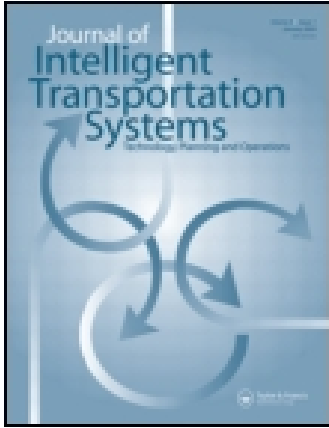


This article was downloaded by: [National Chiao Tung University 國立交通大學]

On: 24 December 2014, At: 17:21

Publisher: Taylor & Francis

Informa Ltd Registered in England and Wales Registered Number: 1072954 Registered office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK



Journal of Intelligent Transportation Systems: Technology, Planning, and Operations

Publication details, including instructions for authors and subscription information:

<http://www.tandfonline.com/loi/gits20>

A Novel Method to Predict Traffic Features Based on Rolling Self-Structured Traffic Patterns

Yu-Chiun Chiou^a, Lawrence W. Lan^a & Chun-Ming Tseng^a

^a Department of Transportation and Logistics Management, National Chiao Tung University, Taipei, Taiwan, Republic of China

Accepted author version posted online: 29 May 2013. Published online: 29 Jul 2014.



CrossMark

[Click for updates](#)

To cite this article: Yu-Chiun Chiou, Lawrence W. Lan & Chun-Ming Tseng (2014) A Novel Method to Predict Traffic Features Based on Rolling Self-Structured Traffic Patterns, Journal of Intelligent Transportation Systems: Technology, Planning, and Operations, 18:4, 352-366, DOI: [10.1080/15472450.2013.806764](https://doi.org/10.1080/15472450.2013.806764)

To link to this article: <http://dx.doi.org/10.1080/15472450.2013.806764>

PLEASE SCROLL DOWN FOR ARTICLE

Taylor & Francis makes every effort to ensure the accuracy of all the information (the "Content") contained in the publications on our platform. However, Taylor & Francis, our agents, and our licensors make no representations or warranties whatsoever as to the accuracy, completeness, or suitability for any purpose of the Content. Any opinions and views expressed in this publication are the opinions and views of the authors, and are not the views of or endorsed by Taylor & Francis. The accuracy of the Content should not be relied upon and should be independently verified with primary sources of information. Taylor and Francis shall not be liable for any losses, actions, claims, proceedings, demands, costs, expenses, damages, and other liabilities whatsoever or howsoever caused arising directly or indirectly in connection with, in relation to or arising out of the use of the Content.

This article may be used for research, teaching, and private study purposes. Any substantial or systematic reproduction, redistribution, reselling, loan, sub-licensing, systematic supply, or distribution in any form to anyone is expressly forbidden. Terms & Conditions of access and use can be found at <http://www.tandfonline.com/page/terms-and-conditions>

A Novel Method to Predict Traffic Features Based on Rolling Self-Structured Traffic Patterns

YU-CHIUN CHIOU, LAWRENCE W. LAN, and CHUN-MING TSENG

Department of Transportation and Logistics Management, National Chiao Tung University, Taipei, Taiwan, Republic of China

In this study, a novel method is proposed to predict the traffic features in a long freeway corridor with a number of time steps ahead. The proposed method, on the basis of rolling self-structured traffic patterns, utilizes the growing hierarchical self-organizing map model to partition the unlabeled traffic patterns into an appropriate number of clusters and then develops the genetic programming model for each cluster to predict its corresponding traffic features. For demonstration, the proposed method is tested against a 110-km freeway stretch, on which 48 time steps of 5-min traffic flows are predicted (i.e., a 4-h prediction). The prediction accuracy of the proposed method is compared with other models (ARIMA, SARIMA, and naive models) and the results support the superiority of the proposed method. Further analyses indicate that applications of the proposed method to larger scale freeway networks require sufficient lengths of observation to acquire enough traffic patterns for training and validation in order to achieve higher prediction accuracy.

Keywords Genetic Programming; Growing Hierarchical Self-Organizing Map; Rolling Self-Structured Traffic Patterns; Traffic Prediction

INTRODUCTION

The advanced traveler information system (ATIS) is one of the most widely deployed intelligent transportation systems (ITS) applications nowadays. ATIS provides advisory information to the road users about traffic regulation, route and location guidance, hazardous situations, safety advisory, and warning messages (Kumar et al., 2005). It requires fast processing, analyzing and storing a large amount of traffic data, and quickly disseminating both pretrip and en route information to users. The pretrip information can enhance drivers' self-belief while using the freeways or allow commuters to make better informed transit choices (Campbell et al., 2003). The en route information, on the other hand, can save users' travel time, help travelers avoid congestion, and thus improve traffic network performance (Maccubbin et al., 2003). The implementation of

ATIS is heavily dependent on the accurate estimates of prevailing and emerging traffic conditions. In the literature, a variety of predictive techniques/algorithms have been developed; however, no matter how advanced the predictive techniques/algorithms are, time lags seem inevitable while converting the detected data into the real applications. The lags typically come from the times required in cumulating, analyzing, and transmitting the data. To cope with the time lags, pragmatic ITS applications must be based on predicted traffic conditions, in lieu of detected traffic conditions, making the prediction of traffic features a prerequisite (Lam et al., 2005; Lan et al., 2010).

Traffic features such as flow, speed, occupancy, and travel time can vary over space and over time. In the past, a considerable number of methods or models have been developed to predict traffic features. Travelers are most interested in the travel time along their paths. Unless the roadway systems have sufficient number of probe vehicles, stationary vehicle detectors (VD) are still the most prevalent and reliable devices to collect traffic information nowadays. However, the traffic information collected by VD makes it hard to accurately predict travel time over a long stretch because VD can only measure the temporally distributed traffic information. Nonetheless, with predicted link

Address correspondence to Yu-Chiun Chiou, Department of Transportation and Logistics Management, National Chiao Tung University, 4F, 118 Sec. 1, Chung-Hsiao W. Rd. Taipei 10012, Taiwan, ROC. E-mail: yechiou@mail.nctu.edu.tw

Color versions of one or more figures in this article can be found online at www.tandfonline.com/gits.

traffic flows or on-ramp traffic flows over a sufficient time period ahead, a dynamic origin–destination (O–D) estimating coupled with an efficient traffic flow modeling (e.g. Chiou et al., 2010) can predict traffic flows and travel time for each O–D pair accordingly. In other words, the predicted traffic flows can further predict the travel time information through properly processing the dynamic traffic assignment and flow modeling.

In the past, numerous studies attempted to predict traffic flows based on VD data or to predict travel time based on probe vehicle data for ATIS development or traffic management. These studies broadly employed parametric and non-parametric models. The parametric models include the linear statistical model (Chang & Miaou, 1999), nonlinear statistical model (Lan et al., 2007, 2010; Vlahogianni, 2009; Vlahogianni et al., 2006, 2008), support vector regression (Hong, 2011), and time-series model (Ghosh et al., 2005; Kamarianakis & Prastakos, 2003; Szeto et al., 2009; Van Der Voort et al., 1996; Williams & Hoel, 2003). The nonparametric models include the sequential learning model (Chen & Grant-Muller, 2001; Sheu et al., 2009a, 2009b), nearest neighbor approach (Smith & Demetsky, 1997; You & Kim, 2000), neural network model (Chang et al., 2002; Chen et al., 2001; Dia, 2001; Dochy et al., 1996; Dougherty and Cobbett, 1997; Kirby et al., 1997; Lan & Huang, 2006; Ledoux, 1997; Smith & Demetsky, 1997; Smith et al., 2002; Van Der Voort et al., 1996; Vlahogianni et al., 2005), spinning network (Huang & Sadek, 2009), Kalman filtering model (Okutani & Stephanedes, 1984), gray prediction model (Chiou et al., 2007), and adaptive fuzzy rule-based model (Dimitriou et al., 2008; Zhang & Ye, 2008). Most of these methods or models, however, focused on short-period prediction—only one or two time intervals ahead—with an inherent assumption of nearly the same traffic pattern throughout the prediction horizon. Should the prediction period becomes longer over which traffic patterns change manifestly, the preceding methods or models will no longer be applicable. For instance, if one wishes to develop an ATIS by providing the freeway users with reliable predicted en route travel time information over few hundred kilometers, one requires predicting the relevant traffic features for the entire stretch several hours ahead—this refers to a “long prediction horizon” in this study. In the literature, relatively few models are designed for predicting traffic features over such a long prediction horizon. The genetically optimized neural network model (Vlahogianni et al., 2005) that can predict over five time-steps ahead and the multivariate structural time-series model (Ghosh et al., 2009) that can predict even longer ahead (1 day ahead for daily model, 1 week ahead for weekly model) are perhaps two exceptions.

Although traffic dynamics may change drastically, similar traffic patterns do repeat over the spatiotemporal domains (Lan et al., 2008; Lan et al., 2010; Sheu et al., 2009a, 2009b). Therefore, if the historical baseline data can be clustered into an appropriate number of traffic patterns, it becomes possible to accurately predict the traffic features for each cluster in a rolling-horizon manner. The objective of this study is to

propose a novel method, based on the rolling self-structured traffic patterns, to predict traffic features over a “long prediction horizon” along a freeway stretch. The core logics of the proposed method contain a growing hierarchical self-organizing map (GHSOM) model and a genetic programming (GP) model. The GHSOM model is to group the similar traffic patterns into the same cluster without the need of predetermination of the number of clusters, whereas the GP model is to perform the traffic prediction in each cluster without the need of prior assumption of data distribution or model specification. To test the proposed method, a case study is demonstrated on a 110-km freeway stretch on which 48 time intervals of 5-min traffic flows are predicted.

The rest of the article is organized as follows. The second section describes the core logics and solution algorithms of the proposed method. The third section presents the case study with training and validation results. The fourth section further evaluates the predictive performance of the proposed method. Finally, concluding remarks and suggestions for future studies are addressed.

THE PROPOSED METHOD

A Rolling Concept

Let $X_k(t+1) = [x_k(t+1), x_k(t+2), \dots, x_k(t+r)]$ denote a sequence of r -period historical traffic features (e.g., 5-min flow rates in the case study) observed at location k starting from time $t+1$. The proposed method aims to predict the flow features h time steps ahead, denoted as $\hat{x}_k(t+r+1), \hat{x}_k(t+r+2), \dots, \hat{x}_k(t+r+h)$. For a shorter period prediction (i.e., $h = 1$ or 2 in most of the previous literature), the predicted results have provided useful information for some ITS applications like traffic-responsive control. However, if the purpose is for ATIS applications, we may need a longer period prediction (i.e., $h \gg 1$) of the preceding flow features. To this end, this study aims to make a longer-period prediction of $\hat{x}_k(t+r+1), \hat{x}_k(t+r+2), \dots, \hat{x}_k(t+r+h)$. (e.g., $h = 48$ time steps of 5-min flow rates in the case study).

A rolling-horizon concept is incorporated into the proposed method. First, we employ the GHSOM model to classify the given traffic patterns into appropriate clusters. Then we use the GP model to predict traffic sequence in each cluster such that a portion of the most updated $(r-s+1)$ time steps of traffic sequence, say $x_k(t+s), x_k(t+s+1), \dots, x_k(t+r)$, is used to predict the traffic sequence at the very next time step, $\hat{x}_k(t+r+1)$, where s is the specific time point. In the GP model, only the time steps after s are selected to predict the consecutive traffic sequences. This predicted traffic sequence together with the previous traffic sequences is further used to predict the consecutive traffic sequences in a rolling manner. Specifically, the traffic feature $\hat{x}_k(t+r+1)$ is predicted based on traffic sequences $x_k(t+s), x_k(t+s+1), \dots, x_k(t+r)$; the traffic feature $\hat{x}_k(t+r+2)$ is predicted based on traffic

sequences $x_k(t + s + 1)$, $x_k(t + s + 2), \dots, x_k(t + r)$, $\hat{x}_k(t + r + 1)$; and so on.

Taking $r = 100$, $s = 81$, and $h = 25$ as an example, in total 100 traffic sequences are used to cluster the traffic pattern by the GHSOM model and in total the 20 most updated traffic sequences from 81st to 100th are used to predict the 101st traffic sequence by the GP model. Following the same vein, the 102nd traffic sequence is predicted by newly updated 20 traffic sequences—the 19 historical traffic sequences (from 82nd to 100th) plus the predicted traffic sequence (101st). A similar process continues until 25 subsequent traffic sequences have been completely predicted. The details of traffic pattern clustering and traffic prediction are further elaborated in the following.

Traffic Pattern Clustering

Pattern clustering is also known by different terms (e.g., cluster analysis, set partitioning, Q-analysis, typology, grouping, clumping, classification, numerical taxonomy, or unsupervised pattern recognition) in different literature. In traffic literature, traffic pattern at a specific location represents a sequence of traffic features such as flow, speed, occupancy, and so on; thus, traffic pattern clustering can be regarded as a classification process by which a group of unlabeled traffic patterns are partitioned into a number of sets—similar patterns in the same cluster and dissimilar patterns in different clusters.

Brucker (1978) and Welch (1983) proved that, for specific objective functions, clustering becomes an NP-hard problem when the number of clusters exceeds three, if one aims to find the optimal clusters. Numerous heuristic algorithms for clustering have been developed, which can generally be divided into five categories: statistics clustering, mathematical programming, network programming, neural network (Chen et al., 2008; Tangsripiroj & Samadzadeh, 2006; Yang et al., 2004, 2010), and metaheuristics (Chiou & Chou, 2010; Chiou & Lan, 2001). Rauber et al. (2002) proposed the growing hierarchical self-organized map (GHSOM) model and proved that it possesses excellent performance in pattern clustering; thus, this study employs GHSOM to conduct the traffic patterns clustering.

In fact, GHSOM is an extension of the self-organizing map (SOM), an artificial neural network that performs clustering by means of unsupervised competitive learning algorithm proposed by Kohonen (1982). During the learning process, the network performs clustering and the model vectors change to reflect the similarity of the neighboring clusters. The goal of the SOM is to represent the points in the source space by corresponding points in a lower dimensional target space (often in a two-dimensional lattice). However, the SOM cannot determine the size of a preset map by ignoring the characteristics of data distribution. In other words, when using the SOM to group traffic patterns into clusters, the number of clusters must be predetermined. Without an in-depth investigation of the

historical traffic patterns, it is difficult to know the appropriate number of clusters in advance. In contrast, GHSOM (Rauber et al., 2002) has a hierarchical structure of multiple layers, where each layer consists of several independent growing SOMs. GHSOM can automatically determine the optimal number of clusters through the growth of layers and maps, and thus it can enhance the applicability of our proposed model.

The GHSOM architecture starts from a top-level map, which grows in size to represent a collection of data at different specific levels. For instance, Layer 1 contains 2×2 units and provides a rather rough organization of the main clusters in the input data. The four independent maps in Layer 2 give more detailed information. The three identified units in Layer 2, which have diversified input data mapping onto them, are further expanded to form a new independent SOM in the subsequent layer (Layer 3), and so on, depicted in Figure 1.

To elucidate the training algorithm of GHSOM, the training algorithm of conventional SOM is given next. A typical SOM network consists of an input layer and an output or competitive layer. The input layer is composed of a set of r -dimensional input vectors $x_k = [x_k(t + 1), x_k(t + 2), \dots, x_k(t + r)]$, where r indicates the number of traffic features (i.e., the flow sequences at consecutive time steps in this study) contained in each input vector. The output layer is an m -dimensional (often $m = 2$) grid, which consists of a set of neurons, each associated with an r -dimensional weight vector $w_i = [w_{i1}, w_{i2}, \dots, w_{ir}]$ with the same dimension as the input vector. The arrangement of the neurons can be rectangular or hexagonal. Conceptually, SOM takes a set of inputs mapping them onto the neurons of two-dimensional grid. Randomly initializing the weight vectors, the SOM network then performs learning as the following steps.

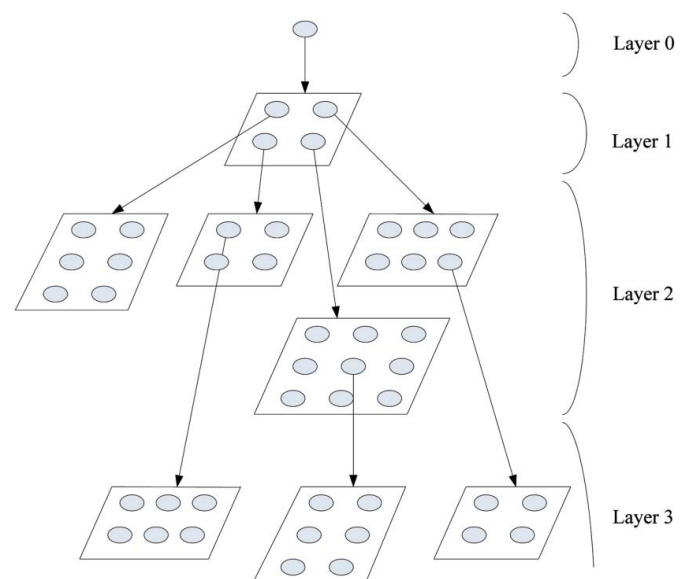


Figure 1 An illustration of the GHSOM architecture.

- Step 1: Randomly initialize the weight vector of each neuron.
- Step 2: Determine the winning neuron. The SOM network determines the winning neuron for a given input vector, selected randomly from the set of all input vectors. For every neuron on the grid, its weight vector is compared with the input vector by using some similarity measures, for example, the Euclidean distance. The neuron for which the weight vector is closest to the input vector is selected as the winning neuron b , expressed as follows:

$$b : \|x_{kt} - w_b(l)\| = \min_i \{\|x_{kt} - w_i(l)\|\}, \quad (1)$$

where l denotes the number of current learning iteration.

- Step 3: Update the weights. After a winning neuron is determined, the weight vectors of winning neuron along with its neighboring neurons are updated so as to “move” toward the input vectors according to the following equation:

$$w_i(l+1) = w_i(l) + h_{bi}(l)(x_{kt} - w_i(l)), \quad (2)$$

where $h_{bi}(l)$ is the neighborhood function. A widely used neighborhood function is based on the Gaussian function expressed as follows:

$$h_{bi}(l) = \alpha(l) \exp\left(-\frac{\|r_i - r_b\|^2}{2\sigma(l)^2}\right), \quad (3)$$

where $\alpha(l)$ is the learning rate function, which controls the amount of weight vector adjustment and decreases with the iterations; r_i and r_b are the locations of the neuron i and winning neuron b in the lattice; and $\sigma(l)$ defines the width of the neighborhood function and it also decreases monotonically.

- Step 4: Test the stop condition. Steps 2 and 3 are repeated until all the patterns in the training set have been processed. In addition, to achieve a better convergence toward the desired mapping, it is usually required to repeat the previous loop until some convergence criteria are met.

Based on the concept of the preceding SOM learning process, the training algorithm of GHSOM basically grows horizontally (by increasing the size of each SOM) and hierarchically (by increasing the number of layers). In horizontal growth, each SOM modifies itself in a systematic way similar to the growing grid so that each neuron does not represent too large an input space. In the hierarchical growth, on the other hand, the principle is to periodically check whether the lowest layer of SOMs have achieved sufficient coverage for the underlying input data. The basic steps of horizontal growth and hierarchical growth of GHSOM are delineated next (Tangsrapiroj & Samadzadeh, 2006):

Horizontal growth:

- Step 1: Randomly initialize the weight vector of each neuron.
- Step 2: Perform the conventional SOM learning algorithm for a preset number of iterations.
- Step 3: Find the error unit e and its most dissimilar neighbor unit d . The error unit e is the neuron with the largest deviation between its weight vector and the input vectors it represents.
- Step 4: Insert a new row or a new column between e and d . The weight vectors of these new neurons are initialized as the average of their neighbors.
- Step 5: Repeat Steps 2–4 until the mean quantization error of the map (MQE_m) is less than $(\tau_1 \cdot mqe_u)$. Here, τ_1 is a threshold specifying the desired level of detail to be shown in a particular SOM; mqe_u is the mean quantization error of the neuron u in the preceding layer of the hierarchy. Equation 4 calculates mqe_u , which is the average distance between the weight vector of neuron u and the input vector mapping onto this neuron:

$$mqe_u = \frac{1}{n_{C_u}} \sum_{x_j \in C_u} \|x_j - w_u\|, \quad n_{C_u} = |C_u|, \quad (4)$$

where C_u denotes the set of input vectors mapping onto unit u ; w_i denotes the weight vector of unit i ; $\|x_j - w_u\|$ denotes the distance between input vector x_j and weight vector w_u ; and $|C_u|$ denotes the cardinality of the set C_u . Furthermore, MQE_m , the mean of all neurons’ quantization errors in the map, is calculated as follows:

$$MQE_m = \frac{1}{n_U} \sum_{i \in U} mqe_i, \quad n_U = |U|, \quad (5)$$

where U denotes the subset of map units.

Hierarchical growth:

- Step 1: Check each neuron to find out whether its mqe_u is greater than $(\tau_2 \cdot mqe_0)$. Here, τ_2 is a threshold specifying the desired quality of input data representation at the end of learning process; mqe_0 is the mean quantization error of the single neuron of Layer 0. Then, assign a new SOM at a subsequent layer of the hierarchy. mqe_0 is computed as follows:

$$mqe_0 = \frac{1}{n_I} \sum_{x_j \in I} \|x_j - m_0\|, \quad n_I = |I|, \quad (6)$$

where m_0 is the mean of the input vectors and I is the set of the input vectors. mqe_0 is regarded as a measurement of the overall dissimilarity of input data.

- Step 2: Train the SOM with input vectors mapping onto this neuron.

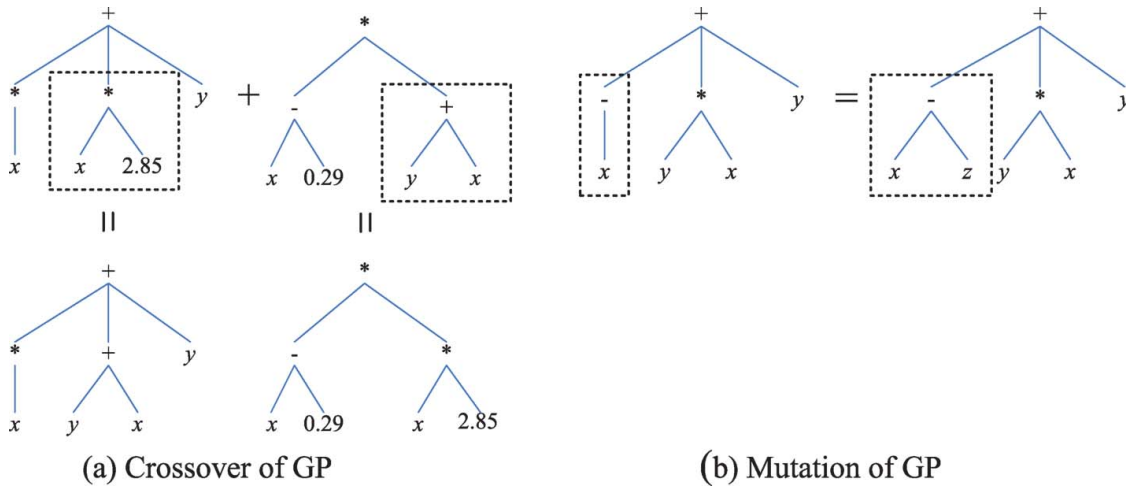


Figure 2 Crossover and mutation operations of GP: (a) crossover of GP; (b) mutation of GP.

Traffic Prediction

After dividing the traffic patterns into appropriate number of clusters, a traffic prediction model is then developed for each cluster, which predicts h time steps ahead based on historical r time steps. It is well known that the artificial neural network (ANN), especially the back propagation network, is a powerful tool for traffic prediction. The performance of ANN for predicting traffic flows has been proven by many studies. However, if an ANN model is introduced in the case study, the network will consist of 240 input neurons and one output neuron, requiring a lot of effort in network pruning and searching for the optimal number of hidden layers and hidden neurons. Additionally, 240 traffic sequences must be inputted into the tuned network for predicting each traffic sequence, making the

network training and prediction troublesome. In contrast to ANN model, the GP model is a global optimization algorithm based on the mechanism of natural selection and offspring generation (Koza, 1992). It starts with a population of randomly generated individual trees, each corresponding to a linear combination of traffic flows in the previous periods. Every generated tree is evaluated for its fitness value, which is further utilized for the selection of generated offspring trees. The tuned GP model requires only a few historical traffic sequences for traffic prediction, which is apparently much easier than the ANN model.

Assume that in total I traffic patterns are to be assigned to cluster l ; each traffic pattern is denoted as $X_{li}(t) = [x_{li}(1), x_{li}(2), \dots, x_{li}(r)]$, $i = 1, 2, \dots, I$. The learning process of GP model is as follows:

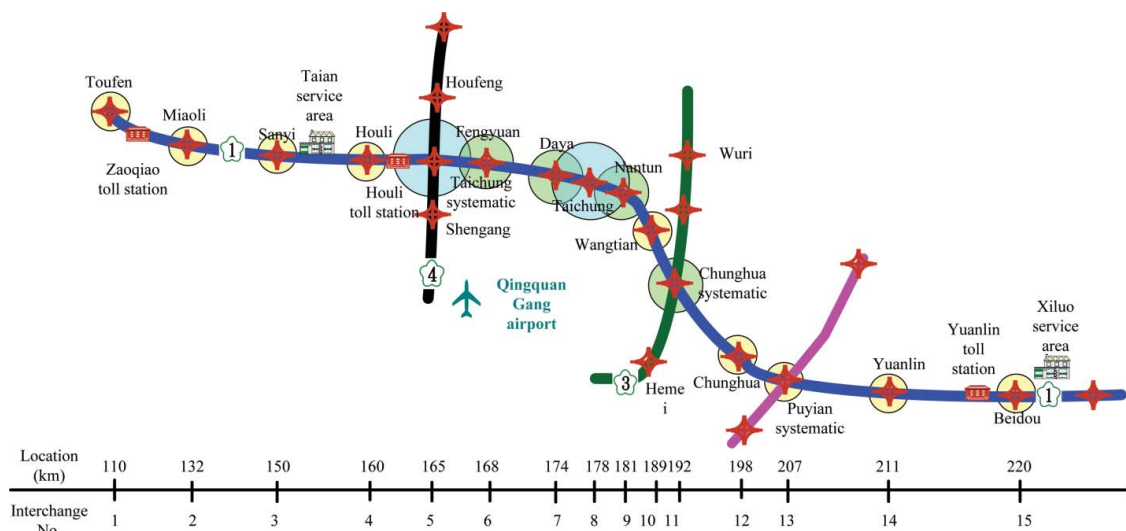
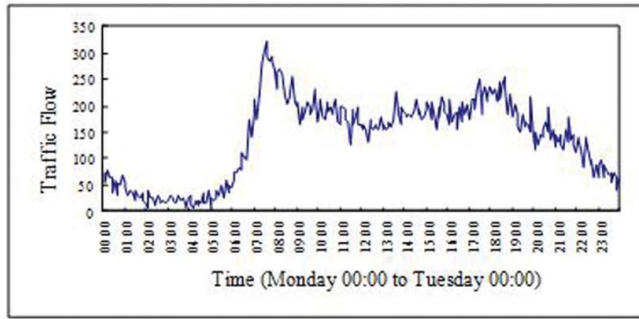
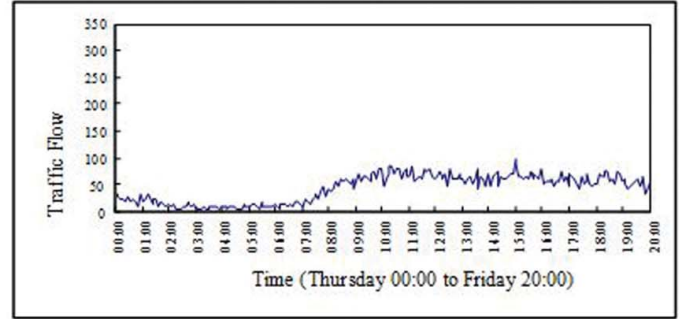


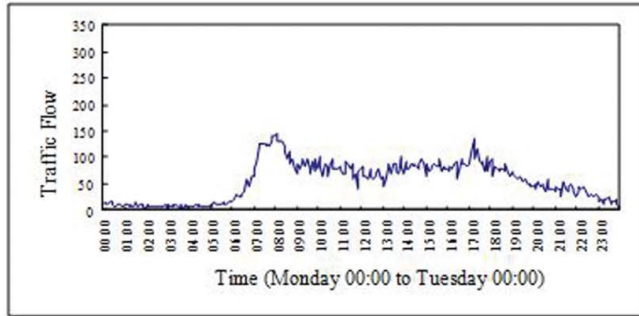
Figure 3 The study corridor.



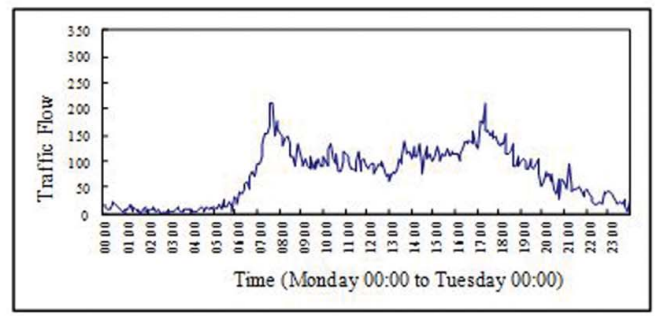
(a) Taichung Interchange



(b) Wangtian Interchange

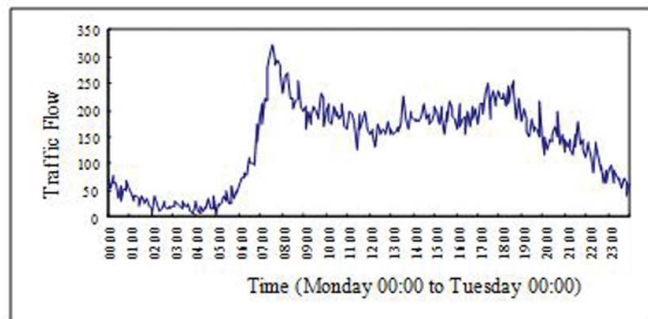


(c) Fengyuan Interchange

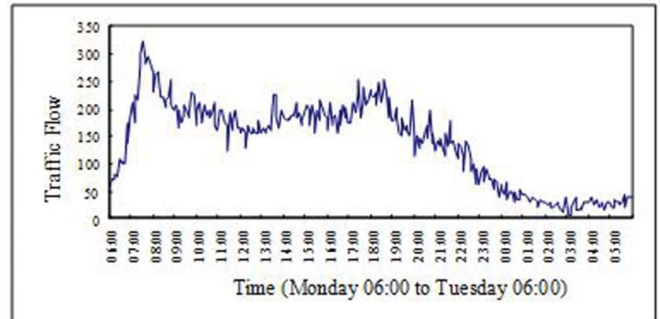


(d) Taichung system Interchange

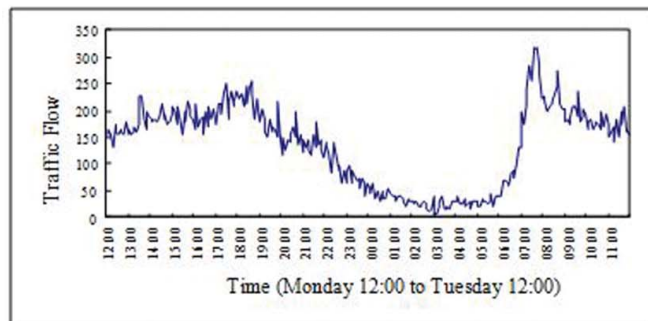
Figure 4 Traffic patterns at different interchanges (Wednesday 00:00 to Thursday 00:00).



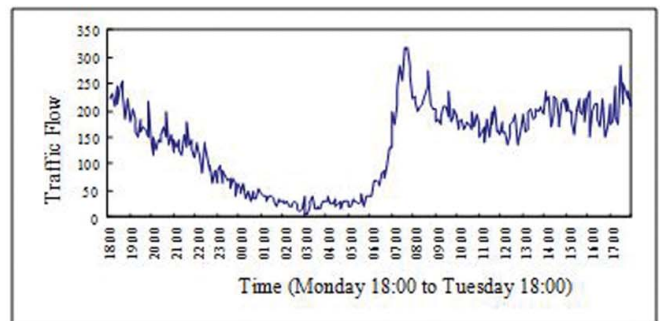
(a) Monday 00:00 to Tuesday 00:00



(b) Monday 06:00 to Tuesday 06:00



(c) Monday 12:00 to Tuesday 12:00



(d) Monday 18:00 to Tuesday 18:00

Figure 5 Traffic patterns at different time periods (Taichung Interchange).

Step 0: Define function set and terminal set. The function set consists of the arithmetic functions of addition, subtraction, multiplication, and division, as well as a conditional branching operator. The terminal set is set as the latest r time steps of traffic flow data.

Step 1: Initialize random population size.

Step 2: Evaluate fitness values of the trees. Randomly select trees from the population, evaluate them with training patterns belonging to this cluster, and then rank them according to their fitness values. A fitness measure is defined as follows:

$$E_{lq} = \sqrt{\frac{\sum_{i=1}^I \sum_{t=1}^h (x_{li}(t+r+1) - f_q(X_{li}(t)))^2}{h}}, \quad (7)$$

where $f_q(\cdot)$ denotes the mathematical expression of tree q , which predicts the traffic flow at next time step based on the inputted historical data at previous time steps, that is, $f_q(X_{li}(1)) = \hat{x}_{liq}(t+r+1)$.

Step 3: If the fitness value approaches to zero, then stop the procedure. Otherwise, proceed to the next step.

Step 4: Create a new traffic pattern by applying genetic operations: reproduction, crossover, and mutation.

Step 4-1: Reproduction. Replace the two traffic patterns with least fit, by the two with best fit.

Step 4-2: Crossover. Create a new offspring by randomly combining the chosen parts of two selected trees in each parent tree and swapping the subtree rooted at crossover points, illustrated in Figure 2a.

Step 4-3: Mutation. Randomly select a mutation point in a tree and substitute the subtree rooted there with a randomly generated sub-tree, illustrated in Figure 2b.

Step 5: Generate new population by using genetic operations, and return to Step 2.

Once a new traffic pattern with r time steps is collected, it automatically assigns to the closest cluster, into which all traffic patterns are classified by GHSOM. The traffic pattern is then fed into the tuned GP model in this cluster to predict the next h time steps in a rolling manner.

CASE STUDY

The Data

The southbound 5-min on-ramp traffic flow data at 15 interchanges from Toufen Interchange to Beidou Interchange, a 110-km stretch of Taiwan No. 1 Freeway (Figure 3), over a week from May 25 to May 31 (Monday through Sunday), 2009, were used in this case study. At each interchange, traffic flows are first aggregated from different ramps. The on-ramp

Table 1 Parameter settings for the GP model.

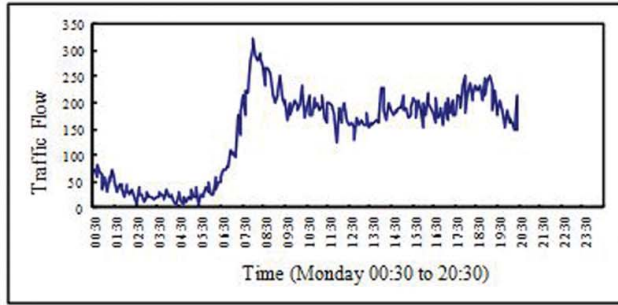
Parameter	Setting
Fitness	Mean square error
Terminal set	$x(t), x(t-1), \dots, x(t-240)$ and random number b
Function set	+, -, ×
Population size	50
Reproduction rate	0.08
Crossover rate	0.60
Mutation rate	0.01
Initial minimum depth	2
Number of generations	300
Initialization method	Direct method

traffic pattern is composed of 288 consecutive 5-min traffic flow data in 1 day (24 h). In total, 2016 time intervals in 1 week can form 1729 (= 2016 - 287) traffic patterns at each interchange. The remaining 287 time steps cannot form a complete traffic pattern and thus are discarded. As a result, in total 25,935 (= 1729 × 15) traffic patterns have been generated for the entire study corridor.

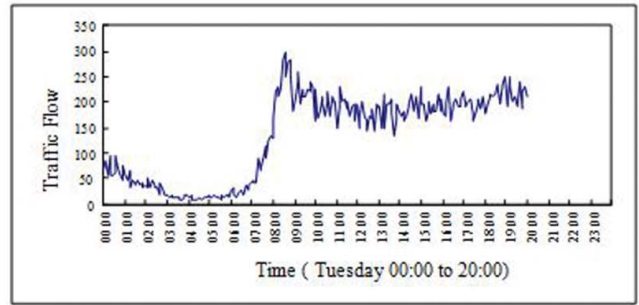
With a 5-min time-step interval, the proposed method aims to predict the next 48 time steps (a 4-h horizon) based on the previous 240 time steps (i.e., 20 h). In other words, the previous 240 traffic flow data are used to determine the closest cluster and then fed into the corresponding tuned GP model to predict the next 48 time-step traffic flows in a rolling manner. Figure 4 demonstrates an example of traffic patterns during the same periods (Wednesday 00:00 to Thursday 00:00) at different interchanges, which shows that the traffic patterns differ remarkably from each other. For instance, Taichung Interchange and Taichung System Interchange do

Table 2 Self-structured traffic patterns in each of the 36 clusters.

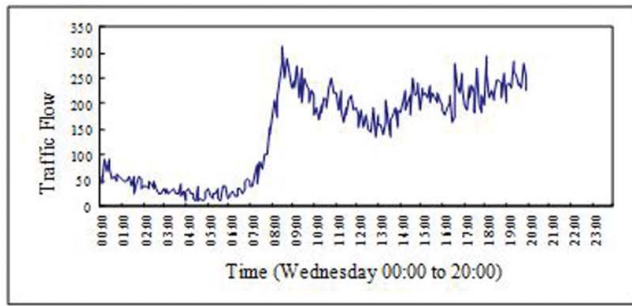
Cluster	Number of patterns	Cluster	Number of patterns
1	143	19	155
2	381	20	214
3	429	21	429
4	1238	22	607
5	440	23	1381
6	2488	24	429
7	155	25	155
8	155	26	238
9	179	27	429
10	405	28	464
11	1298	29	381
12	1238	30	345
13	155	31	155
14	167	32	167
15	464	33	214
16	417	34	179
17	1333	35	321
18	321	36	488



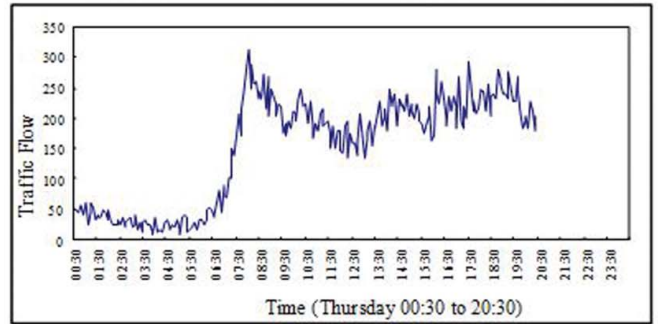
(a) Taichung Interchange (Monday 00:30 to Monday 20:30)



(b) Taichung Interchange (Tuesday 00:00 to Tuesday 20:00)



(c) Taichung Interchange (Wednesday 00:00 to 20:00)



(d) Taichung Interchange (Thursday 00:30 to 20:30)

Figure 6 Four randomly selected traffic patterns from Cluster 1 (urban area).

exhibit significant peak and off-peak traffic patterns, but Wangtian Interchange and Fengyuan Interchange do not.

More detailed traffic patterns at Taichung Interchange during different time periods are further illustrated in Figure 5. This reveals that the traffic patterns at a specific location can be significantly different over time, but similar patterns do repeat themselves over time.

To train and validate the proposed method, the traffic patterns are randomly divided into two sets: a training set (18,155 traffic patterns) and a validation set (7780 traffic patterns), at a ratio of 7:3. The parameter settings and the results of clustering and prediction are presented next.

Parameter Settings

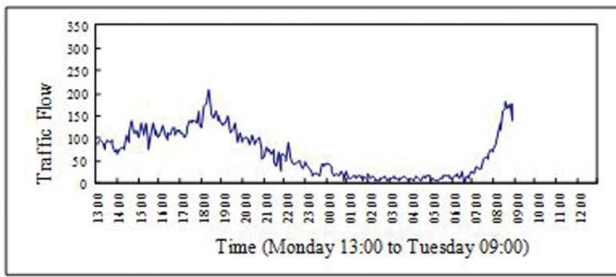
Referring to Rauber et al. (2002), the parameters of the GHSOM model are set as follows: $\tau_1 = 0.85$, $\tau_2 = 0.0035$, and both learning rate function $\alpha(l)$ and neighborhood function $\sigma(l)$ are set as linear monotonically decreasing with iterations. Further referring to Yao and Lin (2009), the parameters of the GP model are set as shown in Table 1. To avoid complicated

traffic prediction function, only three operators (+, -, and \times) are considered in this study. Terminal set contains the traffic flow data in the previous 240 time-step intervals with a randomly generated number b .

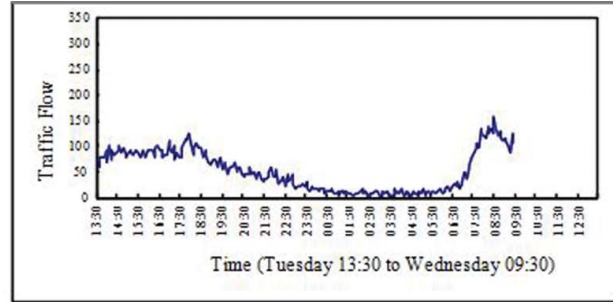
Clustering Results

In total, 4 layers with 36 different clusters have been identified by GHSOM. The number of traffic patterns in each of the 36 clusters ranges from 143 (Cluster 1) to 2488 (Cluster 6), detailed in Table 2. Such self-structured traffic patterns will be used for prediction in the GP model.

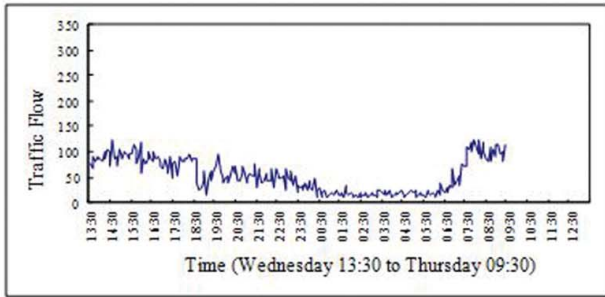
To display the similarity of traffic patterns in the same cluster, the traffic patterns in Cluster 1 (urban area), Cluster 15 (suburban area), and Cluster 30 (rural area) are demonstrated in Figures 6, 7, and 8, respectively. To avoid lengthy discussion, we only present four randomly selected traffic patterns from each of such three clusters. In Figure 6, Cluster 1 contains traffic patterns starting from 00:00 to 20:00 on weekdays in the urban area (e.g., Taichung Interchange and Taichung



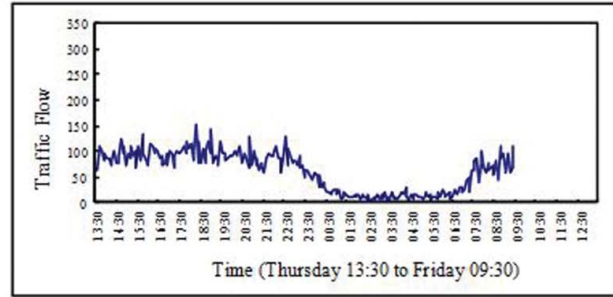
(a) Chunghua Interchange (Monday 13:00 to Tuesday 9:00)



(b) Fengyuan Interchange (Tuesday 13:30 to Wednesday 9:30)

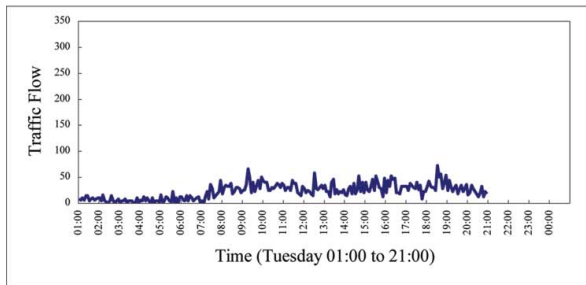


(c) Daya Interchange (Wednesday 13:30 to Thursday 9:30)

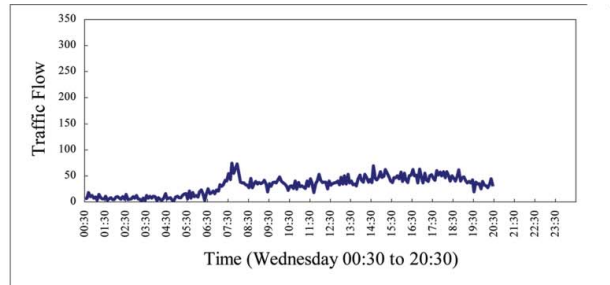


(d) Nantun Interchange (Thursday 13:00 to Friday 9:00)

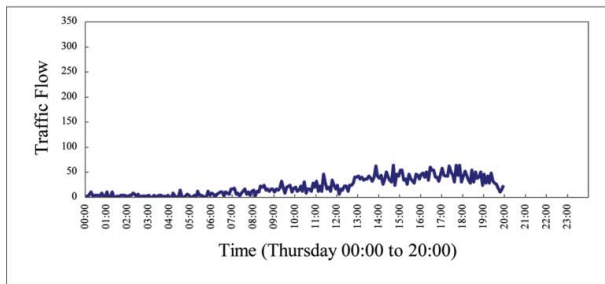
Figure 7 Four randomly selected traffic patterns from Cluster 15 (suburban area).



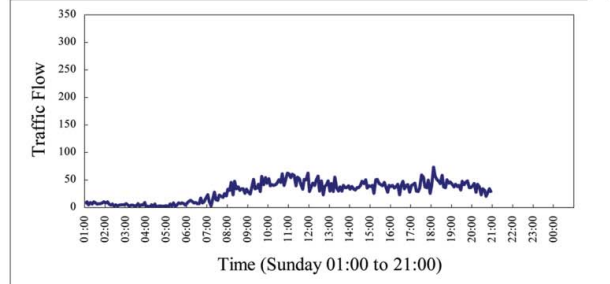
(a) Toufen Interchange (Tuesday 01:00 to 21:00)



(b) Miaoli Interchange (Wednesday 00:30 to 20:30)

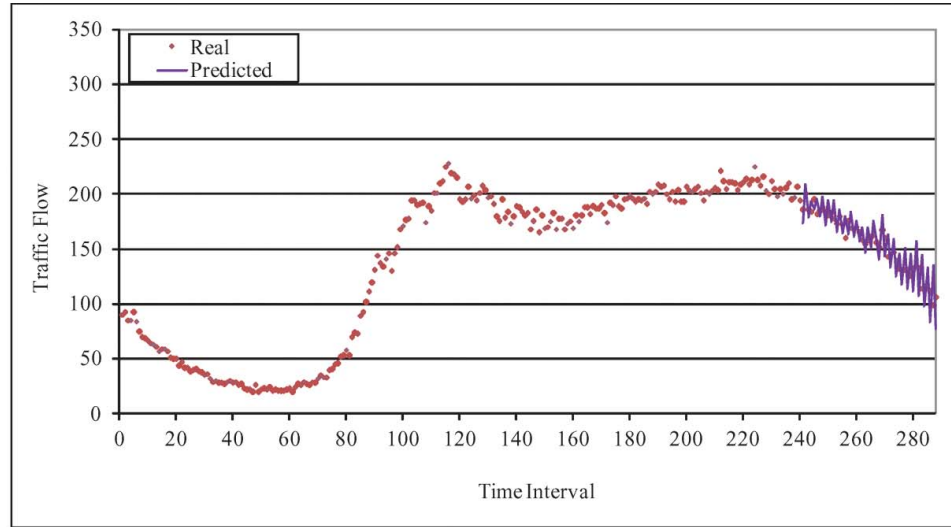


(c) Sanyi Interchange (Tuesday 00:00 to 20:00)

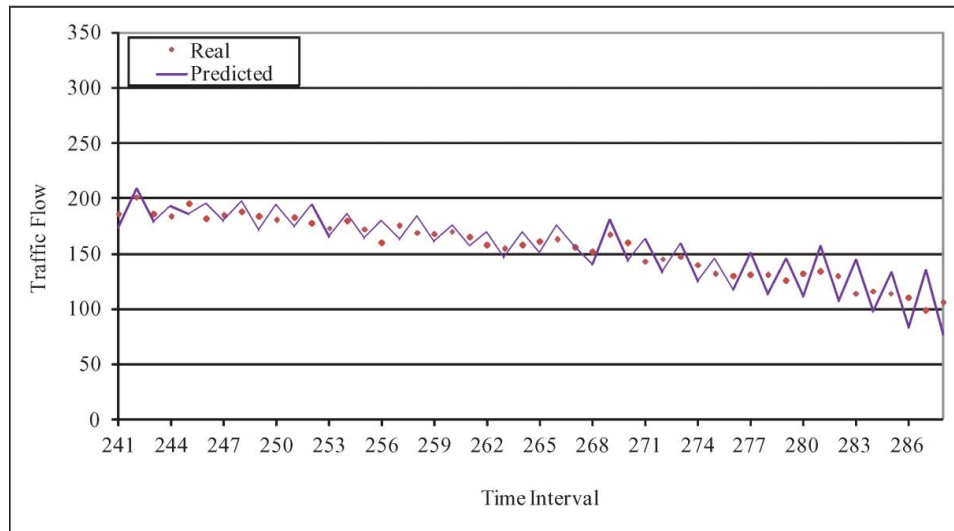


(d) Wangtian Interchange (Sunday 01:00 to 21:00)

Figure 8 Four randomly selected traffic patterns from Cluster 30 (rural area).



(a) A complete horizon (24 h)



(b) A zoom-in plot of predicted horizon (4 h)

Figure 9 Real and predicted traffic of a randomly selected traffic pattern from Cluster 1.

System Interchange) where maximum 5-min flow rates can exceed 300 passenger car units (pcu).

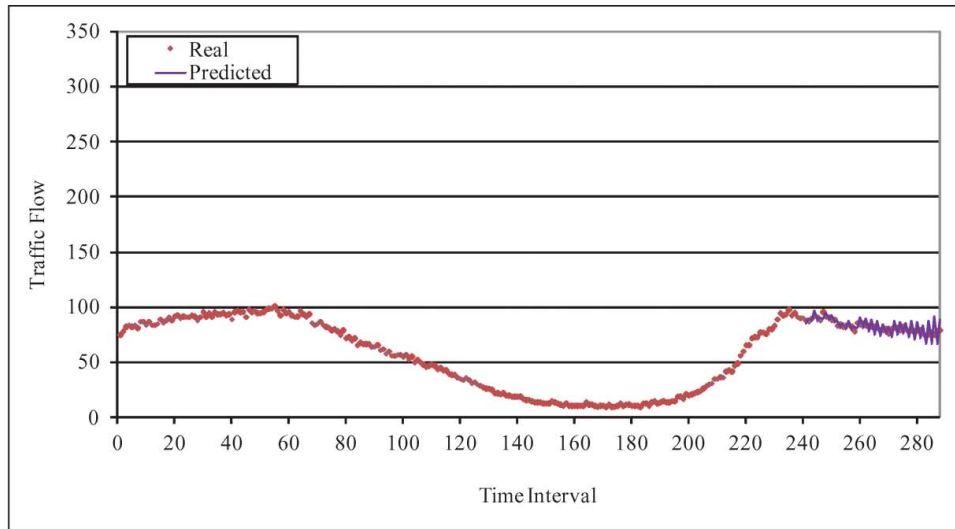
In Figure 7, Cluster 15 contains traffic patterns starting from 13:00 to 09:00 on weekdays in the suburban area (e.g. Chunghua, Fengyuan, Daya, and Nantun Interchanges) where most of the 5-min flow rates are below 150 pcu. It is obvious that the peak and off-peak phenomena of traffic patterns in Cluster 15 are not as sharp as those in Cluster 1.

In Figure 8, Cluster 30 contains traffic patterns starting from 00:00 to 20:00 on weekdays or weekends in the rural area (e.g., Toufen, Miaoli, Sanyi, and Wangtian Interchanges) where most of the 5-min flow rates are lower than 50 pcu. No significant peak/off-peak phenomenon can be identified in this cluster.

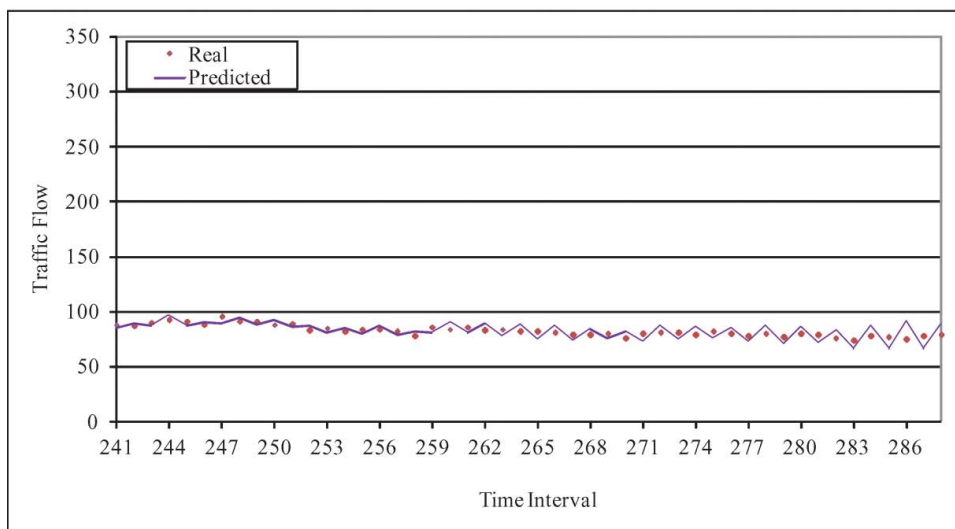
Based on the clustering results, the traffic patterns in the same cluster are similar and those in different clusters are significantly dissimilar, suggesting the correctness of our clustering model.

Prediction Results

Based on the self-structured traffic patterns associated with the 36 clusters, in total 36 GP traffic prediction models are further developed, with each cluster having one prediction model. Here, we only explain three clusters (1, 15, and 30). In Cluster 1, 143 traffic patterns are contained and randomly



(a) A complete horizon (24 h)



(b) A zoom-in plot of predicted horizon (4 h)

Figure 10 Real and predicted traffic of a randomly selected traffic pattern from Cluster 15.

divided into a training set (100 patterns) and a validation set (43 patterns). Based on the training traffic patterns, the GP model for Cluster 1 is finally tuned as follows:

$$\begin{aligned}
 x(t+1) = & 1.01x(t) + 5.41 \times 10^{-6}x(t-1)x(t-2)x(t-7) \\
 & - 1.75 \times 10^{-6}x(t-1)x(t-4)^2 - 2.02 \times 10^{-8}x(t) \\
 & \times 3x(t-6) + 4.05 \times 10^{-8}x(t-1)x(t-5)^3.
 \end{aligned}
 \tag{8}$$

According to Eq. 8, only the traffic flow data at the previous seven time steps are required, without $(t-3)$, to predict the

traffic flow of next time step. For instance, to predict the traffic flow rate in Figure 6a at time step $(t+1)$, say, 20:05, we need to input 7 detected flow rates at $x(t-7) = 19:25$, $x(t-6) = 19:30$, $x(t-5) = 19:35$, $x(t-4) = 19:40$, $x(t-2) = 19:50$, $x(t-1) = 19:55$, and $x(t) = 20:00$. The traffic flow rate at 20:05 can therefore be calculated as $x(t+1)$ according to Eq. 8. To predict in a rolling-horizon manner for the next time step $(t+2)$ at 20:10, the 6 detected flow data from 19:30 to 20:00 together with the already-predicted traffic flow at 20:05 are inputted into Eq. 8. This process continues until all traffic data for the next 4 h (48 time steps) have been obtained.

Following the same vein, the GP models for Cluster 15 and Cluster 30 are finally tuned as follows, respectively:

$$\begin{aligned}
 x(t+1) = & 1.0483x(t) - 0.0942x(t-2) + 0.0031x(t-1) \\
 & \times x(t-2) - 4.897 \times 10^{-5}x(t-5)x(t-7)^2 \\
 & - 3.73310^{-7}x(t)^2x(t-1)x(t-2) + 3.165 \times 10^{-7} \\
 & \times x(t-1)x(t-2)x(t-5)x(t-7) + 1.255 \times 10^{-7} \\
 & \times x(t-4)x(t-7)^3 + 4.97 \times 10^{-8}x(t)x(t-2)x(t-7)^2 \\
 & + 1.038 \times 10^{-7}x(t-1)x(t-4)x(t-5)x(t-6). \tag{9}
 \end{aligned}$$

$$\begin{aligned}
 x(t+1) = & 0.8301x(t) + 0.166x(t-2) + 0.0195x(t)x(t-1) \\
 & - 0.0181x(t)x(t-2) - 1.24 \times 10^{-6}x(t-6) \\
 & \times x(t-7)^2 - 4.53 \times 10^{-6}x(t)x(t-1)x(t-2)^2 \\
 & - 3.81 \times 10^{-6}x(t-1)^2x(t-5)x(t-6) - 3.91 \\
 & \times 10^{-6}x(t-1)x(t-5)x(t-6)^2 - 3.55 \times 10^{-6} \\
 & \times x(t)^2x(t-3)x(t-6) + 7.51 \times 10^{-6}x(t-2)^2 \\
 & \times x(t-5)x(t-6) + 4.16 \times 10^{-6}x(t)^2x(t-1) \\
 & \times x(t-3) + 3.74 \times 10^{-6}x(t-1)^2x(t-6)^2. \tag{10}
 \end{aligned}$$

According to Eqs. 9 and 10, Clusters 15 and 30 require the inputs of traffic flow data at the previous 6 and 7 time steps, respectively. Among the 36 GP traffic prediction models, our results require the inputs of traffic flow data at the previous 12 time steps at most.

Figures 9–11 respectively show the real and predicted traffic flows for one randomly chosen traffic pattern from Cluster 1, one from Cluster 15, and one from Cluster 30. As shown in Figures 9–11, the predicted traffic patterns are very close to real traffic patterns, proving the performance of the proposed method. Figures 9b, 10b, and 11b are the corresponding zoom-in plots of the prediction horizons of Figures 9a, 10a, and 11a. Note that the prediction accuracy deteriorates as the prediction horizon increases, but the deteriorating rate is not significant over time.

EVALUATION

Performance

The following mean absolute percentage error (MAPE) equation is used to evaluate the performance of the proposed method:

$$MAPE = \frac{1}{T \times J} \sum_{j=1}^J \sum_{t=1}^T \left| \frac{x_j(t) - \hat{x}_j(t)}{x_j(t)} \right|, \tag{11}$$

where $x_j(t)$ and $\hat{x}_j(t)$ are the real and predicted traffic flow at time step t at interchange j ; T is the total prediction time steps; and J is the total number of interchanges in the study corridor ($T = 48$ and $J = 15$ in the case study).

Our results show that the MAPE values of Clusters 1, 15, and 30 are 5.10%, 4.85%, and 5.03% in training and 10.15%, 7.18%, and 8.17% in validation, respectively. The highest MAPE values reach 6.11% in training and 17.64% in validation in Cluster 18. The absolute percentage error for a single time step prediction can reach as high as 28.75% in training and 36.73% in validation. Of the 36 clusters, the average training and validation MAPE values are 4.58% and 10.07%, respectively, suggesting a satisfactory prediction accuracy of the proposed method.

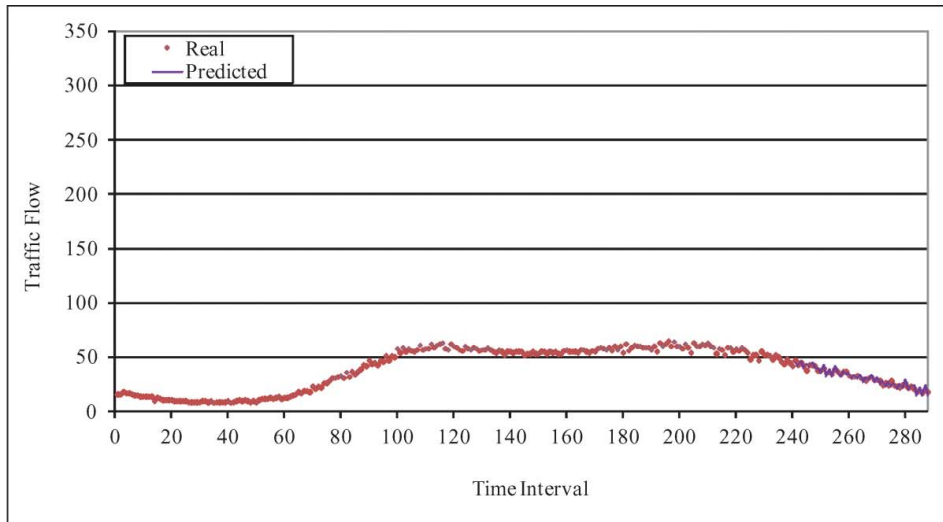
Comparison With Other Models

To demonstrate the superiority of the proposed method, a conventional and a seasonal autoregressive integrated moving average model, ARIMA(p,d,q) and SARIMA(p,d,q)(P,D,Q) s , are developed for each cluster. Similar to the proposed method, both ARIMA and SARIMA models have been calibrated and tested against the historical 240 time-step traffic flows and then used to predict the subsequent 48 time-step traffic flows. Taking Cluster 1 as an example, the best corresponding estimated models are ARIMA(2,1,1) and SARIMA(1,0,1)(0,1,1)₂₈₈. For the ARIMA model, the MAPE values are 21.77% in training and 28.65% in validation, respectively, while the MAPE values for SARIMA are 11.01% in training and 18.56% in validation, respectively. Obviously, the MAPE values for both ARIMA and SARIMA models are much higher than the counterparts of the proposed method (5.10% and 10.15%, respectively), indicating that the proposed method is superior to the ARIMA and SARIMA models.

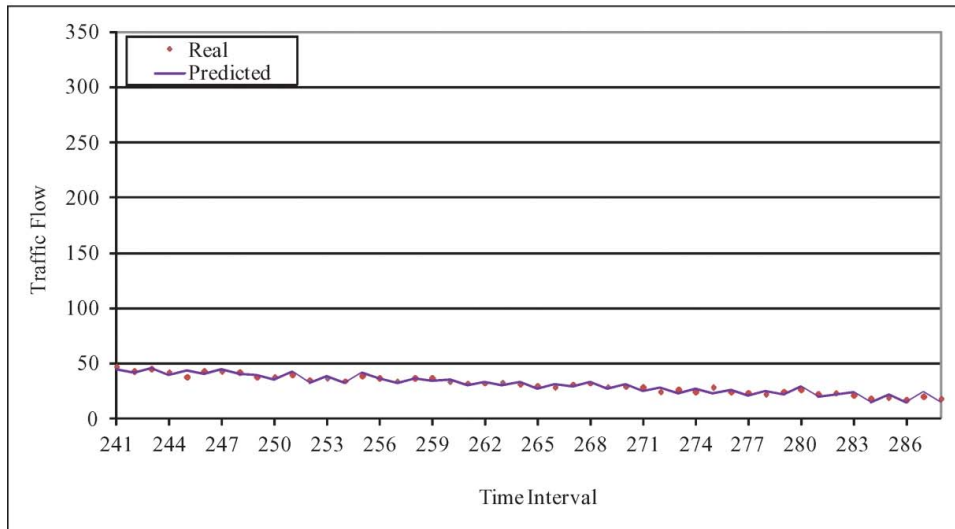
Furthermore, with the self-structured traffic patterns, a naive model is also developed by simply averaging the traffic flow data of the latest 48 time steps. For example, if one traffic pattern is of interest in Cluster 1 where 143 traffic patterns have been identified, then the traffic flows at the next 48 time steps can be predicted by averaging the traffic flows of 143 traffic patterns. Of the 36 clusters, the average MAPE values in training and in validation for the naive model are 25.21% and 32.73%, respectively, much higher than the counterparts of the proposed method. In sum, a comparison with the ARIMA, SARIMA, and naive model has confirmed the superior prediction accuracy of the proposed method.

Comparisons Among Various Lengths of Prediction

The preceding predictions utilized the 240-time-step (20-h) interval historical traffic data to partition the traffic patterns into appropriate number of clusters and then to predict the traffic flows in each cluster at the next 48 time steps (4 h). In



(a) A complete horizon (24 hours)



(b) A zoom-in plot of predicted horizon (4 hours)

Figure 11 Real and predicted traffic of a randomly selected traffic pattern from Cluster 30.

addition to predicting the 48 time steps ahead, this study further compares the prediction accuracy among various lengths of prediction with $L = 72$ (6 h), 96 (8 h), 120 (10 h), 144 (12 h), 192 (16 h), and 240 (20 h). The *MAPE* values in training and in validation are summarized in Table 3. The shorter

lengths (e.g., $L = 48$ and 72) have relatively lower prediction accuracy than the longer ones, suggesting the necessity of inputting a sufficient length of traffic data for both pattern recognition (training and validation) and prediction. However, no significant changes in prediction accuracy are found once the length of prediction exceeds 120 time steps in the case study.

Table 3 *MAPE* values associated with different lengths of prediction.

Lengths (time steps)	<i>MAPE</i> values in Training	<i>MAPE</i> values in Validation
48	7.29%	19.72%
72	7.78%	14.74%
96	5.32%	12.01%
120	5.72%	10.34%
144	5.91%	10.86%
192	5.63%	10.38%
240	4.58%	10.07%

CONCLUDING REMARKS

This study contributes to traffic literature by proposing a novel method to predict the relevant traffic features (5-min flow rates) for the entire 110-km freeway stretch with 48 time steps (4 h) ahead. The proposed method employs the GHSOM model to partition unlabeled traffic patterns into an appropriate number of clusters and then develops a GP model to predict

the traffic features in each cluster, based on the concept of rolling self-structured traffic patterns. The case study demonstrates that the proposed method has achieved relatively high prediction accuracy, superior to the ARIMA, SARIMA, and naive models. Further analyses suggest that applications of the proposed method require a sufficient length of observation to acquire enough traffic patterns for training and validation to achieve high prediction accuracy.

Further applications and modifications of the proposed method can be considered in the future studies. First, although this study has compared the proposed method with ARIMA, SARIMA, and naive models and proved that the proposed method is superior in prediction accuracy, it is interesting to further compare with other methods, such as the genetic clustering model (GCM), artificial neural network (ANN), and support vector machine (SVM). Second, if a sufficient length of traffic observation is acquired, the proposed method should be readily applicable to predict the traffic features in a larger freeway network with longer time steps ahead, which can be pragmatically useful in ATIS applications. Third, this study directly adopts the parameter settings of the GHSOM and GP models suggested by previous researchers. In the future, justifying the parameter/function settings can be attempted, which may further improve the prediction accuracy. Last but not least, incorporating the proposed method with dynamic traffic assignment and flow modeling can further predict the travel time information for different O-D pairs over a long freeway corridor, which can be very useful in ATIS applications. This calls for further exploration.

FUNDING

The authors express sincere thanks to three anonymous reviewers for their constructive comments and suggestions to improve the quality of the original article. This research was partially sponsored by the National Science Council, Republic of China (NSC 96-2628-E-009-171-MY3, NSC 92-2211-E-009-058, NSC-91-2211-E-009-048).

REFERENCES

- Brucker, P. (1978). On the complexity of clustering problems. In M. Beckmann & H. P. Kunzi (Eds.), *Optimization and operation research* (pp. 45–54). Heidelberg, Germany: Springer-Verlag.
- Campbell, J. L., Carney, C., & Kantowitz, B. H. (2003). *Human factors design guidelines for advanced traveler information systems (ATIS) and commercial vehicle operation (CVO)*. Washington, DC: FHWA-RD-98-057-2, Federal Highway Administration, U.S. Department of Transportation.
- Chang, F. J., Chang, L. C., & Huang, H. L. (2002). Real time recurrent neural network for streamflow forecasting. *Hydrological Processes*, **16**, 2577–2588.
- Chang, J. L., & Miaou, S. P. (1999). Real-time prediction of traffic flows using dynamic generalized linear models. *Transportation Research Record*, **1678**, 168–178.
- Chen, Y., Zhang, Y., & Hu, J. (2008). Multi-dimensional traffic flow time series analysis with self-organizing maps. *Tsinghua Science and Technology*, **13**, 220–228.
- Chen, H., & Grant-Muller, S. (2001). Use of sequential learning for short-term traffic flow forecasting. *Transportation Research Part C*, **9**, 319–336.
- Chen, H., Grant-Muller, S., Mussone, L., & Montgomery, F. (2001). A study of hybrid neural network approaches and the effects of missing data on traffic forecasting. *Neural Computing and Applications*, **10**, 277–286.
- Chiou, Y. C., & Lan, L. W. (2001). Genetic clustering algorithms. *European Journal of Operational Research*, **135**, 413–427.
- Chiou, Y. C., Chiou, Y. C., & Ai, C. M. (2007). Rolling grey forecasting models for short-term traffics. *Journal of the Eastern Asia Society for Transportation Studies*, **7**, 2486–2501.
- Chiou, Y. C., & Chou, S. T. (2010). Ant clustering algorithms. *International Journal of Applied Evolutionary Computation*, **1**, 1–15.
- Chiou, Y. C., Lan, L. W., & Tseng, C. M. (2010). Estimation of dynamic freeway origin-destination matrices with cell-based arrival distribution modeling. *Journal of the Eastern Asia Society for Transportation Studies*, **8**, 336–349.
- Dia, H. (2001). An object-oriented neural network approach to short-term traffic forecasting. *European Journal of Operational Research*, **131**, 253–261.
- Dimitriou, L., Tsekeris, T., & Stathopoulos, A. (2008). Adaptive hybrid fuzzy rule-based system approach for modeling and predicting urban traffic flow. *Transportation Research Part C*, **16**, 554–573.
- Dochy, T., Danech-Pajouh, M., and Lechevallier, Y. (1996). Short-term road traffic forecasting using neural network. *RTS English Issue*, 11.
- Dougherty, M., & Cobbett, M. (1997). Short-term inter-urban traffic forecasts using neural networks. *International Journal of Forecasting*, **13**, 21–31.
- Ghosh, B., Basu, B., & O'Mahony, M. M. (2005, January). Time-series modelling for forecasting vehicular traffic flow in Dublin. *Proceedings of TRB Annual Meeting*, Washington, DC, 1–22.
- Ghosh, B., Basu, B., & O'Mahony, M. M. (2009). Multivariate short-term traffic flow forecasting using time-series analysis. *IEEE Transactions on Intelligent Transportation Systems*, **10**, 246–254.
- Hong, W. C. (2011). Traffic flow forecasting by seasonal SVR with chaotic simulated annealing algorithm. *Neurocomputing*, **74**, 2096–2107.
- Huang, S., & Sadek, A. W. (2009). A novel forecasting approach inspired by human memory: The example of short-term traffic volume forecasting. *Transportation Research Part C*, **17**, 510–525.
- Kamarianakis, Y., & Prastakos, P., (2003). Forecasting traffic flow conditions in an urban network: Comparison of multivariate and univariate approaches. *Transportation Research Record*, **1857**, 74–84.
- Kirby, H., Dougherty, M., & Watson, S. (1997). Should we use neural networks or statistical models for short term motorway traffic forecasting? *International Journal of Forecasting*, **13**, 43–50.
- Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, **43**, 50–69.
- Koza, J. R. (1992). *Genetic programming: On the programming of computers by means of natural selection* (pp. 619–632). Cambridge, MA: MIT Press.
- Kumar, P., Singh, V., & Reddy, D. (2005). Advanced traveler information system for Hyderabad City. *IEEE Transactions on Intelligent Transportation Systems*, **6**, 26–37.

- Lam, W. H. K., Chan, K. S., Tam, M. L., & Shi, J. W. (2005). Short-term travel time forecasts for transport information system in Hong Kong. *Journal of Advanced Transportation*, **39**, 289–306.
- Lan, L. W., & Huang, Y. C. (2006). A rolling-trained fuzzy neural network approach for freeway incident detections. *Transportmetrics*, **2**, 11–29.
- Lan, L. W., Sheu, J. B., & Huang, Y. S. (2007). Prediction of short-interval traffic dynamics in multidimensional spaces. *Journal of the Eastern Asia Society for Transportation Studies*, **7**, 2353–2367.
- Lan, L. W., Sheu, J. B., & Huang, Y. S. (2008). Investigation of temporal freeway traffic patterns in reconstructed state spaces. *Transportation Research Part C*, **16**, 116–136.
- Lan, L. W., Lin, F. Y., & Kuo, A. Y. (2010). Three novel methods to predict traffic time series in reconstructed state spaces. *International Journal of Applied Evolutionary Computation*, **1**, 16–35.
- Ledoux, C. (1997). An urban traffic flow model integrating neural networks. *Transportation Research Part C*, **5**, 287–300.
- Maccubbin, R. P., Staples, B. L., & Mercer, M. R. (2003). *Intelligent transportation systems benefits and costs: 2003 update*. Washington, DC: FHWA-OP-03-075, Federal Highway Administration, U.S. Department of Transportation.
- Okutani, I., & Stephanedes, Y. J. (1984). Dynamic prediction of traffic volume through Kalman filtering theory. *Transportation Research Part B*, **18**, 1–11.
- Rauber, A., Merkl, D., & Dittenbach, M. (2002). The growing hierarchical self-organizing map: Exploratory analysis of high-dimensional data. *IEEE Transactions on Neural Networks*, **13**, 1331–1341.
- Sheu, J. B., Huang, Y. S., & Lan, L. W. (2009a). A real-time recurrent learning on predicting short-term traffic dynamics for sustainable management. *International Journal of Environment and Sustainable Development*, **8**, 330–350.
- Sheu, J. B., Lan, L. W., & Huang, Y. S. (2009b). Short-term prediction of traffic dynamics with real-time recurrent learning algorithms. *Transportmetrics*, **5**, 59–83.
- Smith, B. L., & Demetsky, M. J. (1997). Traffic flow forecasting: Comparison of modeling approaches. *Journal of Transportation Engineering*, **123**, 261–266.
- Smith, B. L., Williams, B. M., & Oswald, R. K. (2002). Comparison of parametric and nonparametric models for traffic flow forecasting. *Transportation Research Part C*, **10**, 303–321.
- Szeto, W. Y., Ghosh, B., Basu, B., & O'Mahony, M., (2009). Multivariate traffic forecasting technique using cell transmission model and SAR-IMA model. *Journal of Transportation Engineering*, **135**, 658–667.
- Tangsrapiroj, S., & Samadzadeh, M. H. (2006). Organizing and visualizing software repositories using the growing hierarchical self-organizing map. *Journal of Information Science and Engineering*, **22**, 283–295.
- Van Der Voort, M., Dougherty, M., & Watson, S. (1996). Combining Kohonen maps with ARIMA time-series models to forecast traffic flow. *Transportation Research Part C*, **4**, 307–318.
- Vlahogianni, E. I. (2009). Enhancing predictions in signalized arterials with information on short-term traffic flow dynamics. *Journal of Intelligent Transportation Systems*, **13**, 73–84.
- Vlahogianni, E. I., Karlaftis, M. G., & Golias, J. C. (2005). Optimized and meta-optimized neural networks for short-term traffic flow prediction: A genetic approach. *Transportation Research Part C*, **13**, 211–234.
- Vlahogianni, E. I., Karlaftis, M. G., & Golias, J. C. (2006). Statistical methods for detecting nonlinearity and non-stationarity in univariate short-term time-series of traffic volume. *Transportation Research Part C*, **14**, 351–367.
- Vlahogianni E. I., Karlaftis, M. G., & Golias, J. C. (2008). Temporal evolution of short-term urban traffic flow: A non-linear dynamics approach. *Computer-Aided Civil and Infrastructure Engineering*, **23**, 536–548.
- Welch, W. J. (1983). A mean squared error criterion for the design of experiment. *Biometrika*, **70**, 205–213.
- Williams, B. M., & Hoel, L. A. (2003). Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. *Journal of Transportation Engineering, ASCE*, **129**, 664–672.
- Yang, H. T., Chen, S. C., & Tsai, W. N. (2004). Classification of direct load control curves for performance evaluation. *IEEE Transactions on Power Systems*, **19**, 811–829.
- Yang, Y., Jiang, D., & Xia, M. (2010). Using improved GHSOM for intrusion detection. *Journal of Information Assurance and Security*, **5**, 232–239.
- Yao, L. & Lin, C. C. (2009). Identification of nonlinear systems by the genetic programming-based Volterra filter. *IET Signal Process*, **3**, 93–105.
- You, J. & Kim, T. J. (2000). Development and evaluation of a hybrid travel time forecasting model. *Transportation Research Part C*, **8**, 231–256.
- Zhang, Y. & Ye, Z. (2008). Short-term traffic flow forecasting using fuzzy logic system methods. *Journal of Intelligent Transportation Systems*, **12**, 102–112.