**Research Article**

**Real-Time Travel Time Prediction Based on Evolving Fuzzy Participatory Learning Model**

Yongyi Li,1 Ming Zhang,1 Yixing Ding,2 Zhenghua Zhou,3 and Lingyu Xu1

1Nanjing Tech University, School of Transportation Engineering, Puzhu South Road 30, Jiangbei New Area, Nanjing, China
2ZTE Corporation, 6 Huashen Avenue Road 6, Nanjing, China
3Nanjing Tech University, School of Transportation Engineering, Zhongshan North Road 200, Nanjing, China

Correspondence should be addressed to Yongyi Li; 49232712@qq.com

Received 4 November 2021; Revised 1 April 2022; Accepted 21 April 2022; Published 12 May 2022

Academic Editor: Xianyu Wu

Copyright © 2022 Yongyi Li 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.

Urban expressways take on rapid and external transport in the city due to their fast, safe, and large capacity. Implementing intelligent and active traffic control can effectively improve the performance of urban traffic and mitigate the urban traffic congestion problem. Real-time traffic guidance is one critical way of intelligent active traffic control, and travel time is the most important input for real-time traffic guidance. We employed and improved a machine learning method called the evolving fuzzy participatory learning (ePL) model to predict the freeway travel time online in this paper. The ePL model has a promising non-linear mapping potential, which is well suited for the traffic prediction. We used generalized recursive least square (GRLS) to improve the estimation accuracy of the model’s parameters. This model is a fuzzy control model. Its output is the forecasting result which is also the fuzzy reasoning result. We tested this model by comparing it to other travel time prediction approaches, with the freeway data from the Caltrans Performance Measurement System. The results from the improved ePL model showed mean absolute error of 5.941 seconds, mean absolute percentage error of 1.316%, and root mean square error of 10.923 s. The performances are better than those of the baseline models including ARIMA and BPN. This model can be used to predict the travel time in the field to be used for active traffic control and traffic guidance.

## 1. Introduction

The ways to predict travel time have undergone a significant change in past decades. From data acquisition perspective, most of the research studies try to estimate link travel time using trajectory data from probe vehicle data [1]. The main advantage of this method is that the average travel time can be calculated by summing the estimated average link travel time along the route. Although the research on probe vehicle data (e.g., INRIX, DiDi, Taxi GPS data, etc.) become popular, we cannot ignore that there are areas which do not have access to the probe vehicle data due to market penetration and policy reasons. We believe that the stationary detector data will not fully be replaced by floating data in the near future. The stationary detectors will still be operated in the field even with the increasing probe vehicle technologies.

The data processing approaches have shifted from the traditional model-based approaches to the data-driven approaches because of the extensive applications of detectors and the flourish of big data technology. Some reviews have sensed these changes, e.g., Oh et al. [2, 3] focused on the testing of data-driven approaches: both parametric approaches (the linear regression and time series modelling approaches) and nonparametric approaches (artificial intelligent approaches). The model-based approaches are approaches based on models such as macroscopic model, mesoscopic model, cellular automata model, and so on. They are the main model-based models used in the literature. However, the model-based approaches are not commonly used in travel time prediction in practice because of their high price. Data-driven methods [4] have attracted more and more research interest and achieved inspiring results with the improvement of computing capability and the growth of
traffic data quantity. Data-driven methods usually employ historical travel time [5] and other related variables, e.g., speed, volume, occupancy [6], time of day, day of week, and so on [7]. Among the models and algorithms applied in data-driven methods, the ARIMA model [8] used the historical series of travel time to fit a time series model and then predicted the future travel time one by one. The linear model was used to predict the travel time of one trip departing at current time by combining the latest calculated travel time of the trip and the historical mean travel time of the same trip departing at the same time [9, 10]. Generally, the data-driven approaches, e.g., (extended) Kalman filter (KF) [11] and ARIMA prediction models [12], pay more attention to the results than the explanation of traffic intrinsic quality. In addition to the above model algorithms, researchers are increasingly favoring the use of artificial neural networks to predict travel times. Mane and Pulugurtha considered the factors such as time of day, working days and nonworking days, and historical travel time [13]. They developed an artificial neural network with a double-layer feedforward network to predict expressway travel time. Duan et al. found that there existed communication delays in obtaining the expressway travel time statistics [14]. By training a deep learning model based on LSTM, the model considered the correlation within the travel time series. Most of the studies were based on the card swiping data and detector data of expressway toll stations, while there are few studies on urban roads. However, deep learning approaches require large amount of training data which may not be available in many scenarios.

Among travel time prediction methods developed in the aforementioned literature, we found that (1) most methods have strong nonlinear mapping ability, and (2) they are clustered firstly and then output the prediction results. However, the model structure is relatively complex, difficult to interpret, and requires a large amount of calculation. This work developed a practical and data-driven travel prediction model based on the ePL model, an emerging machine learning method. We followed the idea of “first-clustering-then-forecasting.” We followed the idea of “first-clustering-then-forecasting.” We followed the following steps:

(1) Using the unsupervised clustering approach based on participatory learning to classify a sequence of travel time series.
(2) Calculating the optimal value of predicted travel time.
(3) Evaluating the proposed method and comparing the performance of the proposed method with other methods.

This research is expected to provide an application of artificial intelligence in predicting the travel time under dynamic response mechanism.

2. Problem Statements

2.1. Link Travel Time. A unit segment of the route is shown in Figure 1. In Figure 2, \( v_{i-1} \) and \( v_i \) represent the time mean speed (TMS) (mile/hour). \( q_{i-1} \) and \( q_i \) represent the upstream and downstream flow (veh/hour/lane), respectively, and \( r_i \) and \( s_i \) represent the traffic flow in on-ramp and off-ramp, respectively. The positive integer \( i \) is the section number.

There are two categories of travel time: the dynamic travel time (DTT) and instantaneous traveling time (ITT). The ITT is an estimation of DTT while the DTT is seen as the real travel time.

The DTT on the \( i \)th link can be defined as

\[
\text{DTT}^i = \frac{L_i}{v(x, t_c)} \quad (1)
\]

where \( L_i \) is the length of the \( i \)th section (mile) and \( v(x, t) \) is a continuous function of the two variables: the location \( x \) and the timestamp \( t \). \( t_c \) is a certain value which can make the equation a true statement (Figure 3). It can be inferred that \( t_c \) is hard to find. So, the most used measurement of travel time is ITT. The expression of ITT is

\[
\text{ITT}^i = \frac{L_i}{\overline{v}} \quad (2)
\]

where \( \overline{v} \) is the mean speed of the location speed used in the boundaries in most papers. As shown in Figure 4, \( \overline{v} = [v(x_1, t_1) + v(x_2, t_2)]/2 \) where \( v(x_1, t_1) \) and \( v(x_2, t_2) \) are the location speeds.

Based on the aforementioned definition, the travel time passing through the \( i \)th section can be calculated as

\[
\tau^k = \frac{L_i}{\overline{v}} \quad (3)
\]
where $\tau_k^t$ is the instantaneous travel time at the $k$th time step. Most traffic practitioners preferred to compute TMS or their harmonic mean value instead of space-mean speed (SMS), mainly when using double loop detectors, and nowadays this is the common practice in AITSs. Accurately, this paper transforms the TMS to SMS by the simple method:

$$\overline{v}_{t} = \overline{v}_{t-1} - \frac{\sigma^2_v(t)}{\overline{v}_{t-1}}$$  \hspace{1cm} (4)

where $\sigma^2_v(t)$ is the speed variance $\overline{v}_{t}^2$ (mph2), $\overline{v}_{t} = (v_{i-1} + v_{i})/2$ is the current TMS. This relationship is applicable when the traffic flow is continuous which is often the state of freeway. For simplifying calculation and avoiding the data saturation, the successive data adopted to calculate $\sigma^2_v(t)$ are restricted to the last 3 ones before the current step and its initialized value is 10 empirically.

### 3. Proposed Method

The evolving fuzzy participatory learning is selected to predict the travel time. This method includes two steps: dynamic clustering and aggregation. We first classified the instantaneous travel time series calculated from the detectors’ data. Then, we aggregated the prediction result by the weighted average value of degree of membership.

#### 3.1. Evolving Fuzzy Participatory Learning

There are many machine learning algorithms such as linear regression, decision tree algorithm, logistic regression, and artificial neural networks. Among these, the artificial neural networks and fuzzy algorithms are considered as the most potential artificial intelligence (AI) algorithms. Because of the halo effect of neural network, a lot of other machine learning algorithms are overshadowed. However, this does not mean that other algorithms are useless. In the 90s, Yager [15] proposed a machine learning algorithm called participatory learning (PL). Some scholars [16] then found that this algorithm can be applied to clustering analysis well. Later, some scholars found that the T-S model modified by participatory learning can achieve fine results [17–20]. After years of development [21–27], the model named evolving fuzzy participatory learning (ePL) model appeared, which is the main trend of development of participatory learning. Although the ePL model has been implemented in many fields, no application has been found in travel time prediction.

The participatory learning sketch map which was proposed firstly by Yager and refined in this paper is shown in Figure 2. In Figure 2, the "arousal mechanism" and "learning process" compose the cluster structure. The "arousal mechanism" is independent to the cluster structure. Through the observation vector $x_k$, the degree of compatibility $p_k r \in [0, 1]$ is calculated. According to the degree of compatibility, information will be transmitted to "learning process" in feedback. The "arousal mechanism" uses the arousal index $a_k r \in [0, 1]$ to inspect whether the system is in good condition.

Since the Takagi–Sugeno model (T-S) has been widely used in various disciplines, lots of variations have been developed including the evolving T-S (eTS) models. Angelov and Buswell [28] proposed the evolving rule-base evolution by recursive adaptation of rule structure and parameters and gave the form of eTS. In eTS, the rules in the rule base are optimized with the data process.

The ePL, an improved version of eTS, uses the participatory learning idea to determine whether a cluster needs to be created, modified, or removed. The ePL model consists of two main processes: the dynamic cluster process and the aggregation process. The dynamic cluster process is the main process by which the input vector $x_k$ can be classified. This process is related to a kind of clustering method that uses the participatory learning. The aggregation process fuses the input vector and calculates the prediction result. It includes the parameter identification (recursive least square, RLS) and weighted average value calculation. The general flowchart is shown in Figure 4.

#### 3.2. Dynamic Cluster

This paper uses the participatory learning clustering method to cluster the input traffic data. The advantage is to improve the performance of clustering. Before the model starts, a threshold value $r \in [0, 1]$ is given, and arousal index $a_k r$ will be compared with $r$ in $k$th time
step and ith cluster. If \( ak r \) is bigger than \( r \), then a new cluster whose center is \( x^r \) is created. Otherwise, the system will search the rth cluster that is the most compatible with \( x^r \) and updates the cluster’s center \( \chi^k+1 r \) by the following equation:

\[
x^k+1 r = x^r + G^k_r (x^k - x^r),
\]

(5)

\[
G^k_r = a(\rho^r_k)^{1 - \alpha^r},
\]

(6)

which is the effective leaning rate, where \( a \in [0, 1] \) is the rate of learning and \( ak r \in [0, 1] \) is the value of the rth compatible degree, and its expression is

\[
\rho^r_k = 1 - \frac{||x^k - x^r||}{p},
\]

(7)

\[
r = \arg \max_j \{\rho^r_j\},
\]

(8)

where \( || \cdot || \) is the Frobenius norm, \( p \) is the dimension of input data \( x^r \), and the value \( r \) is equal to \( j \) that is the subscript in \( pk j \) which is the biggest value among the set \{\( pk 1, pk 2, \ldots \)\}. Similar to (5), the participatory leaning theory is employed in the update expression of the arousal indexes:

\[
a^{k+1} r = a^r + \beta (1 - \rho^r_k) - a^k,
\]

(9)

where \( ak r \in [0, 1] \) is the arousal index in Figure 2 (“arousal index \( ak r \)”) and \( r \) here belongs to the positive integer set \{1, 2, \ldots, \( c \)\}, where the \( c \) is the number of rules in the rule base (i.e., the number of clusters). The greater the arousal index is, the higher the probability that the “arousal mechanism” is activated. Coefficient \( \beta \in [0, 1] \) controls the arousal index’s changing speed. The bigger the \( \beta \) is, the more easier the system senses the compatibility variations. The effective learning rate \( Gk i \) is the intermediate variable that controls the learning process directly. If \( ak r = 0 \), \( Gk i = \alpha^k pk i \), the PL process of the system is not aroused. If the arousal index \( ak r \) is becoming larger, the \( Gk \) is becoming lager too. Arousal index \( ak r \) can be seen as a quantized index of the reliability in the belief, i.e., the cluster structure. In the “arousal mechanism,” the compatibility between the new data and the current belief is measured and is sent (by the parameter \( Gk i \)) to the “learning process.” The “learning process” receives the \( Gk i \) and corrects the old cluster center \( \chi^k i \).

To enhance the effectiveness of the clustering results, an integral process should be created to mix the similar clusters. A method to measure the similarity between two clusters is as follows:

\[
\rho^k_{\lambda i} = 1 - \frac{1}{p} \sum_{j=1}^{p} |x^k j - x^\lambda j|.
\]

(10)

According to (10), the mechanism uses this formula to compute the similarity of cluster centers: \( \rho^k \chi^1, \rho^k \chi^2, \ldots, \rho^k \chi^p \). And, if \( \rho^k_{\lambda i} \geq \lambda \), then additional cluster (the rth cluster) need to be removed where the mixing method is implemented.

### 3.3. Aggregation

The aggregation process is a fuzzy reasoning result essentially (Figure 4). Since the ePL is a form of the T-S model, the consequent parameter estimation is necessary. By the consequent parameters, the model can give the fuzzy reasoning result.

Assuming that there are \( c^k \) rules in the ePL model, let \( yr \) be the rth subsystem output travel time, and the model and its first-order variations are as follows:

\[
R_r : \text{IF } x^k \text{is} \Gamma_r \text{THEN } y_r = y_{0r} + \sum_{j=1}^{p} \mu_j x_j,
\]

(11)

where \( R_r \) is the rth fuzzy rule, \( x^k \) represent the input data, \( p \) is the dimension of \( x^k \), \( yr \) is the output of the rth rule, \( \Gamma_r \) is the vector of antecedent fuzzy rules, and \( \mu_j \) are parameters of the consequent. The RLS is often used to estimate the parameter \( \mu_j \) which is introduced in next section. Output \( y \) is regarded as the weighted average value of \( y_j \) (corresponding weight \( \mu_j \)):

\[
y = \frac{\sum_{j=0}^{c^k} \mu_j y_j}{\sum_{j=0}^{c^k} \mu_j},
\]

(12)

where \( y_r \) is the rth rule’s output value and \( \mu_j \) is the corresponding weight value given by the rule’s firing degree by Gaussian membership function. \( \kappa \) denotes a positive constant which defines the affected region of the rth subsystem. If \( \kappa \) is bigger, the affected region is smaller. \( x^k \) is the kth input data (vector). \( \chi_r \) is the rth valuation. Since a rule in rule base represents a cluster, \( \chi_r \) is also the rth cluster’s center.

### 3.4. Parameter Estimation

Prediction methods for time series data based on the recursive least square (RLS) method performed better [29], especially for a time series in which many missing data are continuously present. The general recursive least square algorithm (GRLS) is frequently used to eliminate the heteroscedasticity and is demonstrated to be much more accurate for predicting travel time than some ordinary recursive least square algorithms (e.g., Kalman filter algorithm). This paper used the GRLS to improve the estimation accuracy. The parameters estimated are the consequent parameters in rule base (\( y_r \)) in (11). In the dynamic cluster process, each input vector is classified to one cluster existing in the cluster structure or not existing. After several iterations, a series of data may all be classified to a certain cluster. Parameter identification is to use these data to estimate the cluster’s corresponding rule’s consequent parameters. As the data saturation is the defect of the least square method, recent 10 pairs of data are used to estimate the parameters. This paper uses the generalized least square method (GLS) which can overcome biased estimation in RLS. The output of the ePL model in (11) can be written as follows:

\[
y^k_r = x^k_r y^k + \varepsilon^k,
\]

(13)
where \( y_k = [y_i], i = 1, 2, \ldots, c^k \), the vector \( y^k = [yk 0 y_k 1 \ldots y_k p] \) is the consequent parameter, and \( x_k e = [1 \ x_k^{-2} x_k^{-1} x_k]^T \) in which \( x_k \) is the input vector. \( c^k \) denotes the modelling error at \( k \). In GLS, the hypothesis is that the colored noises \( \xi \) have linear relationship with the white noise \( \varepsilon \):

\[
\xi = \Omega f + \varepsilon \tag{14}
\]

where \( f = [f_1 f_2 \ldots f_m]^T \) is the parameter vector that needs to be estimated and \( \Omega \) is a matrix. Because \( \xi \) is not measured directly, the error \( \varepsilon \) is used instead of \( \xi \) when identifying parameter vector \( f \).

The way to estimate \( \xi \) by \( \varepsilon \) is the typical RLS. First, set \( m = 2 \) in \( f \) and the filter signal:

\[
y^k e = y^k + f_1 y^{k-1} + f_2 y^{k-2}, \tag{15}
\]

and the filter signal of input vector is

\[
x^k e = x^k + f_1 x^{k-1} + f_2 x^{k-2}. \tag{16}
\]

As expressed in (14), the color error can be estimated by the local RLS and recurrence formulas:

\[
p^{k+1} E = P E \left[ I - \frac{(\varepsilon^k)^T \varepsilon^k P E}{1 + \varepsilon^k P E \varepsilon^k} \right], \tag{17}
\]

\[
f^{k+1}_E = f^{k+1}_E + P E (\varphi^{k+1}_E - \varepsilon^k - \bar{\varphi}^k P E \varepsilon^k), \tag{18}
\]

where \( I \) is the identity matrix, \( f_k E = [f_1 f_2 0]^T \) is the estimated vector of \( f \) at \( k \), \( \varphi^k E = [\varepsilon^k c^k \cdots c^{k-1}] \) is the vector, and \( P k + 1 E \) is the covariance matrix. The estimation of \( y^k \) can be expressed as follows:

\[
y^{k+1} = y^k + p^{k+1}(x^k e)^T (x^k e - x^{k+1} y), \tag{19}
\]

where dispersion \( p^{k+1} \) is

\[
p^{k+1} = p^k \left[ I - \frac{(x^{k+1}_e)^T x^{k+1} p^k}{1 + (x^{k+1}_e)^T P^k (x^{k+1}_e)^T} \right], \tag{20}
\]

and the residual error \( \varepsilon^k \) is

\[
\varepsilon^k = y^k - y^k e. \tag{21}
\]

Equations (16)–(20) are used to update the consequent parameters in rule base (11). At each step, the GLS updates one rule’s parameters and other rules’ consequent parameters remain unchanged. The dispersion matrices are updated independently. The initial value of dispersion matrix \( P^0 \) can be set big and \( y^0 \) can be set small.

### 4. Prediction

The authors tried to use the ePL model to predict the route travel time. The vehicle detector station (VDS) data in this case are downloaded from the PeMS (Caltrans Performance Measurement System). The 7.45 mile route of freeway 180-W Milepost 45.88 mile to Milepost 53.33 mile is located in California, America. This path connects Vacaville and Fairfield. There are two-way eight lanes and no traffic signal along the way, and design speed limit is 70 mph. 12 x 4 = 48 loop detectors and six on/off-ramp loop detectors are equipped under the main road and ramps, respectively. The raw data in PeMS are aggregated by 30 s recurrently. Both the sketch map of the path and the prediction diagram are drawn in Figure 5.

As is shown in Figure 5, the ePL predictors compute the link travel time synchronously. It has been assumed that during the time \([k-1, k] \), the total traffic flow remains the same and the speed is uniform. The \( \phi k i \) is postulated to be independent of time \( k \) but having a strong nonlinear relationship with previous series. The influence window data length is denoted as \( w \). To reduce the effect of data saturation, we assume \( w = 3 \) in this experiment. For the ePL predictor in Figure 5, the length of the time window consists of three steps, i.e., \( \phi t k^{k-2}, \phi t k^{k-1}, \phi t k \) are used to predict the \( \phi t k+1 \). This scale of the time window is inferred through the analysis of partial correlation coefficient. In a single ePL predictor, the steps of the proposed algorithm are listed in Algorithm 1.

In the Algorithm 1, it can be found that the state vector \( x^k \) is used to be the cluster object. We choose the state vector \( x^k \) because in the ePL process, the model will create several fuzzy sets which represent the various states of traffic flow. Since the \( 1 \times 3 \) dimension matrix can imply abundant information, the intrinsic evolution principle of traffic flow is excavated. In this sense, this prediction model is a data-driven model. Due to the stochastic nature of traffic flow, we simply suppose that travel time through the route \( r \) is \( y^k \):

\[
y^k = t_{i1} + t_{i2} + \ldots + t_{iN}, \tag{22}
\]

where \( t_{i} \) is the travel time through the \( i \)th section and \( N \) is the number of sections. The workflow chart is drawn in Figure 6.

### 5. Experiment

The raw data which are within 00:00–24:00 in the 14th August and 17th September in 2017 from PeMS are collected to test the performance of the proposed model. The authors select the two days in order to test the proposed approach under the different traffic flow situations such as free flow on Monday (14th August, 2017) and blocked flow on Sunday (17th September, 2017). These series of raw data are aggregated over the time interval of 30 s originally. For a more smooth prediction result, they are further compressed in a 5 min prediction horizon. Some missing data are filled by the spatiotemporal Lagrange interpolating polynomial algorithm. For a persuasive investigation, an ARIMA model (autoregressive integrated moving average model, the number of autoregressive terms \( p = 1 \), the number of nonseasonal differences \( d = 1 \), and the number of lagged forecast errors \( q = 1 \)) is used to be a comparison. The ARIMA model is chosen because it is one of the most popular online forecasting methods, and it can learn and predict online without prior knowledge or data training naturally. Besides, a propagation neural network (BPN) forecasting method
which represents the typical data-driven method is also adopted for comparison. Two hidden layers in which 10 nodes in the first layer and 20 nodes in the second layer are presented in the BPN algorithm. The code for the proposed algorithm is written in the Matlab, which is expedient for creating the ePL model structure.

To analyze the results quantificationally, the forecast accuracy measures, e.g., the mean absolute error (MAE), the mean absolute percentage error (MAPE), and the root mean square error (RMSE), are all employed to narrow our focus (Table 1).

Prediction results are shown in Figure 7, and the prediction efficiency of the ePL model is high. The travel time prediction accuracy is changing widely when the traffic flow is suddenly rising or dropping. The forecasting trends are consistent with the actual trends. After the beginning learning, the prediction accuracy became high gradually along with importation of the new data. Until the abrupt change appeared during the morning peak 8:00∼9:00 or the evening peak 16:00∼18:00, the prediction accuracy became a little lower. This characteristic can also be found in the cluster change lines in Figure 8. The number of rules changed with the abrupt increase/decrease of vehicles in observation series. The new rules are added when the new patterns of data occur. The old rules are mixed when the data are plain.

As shown in Table 1, the error index shows that the proposed ePL model is an alternative of the traditional ARIMA model. The availability of the proposed model is

![Figure 5: Scheme of the prediction model.](image-url)

**Algorithm 1:** The algorithm of predicting travel time by ePL.

1. Initialize the rule base;
2. Read new data $h^k = [v_{i,1}, v_{i}]$;
3. Calculate link travel time $t_{ik}$ by (3);
4. Make the input state vector $x^k = [tt_{k-2}, tt_{k-1}, tt_k]$ and $y^k = [tt_k]$;
5. Use the input state vector $x^k$ to compute every compatibility index $\rho_i$ and arousal index $a_i$ by (7) and (9);
6. According to [...] $\rho_i$, [...], update the rule center by (5) or create a new rule;
7. Update the consequent parameters [...] $c^r_i$, [...] using GRLS in (16)–(20);
8. Calculate each output $y_k^i$ in rule base (1);
9. Remove redundant rule according to the result of (10);
10. Output the forecasting route travel time $y_k^{i+1}$.

which represents the typical data-driven method is also adopted for comparison. Two hidden layers in which 10 nodes in the first layer and 20 nodes in the second layer are presented in the BPN algorithm. The code for the proposed algorithm is written in the Matlab, which is expedient for creating the ePL model structure.

To analyze the results quantificationally, the forecast accuracy measures, e.g., the mean absolute error (MAE), the mean absolute percentage error (MAPE), and the root mean square error (RMSE), are all employed to narrow our focus (Table 1).

Prediction results are shown in Figure 7, and the prediction efficiency of the ePL model is high. The travel time prediction accuracy is changing widely when the traffic flow is suddenly rising or dropping. The forecasting trends are consistent with the actual trends. After the beginning learning, the prediction accuracy became high gradually along with importation of the new data. Until the abrupt change appeared during the morning peak 8:00∼9:00 or the evening peak 16:00∼18:00, the prediction accuracy became a little lower. This characteristic can also be found in the cluster change lines in Figure 8. The number of rules changed with the abrupt increase/decrease of vehicles in observation series. The new rules are added when the new patterns of data occur. The old rules are mixed when the data are plain.

As shown in Table 1, the error index shows that the proposed ePL model is an alternative of the traditional ARIMA model. The availability of the proposed model is
further verified by a hypothesis test which uses t-test as the expression. The test statistic of the t-test is as follows:

\[
t = \frac{\overline{X}_1 - \overline{X}_2}{\sqrt{\left(\frac{(N_1 - 1)\sigma_1^2 + (N_2 - 1)\sigma_2^2}{N_1 + N_2 - 2}\right)\left(\frac{N_1 + N_2}{N_1 N_2}\right)}} \sim t(N_1 + N_2 - 2),
\]  

(23)
where $X_i$, $\sigma_i$, and $N_i$ represent averages, standard deviations, and sample sizes, respectively. To conduct the $t$-test better, it is necessary to determine whether the variance of these sets of data is equal. The result of F-test shows that the hypothesis of equal variance is acceptable. Based on this accepted hypothesis, use a 95% confidence interval ($t_{0.025} = 1.960$) to conduct a double tailed test. The observation values of the $t$-test statistic are shown in Table 2. The result states clearly that the ePL model outperforms all the other methods such as ARIMA and BPN.

### 6. Conclusion

The contribution of this paper is that a universal travel time forecasting model based on the idea of participatory learning is created. The authors improved the previous evolving participatory learning models [17, 22] on the consequent parameter estimation process. An investigation on this model was conducted by the real traffic data from the PeMS. The comparison results to the BPN and the ARIMA algorithm showed that the accuracy of the proposed model was high especially during the violent fluctuation period. The test showed that the proposed model can simulate the strong nonlinear characteristic of travel time well and can give a pretty good forecasting result.

The proposed model is a data-driven model. We call it because of its self-organization, unsupervised, and online learning characteristics while its basic ideas being induced by the macroscopic traffic model. Also, its highly modular structure makes it easily extensible and strongly explanatory. Benefiting from these characteristics, this fuzzy model can adapt to the abrupt changes in observation values rapidly and gives an optimal forecasting value in short and fast application [30].

### Data Availability

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

### Disclosure

Part of this article was presented in World Transport Convention 2017.

### Conflicts of Interest

The authors declare that they have no conflicts of interest.

### Acknowledgments

This research was supported by the National Natural Science Foundation Project of China (grant no. 51878349, 2018) and Transportation Science and Technology Project of Jiangsu Province (grant no. 2016X008). The authors are grateful for the support.

### References


