the holiday type of two days ago if it is a holiday; otherwise, it equals 0
Holiday in a day later	Equal to 1 if a day later is a holiday
Type of holidays in a day later	Including the holiday type of a day later if it is a holiday; otherwise, it equals 0
Holiday in a day ago	Equal to 1 if a day ago is a holiday
Type of holidays in a day ago	Including the holiday type of a day ago if it is a holiday; otherwise, it equals 0
Weather condition	Including sunny, rainy, and snowy
Blockage	Blockage of the road by police
Blockage of the opposite direction	Blockage of the opposite direction by police
Blockage of parallel paths	Blockage of parallel paths by police

TABLE 3: Frequency of traffic states in observations.

Traffic state	Train dataset 1		Train dataset 2		Test dataset	
	Frequency	Frequency percentage	Frequency	Frequency percentage	Frequency	Frequency percentage
Light (A)	3708	46.38	741	31.55	170	16.88
Semiheavy (B)	3822	47.80	1365	58.11	693	68.82
Heavy (C)	465	5.82	243	10.34	144	14.30

Chromosomes, which consist of genes, are binary vectors with 1 representing a feature's presence and 0 its absence. The population is a set of chromosomes (solutions). In the reproduction algorithm, the two-parent chromosomes are split at a random position, and the head of one chromosome is combined with the tail of the other chromosome [29]. The prediction accuracy (fitness) of an internal decision tree is the objective function.

This procedure is implemented in the R software. Among all features, seven features are not qualified by GA. These features are as follows:

- (1) Type of holidays in a day later.
- (2) Holiday in a day later.
- (3) Holiday in a day ago.
- (4) Holiday in three days later.
- (5) Blockage.
- (6) Blockage of the opposite direction.
- (7) Blockage of parallel paths.

Models are trained by selected features by GA and all of the features.

3. Methodology

3.1. Long Short-Term Memory. Recurrent neural networks (RNNs) are deep artificial neural network (ANN) models that keep information in memory. These models consider the dependency between sequential observations. The chief defect of these models is that they only consider short-term dependencies because the gradient of loss function declines exponentially over time. LSTM is a kind of RNN that can handle the long-term dependencies alongside short-term dependencies [30, 31].

The LSTM structure consists of four gates (neural network layers), forget, remember, learn, and output gates. The LSTM model's inputs are long-term memory, short-term memory, and training example (new data). The long-term input goes into the forget gate (1), which decides to forget irrelevant parts. The short-term and training example inputs

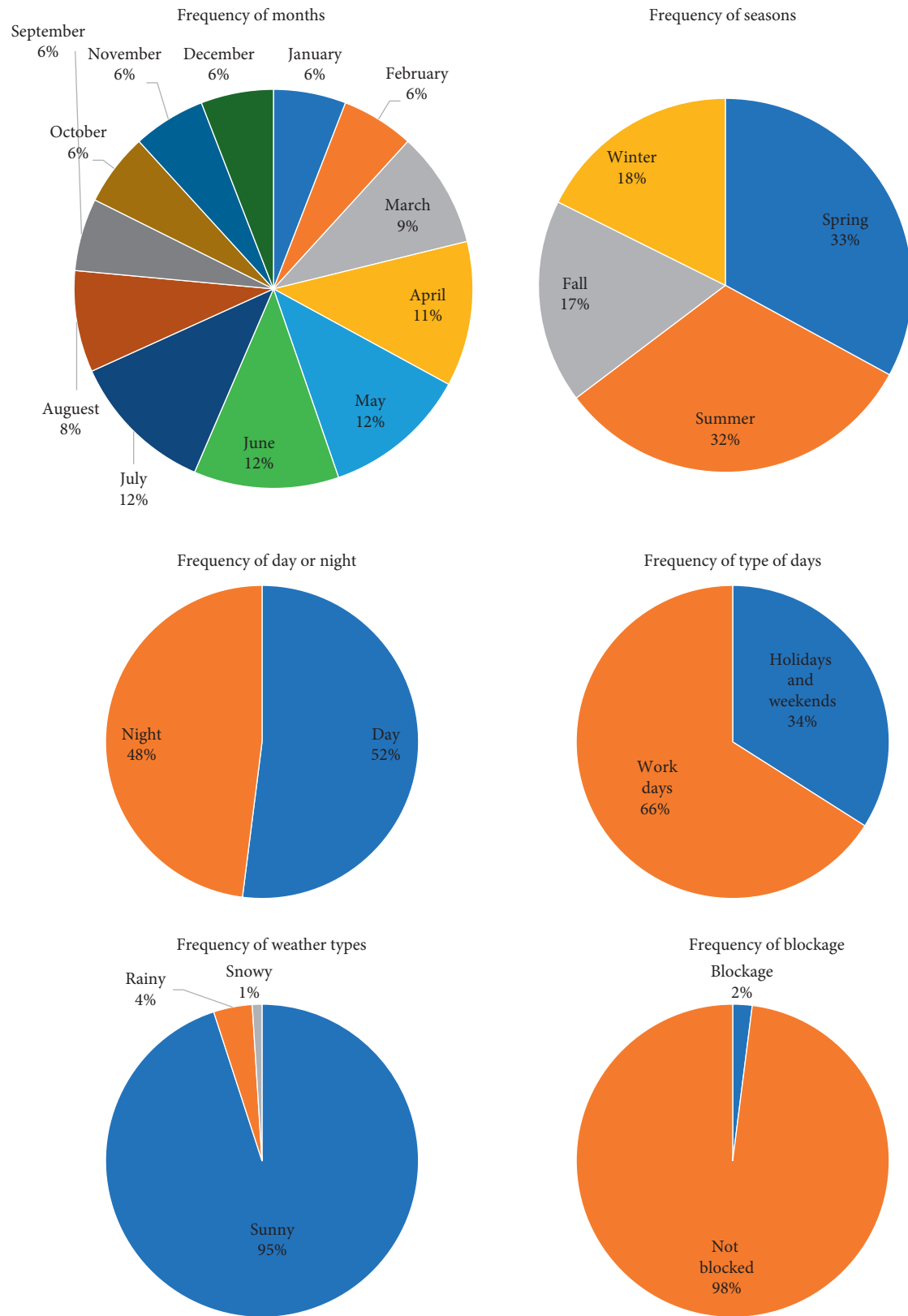


FIGURE 2: Characteristics of candidate features.

go into the learn gate (2), which determines what inputs are to be learned. Passed information (consisting of short- and long-term memories) from forget and learn gates goes into the remember gate (3), producing new long-term memories for the output gate. Finally, the output gate (4) updates short-term memories and the model's final output [31]. The equations of gates in LSTM are

$$f_t = \sigma(w_f [h_{t-1}, x_t] + b_f), \quad (1)$$

$$l_t = \tanh(w_n [h_{t-1}, x_t] + b_n), \quad (2)$$

$$r_t = l_t + f_t, \quad (3)$$

$$o_t = \sigma(w_o [h_{t-1}, x_t] + b_o), \quad (4)$$

$$\sigma(t) = \frac{1}{1 + e^t}. \quad (5)$$

f_t , l_t , r_t , and o_t are factors of forget, learn, remember, and output gates, respectively. σ is the sigmoid function (5). w_x is the weight for the gate(x) neurons. h_{t-1} is the output of the previous LSTM block. x_t is the input at the current timestamp and b_x is the bias for the gate(x). Figure 3 shows the architecture of the LSTM network.

3.2. Support Vector Machine. SVM is a supervised machine learning classifier used for classification and regression (SVR) problems. This model finds a hyperplane in an N-dimensional space to classify the data points distinctly. The model finds a hyperplane with the maximum distance between data points of classes (support vectors). The loss function of SVM to maximize the margin is hinge loss. Future data can be classified based on their position relative to that hyperplane [33].

In many real situations, the data is not linearly separable. Applying a transformation by the kernel function is essential before classification. This study uses the radial basis function (RBF) kernel function among different kernel functions. The formulation of RBF function is as (6) [34]:

$$K(X_i, X) = \exp\left(-\frac{X_i - X_j^2}{2\sigma^2}\right), \quad (6)$$

where σ is a free parameter to be calibrated. $X_i - X_j^2$ is the squared Euclidean distance between the two feature vectors X_i and X_j .

3.3. K-Nearest Neighbor. The KNN model is a supervised machine learning algorithm for both classification and regression problems. The main idea of KNN is to find a predefined number of training samples (K) closest in the distance to the new point and predicts the class by voting. This algorithm can be summarized in 4 steps [23].

Step 1: store the training samples in an array of data points.

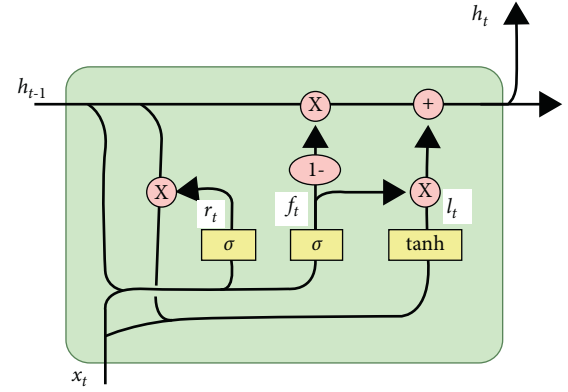


FIGURE 3: Architecture of an LSTM network [32].

Step 2: calculate the distance of training samples and new data point p.

Step 3: find the K smallest distance obtained.

Step 4: return the majority class of K smallest distance.

Euclidean, Manhattan, and Minkowski are well-known distance functions. This paper used Euclidean distance to calculate the distance between data points, X_i and X_j :

$$\begin{aligned} d(X_i, X_j) &= \sqrt{(X_{i1} - X_{j1})^2 + (X_{i2} - X_{j2})^2 + \dots + (X_{in} - X_{jn})^2}, \\ &= \sqrt{\sum_{i=1}^n (X_{in} - X_{jn})^2}. \end{aligned} \quad (7)$$

3.4. Random Forest. RF is a supervised learning algorithm that can be used for both classification and regression. It consists of many individual decision trees that spit out a class prediction, and the class with the most votes becomes our model's prediction. The following steps show how this algorithm works [35]:

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

Step 2: construct a decision tree for every sample.

Step 3: get the prediction result from every decision tree.

Step 4: perform voting for every predicted result.

Step 5: the most voted prediction result is the final prediction result.

Decision trees start with a node and branch to another node. This paper uses the entropy formula to determine how the dataset branches from each node. Equation (8) presents the entropy formula [36].

$$\text{Entropy} = -\sum_{i=1}^c p_i \log_2(p_i), \quad (8)$$

where p_i is the relative frequency of class i , i is the index of classes, and c is the total number of classes.

3.5. Ensemble Learning. At this step, ensemble learning methods put predictions of introduced methods together to provide one unique final prediction. The final prediction is expected to have higher accuracy than the accuracy of LSTM, SVM, KNN, and RF.

Four different voting methods are defined as follows:

Voting to a better state: predictions are the majority vote of contributing models. If majorities are equal, priority is A, B, and C, respectively.

Voting to a worse state: predictions are the majority vote of contributing models. If majorities are equal, priority is C, B, and A, respectively.

Best state: it selects A if at least one model predicts A, else it selects B if at least one predicts B; otherwise, it selects C.

Worst state: it selects C if at least one model predicts C, else it selects B if at least one predicts B; otherwise, it selects A.

Another method is OL, which is a statistical method. In this method, input models' importance for predicting each traffic state is determined by estimating coefficients and t-state. Let us define s_{qi}^* is a linear function consisting of a vector of input variables x_{qi} , corresponding coefficients γ , and random term η_{qi} . q is the index of hour, i is the index of traffic state, and δ_{AB} and δ_{BC} are thresholds [37].

$$s_{qi}^* = \gamma x_{qi} + \eta_{qi}. \quad (9)$$

The final output is A if $s_{qi}^* \leq \delta_{AB}$, is B if $\delta_{AB} < s_{qi}^* \leq \delta_{BC}$, and is C if $\delta_{BC} < s_{qi}^*$.

Model parameters are estimated by maximizing the likelihood function.

$$\log L = \sum_{i=1}^3 M_{qi} \log[\text{Pr}_{qi}], \quad (10)$$

where M_{qi} is equal to 1 if state i occurred in hour q , else zero, and Pr_{qi} is occurrence probability of state i in hour q [37].

Figure 4 shows the proposed ensemble learning process.

All of the models are implemented in the R software.

4. Results and Discussion

LSTM, SVM, KNN, and RF models are trained by train dataset 1. To tune the parameters of models, different values are set for them. The final parameters are selected in terms of the accuracy of predictions on the rest of the dataset. These parameters include K in KNN, the number of trees to grow (NT), and the number of variables randomly sampled as candidates at each split (NV) in the RF model and cost (C) in the SVM model. A summary of parameters tuning of models is presented in Table 4.

After training models by optimum parameters, the accuracy of predictions on train dataset 1 and train dataset 2+test datasets are calculated and presented in Table 5.

According to Table 5, RF, SVM, KNN, and LSTM models can be sorted in terms of accuracy. Results indicated that the RF model trained by the GA features outperforms other models with 78.66% accuracy. Also, SVM and KNN have similar performances, and LSTM predictions have less accuracy than the other models. Using GA features increases the efficiency of LSTM, KNN, and RF models. Reducing data dimension and shortening computational time without eliminating any useful information are other advantages of feature selection by GA.

Figure 5 shows the accuracy changes during the time. The horizontal axis represents weeks in the train 2+test dataset, and the vertical axis represents accuracies in percent. Also, Figure 6 shows the prediction accuracy of models for three random days. Based on Figures 5 and 6, there is no most accurate model for all the times due to the temporal variation of accuracies. This finding emphasizes using ensemble methods to provide unique predictions with the highest possible accuracy.

At the next step, OL is calibrated by using predictions of contributing models for train dataset 2. The performance of ensemble methods, including voting algorithms and OL and single models, is evaluated for the test dataset. The results are presented in Table 6.

Table 6 shows that the OL model outperforms its input with 81.03% prediction accuracy. For every period, the OL model can detect the more accurate single model and put more value on that model's prediction. After OL, the worst state voting algorithm provides 78.35% of accuracy. It means that single models have more tendencies to predict light state and sometimes miss heavy state.

It is essential to evaluate models for predicting each traffic state. For this purpose, among diverse evaluation metrics, F-measure (F_1) and balanced accuracy (because of an unbalanced distribution of observations) are calculated by using confusion matrices (Table 7) and equations 11 and 12 (see Akosa, 2017, Labatut and Cherifi, 2012, and Tharwat, 2018, about calculating recall, precision, and specificity).

$$\text{Balanced accuracy} = \frac{(\text{Recall} + \text{Specificity})}{2}, \quad (11)$$

$$F_1 = \frac{2(\text{Precision} * \text{Recall})}{\text{Precision} + \text{Recall}}. \quad (12)$$

These metrics are calculated for single and ensemble methods and presented in Table 8.

Based on Table 8, the OL predicts all states more accurately than voting and single machine learning methods. This model predicts states A, B, and C with balanced accuracy equal to 89, 73.4, and 58.5 percent, respectively. Also, OL's F_1 s equal 0.813 for state A, 0.872 for state B, and 0.292 for state C and all of them are the

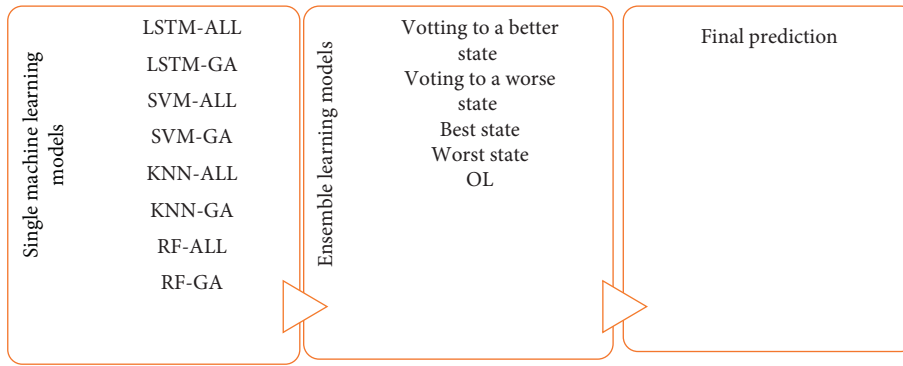


FIGURE 4: Proposed ensemble learning process.

TABLE 4: Optimum values for parameters of models.

Parameter	Model	The optimum value by using all features	The optimum value by using selected features by GA
K	KNN	26	37
NT	RF	162	109
NV	RF	9	8
C	SVM	5	3

TABLE 5: Accuracy of predictions of single models.

Model	Train 1-all features	Train 2+test-all features	Train 1-GA features	Train 2+test-GA features
LSTM	76.65	70.77	75.93	70.89
SVM	82.09	76.25	80.60	75.48
KNN	80.10	74.94	79.15	75.41
RF	97.94	78.31	98.09	78.66

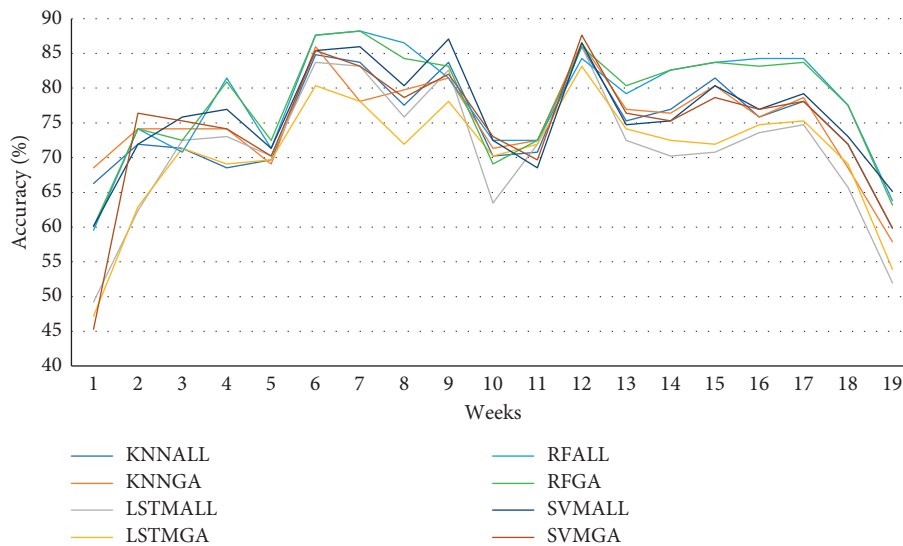


FIGURE 5: Accuracy changes of single models during the time.

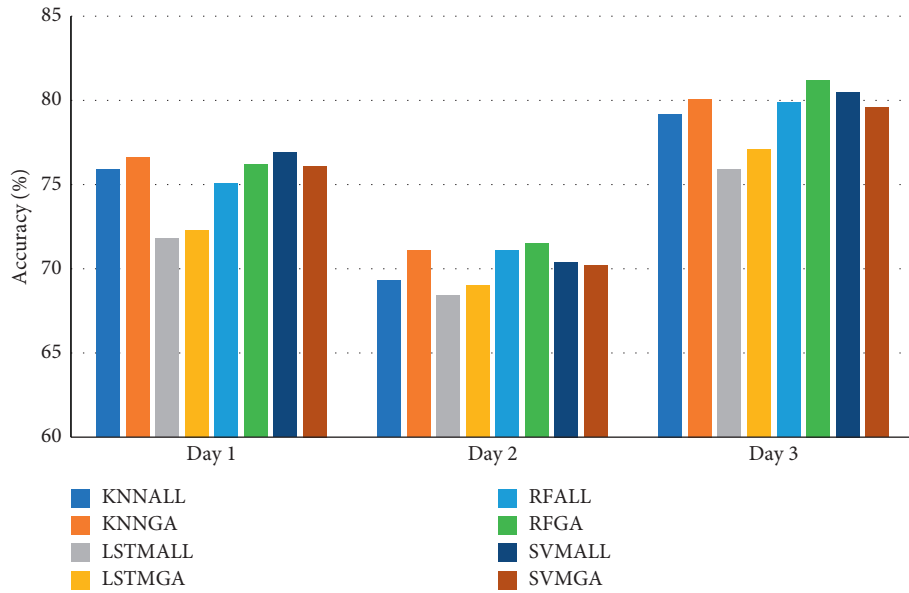


FIGURE 6: The prediction accuracy of models for three random days.

TABLE 6: Accuracy of predictions of ensemble and single models for test datasets.

Model	Test-all features	Test-GA features
LSTM	68.92	70.71
SVM	75.37	74.38
KNN	74.18	73.58
RF	76.96	76.86
Voting to a better state		75.17
Voting to a worse state		75.67
Best state		67.63
Worst state		78.35
OL		81.03

TABLE 7: Confusion matrices of predictions for the test dataset.

True prediction	A	B	C
Model		LSTM-all	
A	151	151	3
B	19	542	140
C	0	0	1
Model		SVM-all	
A	136	81	0
B	34	610	131
C	0	2	13
Model		KNN-all	
A	144	97	1
B	26	585	125
C	0	11	18
Model		RF-all	
A	150	78	0
B	20	614	133
C	0	1	11

TABLE 7: Continued.

True prediction	A	B	C
Model		Voting to a better state	
A	149	86	1
B	21	607	142
C	0	0	1
Model		Best state	
A	162	175	3
B	8	518	140
C	0	0	1
Model		OL	
A	141	36	0
B	29	649	118
C	0	8	26
Model		LSTM-GA	
A	137	119	1
B	33	574	142
C	0	0	1
Model		SVM-GA	
A	143	88	1
B	27	605	142
C	0	0	1
Model		KNN-GA	
A	144	90	1
B	26	594	140
C	0	9	3
Model		RF-GA	
A	151	76	0
B	19	613	134
C	0	4	10
Model		Voting to a worse state	
A	142	76	1
B	28	617	140
C	0	0	3
Model		Worst state	
A	116	26	0
B	54	645	116
C	0	22	28

TABLE 8: Evaluation metrics for predicting each traffic state.

Model	State	Precision	Recall	Specificity	Balanced accuracy	F1
LSTM-all	A	0.495	0.888	0.779	0.834	0.636
	B	0.773	0.782	0.489	0.635	0.778
	C	1	0.007	1	0.503	0.014
LSTM-GA	A	0.533	0.806	0.827	0.817	0.642
	B	0.766	0.828	0.441	0.635	0.796
	C	1	0.007	1	0.503	0.014
SVM-all	A	0.627	0.8	0.885	0.842	0.703
	B	0.787	0.88	0.475	0.677	0.831
	C	0.867	0.09	0.997	0.544	0.164

TABLE 8: Continued.

Model	State	Precision	Recall	Specificity	Balanced accuracy	F1
SVM-GA	A	0.616	0.841	0.872	0.857	0.711
	B	0.782	0.873	0.46	0.667	0.825
	C	1	0.007	1	0.503	0.014
KNN-all	A	0.616	0.841	0.872	0.857	0.711
	B	0.782	0.873	0.46	0.667	0.825
	C	1	0.007	1	0.503	0.014
KNN-GA	A	0.613	0.847	0.868	0.857	0.711
	B	0.782	0.857	0.47	0.663	0.818
	C	0.25	0.021	0.988	0.504	0.038
RF-all	A	0.658	0.882	0.889	0.886	0.754
	B	0.801	0.886	0.513	0.699	0.841
	C	0.917	0.076	0.999	0.538	0.141
RF-GA	A	0.665	0.888	0.891	0.889	0.761
	B	0.8	0.885	0.513	0.699	0.84
	C	0.714	0.069	0.995	0.532	0.127
Voting to a better state	A	0.631	0.876	0.875	0.876	0.734
	B	0.788	0.876	0.479	0.678	0.83
	C	1	0.007	1	0.503	0.014
Voting to a worse state	A	0.648	0.835	0.89	0.862	0.73
	B	0.786	0.89	0.463	0.677	0.835
	C	1	0.021	1	0.51	0.041
Best state	A	0.476	0.953	0.745	0.849	0.635
	B	0.778	0.747	0.524	0.636	0.762
	C	1	0.007	1	0.503	0.014
Worst state	A	0.817	0.682	0.963	0.823	0.744
	B	0.791	0.931	0.459	0.695	0.855
	C	0.56	0.164	0.972	0.583	0.289
OL	A	0.797	0.829	0.949	0.89	0.813
	B	0.815	0.937	0.532	0.734	0.872
	C	0.765	0.181	0.99	0.585	0.292

highest achieved F_1 for each traffic state. Table 7 shows that using the OL increases the balanced accuracy of predicting traffic states A, B, and C, about 0.1%, 3.5%, and 4.1%, respectively, compared to the highest accuracy achieved by the models it puts together.

OL coefficients help to find the importance of predicting each traffic state by each model. Table 9 shows the results of the OL model. Predictions of models converted to binary (dummy) variables to be used in the OL model. Values in parentheses show t-state.

Negative coefficients decrease s_{qi}^* . It means that they increase the probability of states A and B compared to state C, as a base traffic state. For example, predicting traffic state A by KNN-all decreases s_{qi}^* by 1.57 units. This decrease leads to an increase in the occurrence probability of state A compared to the traffic state C. T-state under 1.56 shows statistically insignificant variables at the 90% level of significance. For example, LSTM predictions are statistically insignificant in predicting traffic states.

Theoretically, the proposed ensemble learning process has no prediction time horizon limitation, but the accuracy of prediction models decreases as time passes. The prediction horizon is different in previous studies. Some previous studies suggest 6 months to have accurate prediction [38], but it completely depends on the employed model and data. Also, Figure 5 shows that the prediction accuracy of single models decreases dramatically after 17 weeks. Finally, 6-month prediction time horizon seems to be suitable based on the literature and Figure 5.

Finally, predicted traffic states could be informed to travelers and transportation operators via advanced traveler information systems. Travelers will have more insights for choosing their departure times and routes to destinations. Also, using these predictions, system operators are better prepared to deal with unsuitable traffic conditions, and they may implement policies such as access restrictions or increasing the number of route lanes on the schedule to avoid high congestion.

TABLE 9: Results of the OL model.

Traffic states	KNN-all	KNN-GA	LSTM-all	LSTM-GA	RF-all	RF-GA	SVM-all	SVM-GA
State A	-1.57 (-3.46)	-2.88 (-5.00)	-0.18 (-0.46)	-0.64 (-1.15)	-2.53 (-30.1)	-3.82 (-4.16)	-3.17 (-7.91)	2.36 (4.09)
State B	-1.16 (-2.94)	-1.79 (-3.41)	-0.13 (-0.44)	-0.64 (-1.27)	-1.37 (-1.75)	-2.66 (-3.05)	-2.12 (-7.00)	1.95 (3.99)
State C (base)	—	—	—	—	—	—	—	—
Other parameters								
Intercept	8.47 (10.93)	Threshold 1	-2.01 (-5.02)	Threshold 2	2.52 (6.56)		Log-likelihood	-1352.3

5. Conclusion

Short-term traffic state prediction is a tool in the advanced traveler information system that aims to bring a more sustainable and more reliable transportation network. By predicting the near future of transportation network performance, travelers and system operators are more ready to face congested traffic or avoid getting stuck in traffic. This paper predicts the nominal practical traffic state that is more understandable for travelers. Many features are extracted in the preprocessing step related to solar and lunar calendars, weather conditions, and blockages. Feature selection is made by GA systematically. Then machine learning models consisting of LSTM, KNN, SVM, and RF models are trained using the GA selected features and all features. Ensemble methods, including four voting methods and the OL model, use all predictions and predict one final prediction to inform the road passengers and transportation agencies. The final results show that OL obtains the highest accuracy among machine learning and ensemble learning algorithms, which equals 81.03%. The highest accuracy of single machine learning methods is 76.96%, achieved by RF. The feature selection by GA maintains the accuracy of predictions and increases the accuracy of some models. Regarding F_1 and balanced accuracy, traffic states A, B, and C are predicted more accurately by the OL model in the ensemble learning process. This model provides interpretable coefficients, which can be used to show the importance of models prediction.

For future studies, using and comparing other ensemble learning methods such as gradient boosting decision trees (GBDT) and neural network bagging ensemble hybrid model is proposed.

Data Availability

The traffic time-series data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this manuscript.

Acknowledgments

The authors would like to acknowledge the Road Maintenance and Transportation Organization (RMTO) for supporting this research by providing suburban traffic data of Iran.

References

- [1] S. Majumdar, M. M. Subhani, B. Roullier, A. Anjum, and R. Zhu, "Congestion prediction for smart sustainable cities using IoT and machine learning approaches," *Sustainable Cities and Society*, vol. 64, Article ID 102500, 2021.
- [2] Y. E. Mokaddem and F. Jawab, "Researches and applications of intelligent transportation systems in urban area: systematic literature review," *ARPJ J. Eng. Appl. Sci.*, vol. 14, no. 3, pp. 639–652, 2019.
- [3] Y. Tian and L. Pan, "Predicting short-term traffic flow by long short-term memory recurrent neural network," in *Proceedings of the 2015 IEEE International Conference on Smart city/SocialCom/SustainCom (SmartCity)*, December 2015.
- [4] S. Shahriari, M. Ghasri, S. A. Sisson, and T. Rashidi, "Ensemble of ARIMA: combining parametric and bootstrapping technique for traffic flow prediction," *Transportmetrica: Transportation Science*, vol. 16, no. 6, pp. 1–22, 2020.
- [5] L. Zheng, H. Huang, C. Zhu, and K. Zhang, "A tensor-based K-nearest neighbors method for traffic speed prediction under data missing," *Transportation Business: Transport Dynamics*, vol. 8, no. 1, pp. 182–199, 2020.
- [6] M. Zahid, Y. Chen, A. Jamal, and M. Q. Memon, "Short term traffic state prediction via hyperparameter optimization based classifiers," *Sensors*, vol. 20, no. 3, 685 pages, 2020.
- [7] T. P. Carvalho, F. A. A. M. N. Soares, R. Vita, R. d. P. Francisco, J. P. Basto, and S. G. S. Alcalá, "A systematic literature review of machine learning methods applied to predictive maintenance," *Computers & Industrial Engineering*, vol. 137, Article ID 106024, 2019.
- [8] G. Chang, Y. Zhang, D. Yao, and Y. Yue, "A summary of short-term traffic flow forecasting methods," in *Proceedings of the ICCTP 2011: Towards Sustainable Transportation Systems*, pp. 1696–1707, Nanjing, China, July 2011.
- [9] P. M. Maçaira, A. M. T. Thome, F. L. C. Oliveira, and A. L. C. Ferrer, "Time series analysis with explanatory variables: a systematic literature review," *Environmental Modelling & Software*, vol. 107, pp. 199–209, 2018.
- [10] J. Fan and Q. Yao, "Nonlinear Time Series," *Nonparametric and Parametric Methods*, Springer Science & Business Media, New York, NY, USA, 2008.
- [11] E. Bowitz, "Disability benefits, replacement ratios and the labour market. A time series approach," *Applied Economics*, vol. 29, no. 7, pp. 913–923, 1997.
- [12] B. Kedem and K. Fokianos, *Regression models for time series analysis*, Vol. 488, John Wiley & Sons, Hoboken, New Jersey, USA, 2005.
- [13] T. Alghamdi, K. Elgazzar, M. Bayoumi, T. Sharaf, and S. Shah, "Forecasting traffic congestion using ARIMA modeling," in *Proceedings of the 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, June 2019.
- [14] C. Ding and J. Duan, Y. Zhang, X. Wu, and G. Yu, "Using an ARIMA-GARCH modeling approach to improve subway short-term ridership forecasting accounting for dynamic volatility," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 4, pp. 1054–1064, 2017.
- [15] T. Wuest, D. Weimer, C. Irgens, and K.-D. Thoben, "Machine learning in manufacturing: advantages, challenges, and applications," *Production & Manufacturing Research*, vol. 4, no. 1, pp. 23–45, 2016.
- [16] E. Alpaydin, *Introduction to Machine Learning*, MIT press, Cambridge, MA, USA, 2020.
- [17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [18] W. S. Noble, "What is a support vector machine?" *Nature Biotechnology*, vol. 24, no. 12, pp. 1565–1567, 2006.
- [19] L. Peterson, "K-nearest neighbor," *Scholarpedia*, vol. 4, no. 2, p. 1883, 2009.
- [20] A. Liaw and M. Wiener, "Classification and regression by random forest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.

- [21] B. Du, H. Peng, S. Wang et al., "Deep irregular convolutional residual lstm for urban traffic passenger flows prediction," *IEEE Transactions on Intelligent Transportation Systems*, 2019.
- [22] D. Kang, Y. Lv, and Y.-y. Chen, "Short-term traffic flow prediction with LSTM recurrent neural network," in *Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, October 2017.
- [23] P. Cai, Y. Wang, G. Lu, P. Chen, C. Ding, and J. Sun, "A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting," *Transportation Research Part C: Emerging Technologies*, vol. 62, pp. 21–34, 2016.
- [24] Y. Liu and H. Wu, "Prediction of road traffic congestion based on random forest," in *Proceedings of the 2017 10th International Symposium on Computational Intelligence and Design (ISCID)*, December 2017.
- [25] X. Li, Y. Zhang, M. Du, and J. Yang, "The forecasting of passenger demand under hybrid ridesharing service modes: a combined model based on WT-FCBF-LSTM," *Sustainable Cities and Society*, vol. 62, p. 102419, 2020.
- [26] F. Moretti, S. Pizzuti, S. Panzieri, and M. Annunziato, "Urban traffic flow forecasting through statistical and neural network bagging ensemble hybrid modeling," *Neurocomputing*, vol. 167, pp. 3–7, 2015.
- [27] S. Yang, J. Wu, Y. Du, Y. He, and X. Chen, "Ensemble learning for short-term traffic prediction based on gradient boosting machine," *Journal of Sensors*, vol. 2017, Article ID 7074143, 2017.
- [28] N. D. Bokde, Z. M. Yaseen, and G. B. Andersen, "ForecastTB—an R package as a test-bench for time series forecasting-application of wind speed and solar radiation modeling," *Energies*, vol. 13, no. 10, 2578 pages, 2020.
- [29] M. Kuhn and K. Johnson, *Applied predictive modeling*, Vol. 26, Springer, , New York, NY. USA, 2013.
- [30] D. Mandic and J. Chambers, *Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability*, Wiley, Hoboken, New Jersey, United States, 2001.
- [31] A. Sherstinsky, "Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network," *Physica D: Nonlinear Phenomena*, vol. 404, Article ID 132306, 2020.
- [32] Y. Tian, K. Zhang, J. Li, X. Lin, and B. Yang, "LSTM-based traffic flow prediction with missing data," *Neurocomputing*, vol. 318, pp. 297–305, 2018.
- [33] S. Mokhtarimousavi, J. C. Anderson, A. Azizinamini, and M. Hadi, "Improved support vector machine models for work zone crash injury severity prediction and analysis," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2673, no. 11, pp. 680–692, 2019.
- [34] S. Han, C. Qubo, and H. Meng, "Parameter selection in SVM with RBF kernel function," in *World Automation CongressIEEE*, 2012.
- [35] G. Biau and E. Scornet, "A random forest guided tour," *Test*, vol. 25, no. 2, pp. 197–227, 2016.
- [36] M. Yin, D. Yao, J. Luo, X. Liu, and J. Ma, "Network backbone anomaly detection using double random forests based on non-extensive entropy feature extraction," in *Proceedings of the 2013 Ninth International Conference on Natural Computation (ICNC)*, July 2013.
- [37] S. Seyedabrishami, A. R. Izadi, H. S. Rayaprolu, and R. Moeckel, "Car Ownership: A Joint Model for Number of Cars and Fuel Types," *Transportation Research Procedia*, vol. 41, pp. 572–576, 2019.
- [38] A. Rasaizadi and S. E. Seyedabrishami, "Traffic state prediction by machine learning algorithms for short-term and mid-term prediction time horizons," *Amirkabir Journal of Civil Engineering*, 2021.