Neural-Network-Based Models for Short-Term Traffic Flow Forecasting Using a Hybrid Exponential Smoothing and Levenberg–Marquardt Algorithm

Kit Yan Chan, *Member, IEEE*, Tharam S. Dillon, *Life Fellow, IEEE*, Jaipal Singh, *Member, IEEE*, and Elizabeth Chang, *Senior Member, IEEE*

***Abstract*—This paper proposes a novel neural network (NN) training method that employs the hybrid exponential smoothing method and the Levenberg–Marquardt (LM) algorithm, which aims to improve the generalization capabilities of previously used methods for training NNs for short-term traffic flow forecasting. The approach uses exponential smoothing to preprocess traffic flow data by removing the lumpiness from collected traffic flow data, before employing a variant of the LM algorithm to train the NN weights of an NN model. This approach aids NN training, as the preprocessed traffic flow data are more smooth and continuous than the original unprocessed traffic flow data. The proposed method was evaluated by forecasting short-term traffic flow conditions on the Mitchell freeway in Western Australia. With regard to the generalization capabilities for short-term traffic flow forecasting, the NN models developed using the proposed approach outperform those that are developed based on the alternative tested algorithms, which are particularly designed either for short-term traffic flow forecasting or for enhancing generalization capabilities of NNs.**

***Index Terms*—Exponential smoothing method, Levenberg-Marquardt (LM) algorithm, neural networks (NNs), short-term traffic flow forecasting.**

I. Introduction

FORECASTING of road traffic flow conditions is essential for advanced traffic management information systems, which mainly aim to reduce traffic congestion and improve mobility of transportation. Short-term traffic flow forecasting, which has a horizon of only a few minutes, is highly suitable for traffic management information systems in supporting proactive dynamic traffic control to anticipate traffic congestion [3], [22], [56]. Short-term traffic flow forecasting models can be generated by conventional statistical methods such as filtering techniques [38], [35], autoregressive integrated moving average methods [42], and k-nearest-neighbor approaches [10]. Even if the models developed by such statistical methods can obtain reasonable prediction accuracy for future traffic flow conditions, they have two common limitations: 1) It is difficult to specify the most suitable model without human expertise. 2) The models generated by these methods may not be able to capture some strongly nonlinear characteristics of short-term traffic flow data. To address these limitations, neural network (NN) approaches have commonly been used for short-term traffic flow forecasting [7], [12], [14]–[16], [28]. However, the sole use of NN approaches may not achieve the best generalization capability for traffic flow forecasting, and usually, the methodologies for enhancing generalization capabilities are discussed within two classes.

1) *Hybrid NN approaches*, which incorporate other computational intelligence methods or statistical prediction methods that have been recently investigated for enhancing the generalization capabilities of NNs. A commonly used method for time series forecasting, i.e., Takagi-Sugeno fuzzy NNs [24], [25], [34], [36], which combine the mechanisms of fuzzy logic and feedforward NNs, has been proposed for short-term traffic flow forecasting [20], [39], [48]. Stathopoulos *et al.* [43] proposed a hybrid NN to forecast short-term traffic flow, which is developed by a fuzzy-rule-based system, which combines the forecasting outputs from an NN and a Kalman filter. Srinivasan *et al.* [41] proposed a hybrid NN, which consists of two components, i.e., a fuzzy filter and a feedforward NN. The fuzzy filter performs the clustering operation on traffic flow data and provides a rough prediction, which is the input of the feedforward NN. Accurate short-term traffic flow forecasting is produced by the feedforward NN, which utilizes each cluster as input for modeling the input–output relation. Tan *et al.* [44] proposed a hybrid NN, which combines the mechanism of an NN with classical forecasting methods, including moving average and autoregressive moving averages. The output forecasting results obtained by the classical forecasting methods are used as inputs of the NN, and the NN generates the final traffic flow forecast based on these inputs. While these hybrid NN approaches outperform the pure NN approach on short-term traffic flow forecasting, more NN parameters are required to be tuned or optimized on these hybrid NNs than those on the pure NNs. More computational power and memory space are required when implementing hybrid NNs than are required by the pure NNs. The memory footprints of hybrid NNs were found to be very large, which limits their potential applications. The hybrid NNs are therefore not suitable to be adaptively

Manuscript received January 21, 2011; revised May 26, 2011 and August 31, 2011; accepted October 22, 2011. Date of publication November 26, 2011; date of current version May 30, 2012. The Associate Editor for this paper was D. Srinivasan.

The authors are with the Digital Ecosystems and Business Intelligence Institute, Curtin University of Technology, Perth, WA 6102 Australia (e-mail: Kit.Chan@curtin.edu.au).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2011.2174051

tuned, compared with the pure NNs, as more expensive processors with more memory space and computational power are required for the hybrid NNs.

2) *Preventing overfitting in NN training* enhances generalization capabilities. This can be done by adding noise to the available training data to generate larger sets of training samples [26]. Generalization performance can be enhanced, but more computational time and effort are required due to the additional training data that are required to be fitted by the NN models. Another commonly used approach is cross validation [40], where the training data are divided into two data sets, i.e., the fitting data set and the validation data set. Only the fitting data set participates in NN learning, and the validation data set is used to compute validation error, which approximates the generalization error. Once the validation error increases, the training is terminated because the NN model may begin to fit the noise in the training data and overfitting may occur. Liu *et al.* [31] mentioned that applying proper cross validation is not a straightforward way to avoid overfitting. However, it is difficult to ensure that the validation data set is representative enough regarding the data distribution, so that the validation error can provide an unbiased estimate of the real generalization capability for short-term traffic flow forecasting [27].

In this paper, a simple but effective approach, i.e., the hybrid exponential smoothing and Levenberg–Marquardt (LM) algorithm (EXP-LM), is proposed to train NNs to produce high generalization capability in the short-term traffic flow forecasted. EXP-LM incorporates the mechanisms of the exponential smoothing method and the LM algorithm. Observing the characteristics of the traffic flow data indicates that its landscape is highly lumpy. As lumpiness is included in training, the training error can be decreased to a small value by fitting the lumpiness. However, having a small training error that is too small may degrade the generalization capability on the short-term traffic flow forecasting on unseen data. If the lumpiness of the original traffic flow data is removed, the generalization capability would be enhanced [54]. In EXP-LM, the exponential smoothing method [11], [47] is used to remove lumpiness in traffic flow data before applying the data to develop NN models. It is used because it is simple, and only relatively small extra computational effort is required [29]. A similar approach has been applied on electric short term load forecasting, in which better results can be achieved than those obtained by only using the original data [13]. After removing the lumpiness based on exponential smoothing, EXP-LM uses the LM algorithm to train NNs based on the exponential smoothed data. The resulting NNs are intended to fit the traffic flow characteristics where the lumpiness is removed. Comparisons were conducted based on the NNs generated by the EXP-LM and the other existing approaches to train NNs for traffic flow forecasting. The results show that NNs with better generalization capabilities in short-term traffic flow forecasting can be obtained by using the hybrid exponential smoothing and back-propagation (BP) algorithm, namely EXP-BP, compared with other tested methods, where the mechanisms of EXP-BP are identical to EXP-LM, except that the LM algorithm is replaced with the BP algorithm.

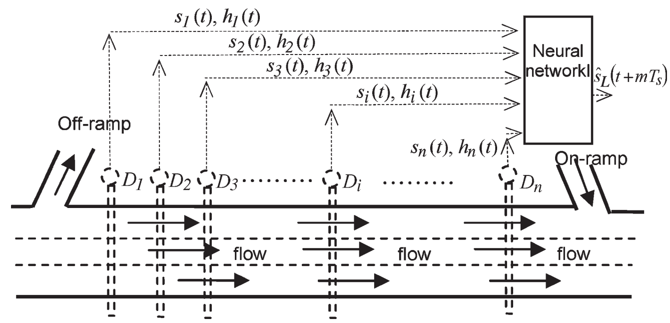


Fig. 1. Schematic of short-term traffic flow forecasting of the freeway.

The rest of this paper is organized as follows: Section II shows the configuration of the NN for short-term traffic flow forecasting. Section III discusses the mechanisms of EXP-LM. Section IV shows and discusses the results obtained by EXP-LM and other tested algorithms for forecasting short-term traffic flow conditions in different locations of the Mitchell freeway in Western Australia. Finally, a conclusion is given in Section V.

II. Neural Networks for Short-Term Traffic Flow Forecasting

The NN for short-term traffic flow forecasting was developed based on traffic flow data collected from n detector stations $(D_1, D_2, \ldots, D_n)$, which are located between the off-ramp and on-ramp of the freeway, as shown in Fig. 1. D_i captures two traffic flow measures, i.e., the average speed $s_i(t)$ of vehicles passing through and the average headway $h_i(t)$ between two consecutive vehicles passing through between time t and time $t + T_s$, where T_s is the sampling time. In general, if the average captured speeds of the vehicles are near the speed limit of the freeway and the average captured headway between vehicles is high, the traffic flow condition is considered to be smooth on the freeway.

Future short-term traffic flow can be forecasted by the NN, based on the current and past traffic flow. The current traffic flow at time t is indicated by the current average speed $s_i(t)$ and current average headway $h_i(t)$. The past traffic flow is indicated by the past average speed $s_i(t - k \cdot T_s)$ and past average headway $h_i(t - k \cdot T_s)$, which was collected by D_i at time $(t - k \cdot T_s)$ with $i = 1, 2, \ldots, n$ and $k = 1, 2, \ldots, p$, whereas the past traffic flow data within p sampling time interval/period are collected. The future short-term traffic flow, which is the output generated by the NN, is indicated by the predicted average speed of vehicles $\hat{s}_L(t + mT_s)$ passing through the Lth detector station D_L at time $(t + m \cdot T_s)$, where future traffic flow with m sampling time ahead is forecasted.

To predict the future traffic flow at the location of D_L, the following multilayer (three layers) NN is implemented, where satisfactory results can be obtained for traffic flow forecasting [7], [14], [16], [22]. The NN is formulated as follows:

$$\hat{s}_L(t+mT_s) = \sum_{i=1}^{n}\sum_{j=1}^{M}\left[\beta_{j,i}^{h}\Psi\left(\gamma_{0,j,i}^{h} + \sum_{k=1}^{p}\gamma_{k,j,i}^{h} h_i(t-kT_s)\right) + \beta_{j,i}^{s}\Psi\left(\gamma_{0,j,i}^{s} + \sum_{k=1}^{p}\gamma_{k,j,i}^{s} s_i(t-kT_s)\right)\right] + \alpha_0 \quad (1)$$

where M is the number of nodes in the hidden layer; α_0, $\beta^h_{j,i}$, $\beta^s_{j,i}$, $\gamma^h_{0,j,i}$, $\gamma^s_{0,j,i}$, $\gamma^s_{k,j,i}$, and $\gamma^h_{k,j,i}$ are the parameters of the NN, i.e., NN weights; and $\Psi(.)$ is the activation function of the hidden set in which sigmoid functions is a commonly used function. The NN weights can be determined based on the N_D collected traffic flow data, which are in the form of

$$d(l) = [\theta(l), \varphi(l)] \text{ with } l = 1, 2, \ldots, N_D \tag{2}$$

where N_D is the number of collected traffic flow data for training, and $\theta(l)$ is the lth future traffic flow data, which is the average speed of vehicles collected from the Lth detector station at time $(t(l) + mT_s)$; $\theta(l)$ is denoted by

$$\theta(l) = s_L\left(t(l) + mT_s\right) \tag{3}$$

and $\varphi(l)$ is the lth current and past traffic flow data, which is collected from the n detector stations and is denoted by

$$\begin{aligned}\varphi(l) = \Big[&h_1\left(t(l) - T_s\right), h_1\left(t(l) - 2T_s\right), \ldots, h_1\left(t(l) - pT_s\right)\\ &h_2\left(t(l) - T_s\right), h_2\left(t(l) - 2T_s\right), \ldots\\ &h_2\left(t(l) - pT_s\right), \ldots., h_n\left(t(l) - T_s\right)\\ &h_n\left(t(l) - 2T_s\right), \ldots, h_n\left(t(l) - pT_s\right), s_1\left(t(l) - T_s\right)\\ &s_1\left(t(l) - 2T_s\right), \ldots, s_1\left(t(l) - pT_s\right), s_2\left(t(l) - T_s\right)\\ &s_2\left(t(l) - 2T_s\right), \ldots, s_2\left(t(l) - pT_s\right), \ldots..\\ &s_n\left(t(l) + T_s\right), s_n\left(t(l) - 2T_s\right), \ldots, s_n\left(t(l) - pT_s\right)\Big].\end{aligned} \tag{4}$$

$h_i(t(l))$ and $s_i(t(l))$ are the average headway between cars and the average speed of cars collected by D_i, respectively, at time $t(l)$ with respect to the lth traffic flow data. Based on the collected traffic flow data $d(l) = [\theta(l), \varphi(l)]$ with $l = 1, 2, \ldots, N_D$, the NN can be evaluated based on the mean absolute relative error (e_{MARE}), which indicates the differences between the collected future traffic flow data and the predicted future traffic flow. e_{MARE} is formulated as

$$e_{\mathrm{MARE}} = \frac{1}{N_D} \frac{\sum_{l=1}^{N_D} \left|\theta(l) - \hat{\theta}(l)\right|}{\theta(l)} \tag{5}$$

where $\theta(l)$ is the lth collected future traffic flow data; $\hat{\theta}(l)$ is the prediction of future traffic flow, which is denoted by

$$\hat{\theta}(l) = \hat{s}_L\left(t(l) + mT_s\right) \tag{6}$$

and $\hat{s}_L(t(l) + mT_s)$ is determined based on (1) to forecast the average future traffic speed at the location of D_L.

Then, the LM algorithm is a commonly used method to train NNs by minimizing the mean absolute relative error e_{MARE} [59]. It starts by randomly generating the first two initial guesses of NN weights $w(0)$ and $w(1)$ at the 0th and the first iteration, where

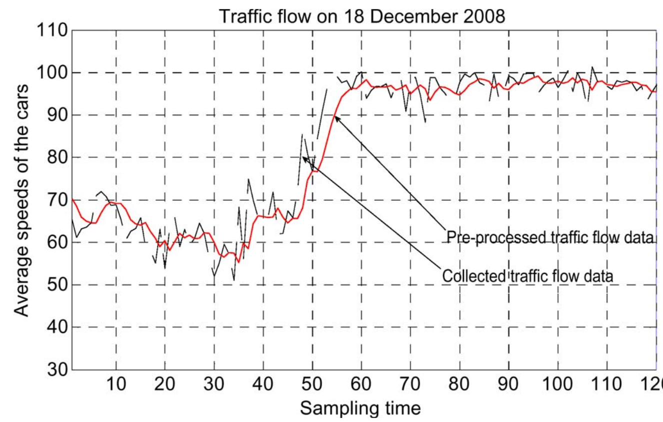


Fig. 2. Traffic flow data collected from 18 December 2008.

$$\begin{aligned}w(0) = \big[&\alpha_0(0), \beta^h_{j,i}(0), \beta^s_{j,i}(0), \gamma^h_{0,j,i}(0), \gamma^s_{0,j,i}(0),\\ &\gamma^h_{k,j,i}(0), \gamma^s_{k,j,i}(0)\big]\end{aligned} \tag{7}$$

$$\begin{aligned}w(1) = \big[&\alpha_0(1), \beta^h_{j,i}(1), \beta^s_{j,i}(1), \gamma^h_{0,j,i}(1), \gamma^s_{0,j,i}(1),\\ &\gamma^h_{k,j,i}(1), \gamma^s_{k,j,i}(1)\big]\end{aligned} \tag{8}$$

with $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, M$, and $k = 1, 2, \ldots, p$, respectively. Then, the LM algorithm updates the NN weights at the $(\varsigma + 1)$th iteration using the following formulation:

$$w(\varsigma+1) = w(\varsigma) + \left[J^T\left(w(\varsigma)\right) J\left(w(\varsigma)\right) + \mu I\right]^{-1} J^T\left(w(\varsigma)\right) R \tag{9}$$

where

$$R = \left[\,\theta(1) - \hat{\theta}(1) \quad \theta(2) - \hat{\theta}(2) \quad \ldots \quad \theta(N_D) - \hat{\theta}(N_D)\,\right]^T. \tag{10}$$

The details of determination of the Jacobian matrix $J(w(\varsigma))$ can be referred to [59]. To forecast average headway, similar formulation can be used by replacing $\hat{s}_L(t(l) + mT_s)$ with the forecast average headway, $\hat{h}_L(t(l) + mT_s)$, in (1). In addition, (3) is redefined by $\theta(l) = h_L(t(l) + mT_s)$.

III. Exponential Smoothing and Levenberg–Marquardt Algorithm

When the NN is being trained by the LM algorithm, the goodness-of-fit of the NN increases, and at the same time, e_{MARE} decreases. When e_{MARE} is equal to zero, the NN can fit all the collected traffic flow data, and also, all the characteristics of the collected traffic flow data are included. For example, Fig. 2 shows the traffic flow data regarding the average speeds of vehicles. The traffic flow data were collected from Mitchell Freeway, which is near the on-ramp of Reid Highway, Western Australia. It was collected over the 2-h peak traffic periods (7.30–9.30 am) on December 18, 2008, where the sampling time was 1 min. An NN can be obtained by fitting all the collected traffic flow data that have lumpy characteristics. However, these lumpy characteristics may not be helpful for forecasting future short-term traffic flow. The inclusions of these lumpy characteristics may overfit the NN, which can achieve a small e_{MARE} with respect to the collected traffic flow data used for training purposes but cannot achieve good generalization capability for unseen data.

To avoid overtraining, it is essential that this lumpy characteristic be filtered from the collected traffic flow data before implementing the LM algorithm to train the NNs. In this research, a new algorithm called EXP-LM, which incorporates the mechanisms of the exponential smoothing method and the LM algorithm, is proposed to train NNs for traffic flow forecasting. The EXP-LM first uses the mechanism of the exponential smoothing method, which is a simple and intensively used method for preprocessing time series data [4], by filtering out the lumpiness. It then uses the mechanism of the LM algorithm to train the NNs based on the preprocessed data, which is denoted by $d'(l) = [\theta'(l), \varphi(l)]$, with $l = 1, 2, \ldots, N_D$.

In EXP-LM, the lth filtered traffic flow data $\theta'(l)$ are modified by the $(l-1)$th filtered traffic flow data $\theta'(l-1)$ and the $(l-1)$th collected traffic flow data $\theta(l-1)$, based on a proportion of the error $(\theta(l-1) - \theta'(l-1))$, which is expected at the lth traffic flow data. The lth filtered traffic flow data $\theta'(l)$ with $l \geq 3$ is defined by the following equation:

$$\theta'(l) = \theta'(l-1) + \alpha\left(\theta(l-1) - \theta'(l-1)\right) \qquad (11)$$

where α is the smoothing constant within the range $0.1 < \alpha \leq 0.9$. The first and the second filtered traffic flow data are initialized by $\theta'(1) = \theta(1)$ and $\theta'(2) = (\theta(1) + \theta(2) + \theta(3))/3$, respectively.

If the value of the exponential smoothing parameter α is larger, then the change in the filtered traffic flow data $\theta'(l)$ is more rapid, and the more lumpy characteristics in the traffic flow data can be retained. If the value of α is smaller, then the change in the filtered traffic flow data $\theta'(l)$ is slower, and the more lumpy characteristics in the traffic flow data can be filtered out. To estimate the best exponential smoothing parameter α, which can be used by EXP-LM for filtering lumpiness in traffic flow data, a grid search with increments of $(0.8/N_G)$ of the parameter space between $\alpha = 0.1$ and $\alpha = 0.9$ is used, where N_G is the number of grids of the grid search. The higher the N_G, the smaller the $R^2(\alpha)$ that can be obtained. The best α is chosen to produce the smallest sum of squares for the residuals, which is defined as

$$R^2(\alpha) = \sum_{i_2=1}^{N_D} \left(\theta'(i_2)|_\alpha - \theta(i_2)\right)^2. \qquad (12)$$

The filtered traffic flow data, which were preprocessed by the exponential smoothing method, are shown in Fig. 2. It shows that the filtered traffic data seek to filter out the lumpiness due to irregular variation on the collected traffic flow data. Lumpiness may downgrade the generalization capability of the NN. If the filtered traffic flow data, which exclude the lumpiness, are used for training the NN, better generalization capability is more likely to be developed by EXP-LM. The mechanism of EXP-LM is illustrated by four steps.

Step 1 Collect the traffic flow data $d(l) = [\theta(l), \varphi(l)]$ in the form of (2), with $l = 1, 2, \ldots, N_D$.

Step 2 Select the best exponential smoothing parameter α by using the grid search where $0.1 < \alpha \leq 0.9$.

Step 2.1 Initialize N_G exponential smoothing parameters as

$$\alpha(i_2) = \left(\frac{0.8}{N_G}\right)(i_2 - 1) + 0.1$$

with $i_2 = 1, 2, \ldots, N_G$, where N_G is a constant.

Step 2.2 Evaluate the sum of squares for the residuals $R^2(\alpha(i_2))$, for all $\alpha(i_2)$, with $i_2 = 1, 2, \ldots, N_G$, based on (12).

Step 2.3 Determine the best exponential smoothing parameter, $\alpha_{\text{best}} = \alpha(i_2)$, where $R^2(\alpha(i_2)) < R^2(\alpha(i_3))$, with $\forall i_2, i_3$, but $i_2 \neq i_3$.

Step 3 Generate the filtered traffic flow data $d'(l) = [\theta'(l), \varphi(l)]$ based on the exponential smoothing method, in which α_{best} is used.

Step 3.1 Initialize the first and second filtered data $\theta'(1)$ and $\theta'(2)$, respectively.

Step 3.2 Generate the lth filtered traffic flow data based on (11), where $l \geq 3$.

Step 4 The NN is developed based on the filtered traffic flow data $d'(l) = [\theta'(l), \varphi(l)]$ using the LM algorithm.

Step 4.1 Initialize the first and second sets of NN weights $w(0)$ and $w(1)$ by (7) and (8), respectively.

Step 4.2 Update the NN weights $w(\varsigma + 1)$, based on (9), where e_{MARE} and $\hat{\theta}(l)|_{w(\varsigma)} = \hat{s}_L(t(l) + mT_s)|_w$ are determined by (5) and (1), respectively.

Step 4.3 Goto Step 4.2 until the termination iteration is reached or e_{MARE} reaches a satisfactory value.

IV. Experimental Results

In this section, the effectiveness of the EXP-LM method for training NN models for short-term traffic flow forecasting is evaluated based on traffic flow data collected from a freeway in Western Australia. First, comparisons between the EXP-LM and the other LM algorithms, which involve mechanisms for avoiding overfitting, are undertaken. Then, results based on the EXP-LM, which integrates with other advanced LM algorithms, are presented. Finally, the results of further evaluations are given to further demonstrate the effectiveness of the LM algorithms.

A. Traffic Flow Data

The NNs were developed using 12 traffic flow data sets, which are illustrated in Table I, where the dates and locations of traffic flow data taken are shown. The traffic flow data sets were collected from weeks 38, 41, and 52 in 2008, and weeks 2, 12, and 27 in 2009. Six of the traffic flow data sets (Reid-2008-38, Reid-2008-41, Reid-2008-52, Reid-2009-02, Reid-2009-12, and Reid-2009-27) were collected from the Reid Highway and Mitchell Freeway intersection, Western Australia, where the two detector stations were installed to collect the data. These two detector stations were located near the on-ramp and off-ramp of Reid Highway, respectively. The other six traffic flow data sets (Erindale-2008-38, Erindale-2008-41, Erindale-2008-52, Erindale-2009-02, Erindale-2009-12, and Erindale-2009-27) were collected from the Erindale Street and Mitchell

TABLE I
DESCRIPTION OF THE 12 COLLECTED TRAFFIC FLOW DATA SETS

Data collection dates	Data collected from the intersection of Reid Highway	Data collected from the intersection of Erindale Road
Week 38 in 2008 (15 Sep. 2008 – 19 Sep. 2008)	Reid-2008-38	Erindale-2008-38
Week 41 in 2008 (6 Oct. 2008 – 10 Oct. 2008)	Reid-2008-41	Erindale-2008-41
Week 52 in 2008 (22 Dec. 2008 – 24 Dec. 2008)	Reid-2008-52	Erindale-2008-52
Week 02 in 2009 (5 Jan. 2008 – 9 Jan. 2009)	Reid-2009-02	Erindale-2009-02
Week 12 in 2009 (16 Mar. 2008 – 20 Mar. 2009)	Reid-2009-12	Erindale-2009-12
Week 27 in 2009 (29 Jun. 2008 – 3 Jul. 2009)	Reid-2009-27	Erindale-2009-27

Freeway intersection, Western Australia, where the three detector stations were installed to collect data. These three detector stations were located near the off-ramp of Erindale Road, between the off-ramp and the on-ramp of Erindale Road, and near the on-ramp of Erindale Road, respectively.

The traffic flow data sets were collected over the 2-h peak traffic period (7.30–9.30 am) on the five business days of the week, i.e., Monday, Tuesday, Wednesday, Thursday, and Friday. Sixty seconds (1 min) of sampling time was used, and a total of 600 observations were included in each set of traffic flow data. Each traffic flow data set was divided into two subsets. The first subset of traffic flow data, i.e., the training data, collected from Monday to Thursday (comprising 80% of all the observations), was used for training the NNs. The second subset of traffic flow data, i.e., test data, collected from Friday (comprising 20% of all the observations), was used to evaluate the generalization capability of the trained NNs.

B. Experimental Results

The EXP-LM was implemented in Matlab. Four NNs (i.e., NN_2^{Reid}, NN_6^{Reid}, $NN_2^{Erindale}$, and $NN_6^{Erindale}$) were developed to forecast short-term traffic flow regarding Reid Highway and Erindale Road. For Reid Highway, NN_2^{Reid} and NN_6^{Reid} were developed to forecast traffic flow condition near the on-ramp of Reid Highway with two and six sampling periods ahead of time, respectively. For Erindale Road, $NN_2^{Erindale}$ and $NN_6^{Erindale}$ were developed to forecast traffic flow between the on-ramp and off-ramp of Erindale Road with two and six sampling periods ahead of time, respectively. They all used the last six sampling periods of the past traffic flow conditions to forecast the future traffic flow conditions.

Comparison Within LM Methods: To evaluate the effectiveness of EXP-LM, the following algorithms have been applied, and the results have been compared with those obtained by EXP-LM.

1) *Standard LM algorithm (S-LM)*, which is identical to EXP-LM, except that no filtering method is involved. The results obtained by S-LM can be used to compare the effect of using the exponential smoothing method, as the only difference between S-LM and EXP-LM is that EXP-LM involves exponential smoothing method to preprocess data but S-LM does not.
2) *Hybrid simple moving average and LM algorithm (SM-LM)* uses a simple moving average method as a smoothing method to filter the lumpiness in traffic flow data before using the LM algorithm to train the NNs. In the SM-LM, the lth filtered traffic flow data $\theta'(l)$ are generated based on the past four traffic flow data as follows:

$$\theta'(l) = \frac{1}{4}\left(\theta(l-1) + \theta(l-2) + \theta(l-3) + \theta(l-4)\right) \quad (13)$$

with $l > 4$, where $\theta'(1) = \theta(1)$, $\theta'(2) = \theta(2)$, $\theta'(3) = \theta(3)$, and $\theta'(4) = \theta(4)$.
3) *Hybrid weighted moving and LM algorithm (WM-LM)* uses the weighted moving method to filter lumpiness in the traffic flow data. In the WM-LM, the lth filtered traffic flow data $\theta'(l)$ is generated based on the past four traffic flow data as follows:

$$\theta'(l) = \frac{(4\cdot\theta(l-1) + 3\cdot\theta(l-2) + 2\cdot\theta(l-3) + \theta(l-4))}{10} \quad (14)$$

with $l > 4$, where $\theta'(1) = \theta(1)$, $\theta'(2) = \theta(2)$, $\theta'(3) = \theta(3)$, and $\theta'(4) = \theta(4)$.

The results obtained by SM-LM, WM-LM, and EXP-LM can be used to compare different smoothing methods used on the algorithms for training NNs.
4) *Cross-validation based LM algorithm [LM-CROSS-(τ)]* uses the mechanisms of cross-validation [2], [37] to avoid overtraining NNs. In LM-CROSS-(τ), the fitting data (comprising 60% of all the observations) were used for computing the NN weights, whereas the cross-validation data (comprising 20% of all the observations) were used to prevent overfitting when training the NNs. The error for the cross-validation data is monitored during the training process. It normally decreases during the initial phase of training, as does the error for the training data. When the NN begins to overfit the training data, the error for the cross-validation data begins to increase. LM-CROSS-(τ) stops training the NNs when the error for the cross-validation data at the $(\varsigma + \tau)$th iteration is higher than those at the ςth iteration. LM-CROSS-(5) and LM-CROSS-(10) were implemented. As LM-CROSS-(τ) is a commonly used method for avoiding overfitting, the results obtained by LM-CROSS-(τ) is significant to compare with the results obtained by EXP-LM.

The following parameters have been used in the five algorithms: the number of hidden nodes used in the NNs is $\log_2(480) \approx 9$, in which the number of training data N_D is 480 and $\log_2(N_D)$ is the recommended number of hidden nodes suggested in other works, such as [46]; the termination iteration is 100; termination occurs in EXP-LM, SM-LM, and WM-LM, when the termination iteration is reached or e_{MARE} is less than 0.01; and termination occurs in LM-CROSS-(5) and LM-CROSS-(10), when the error for the validation data increases.

TABLE II
TRAINING ERROR OBTAINED FOR REID HIGHWAY BASED ON EXP-LM, SM-LM, WM-LM, S-LM, LM-CROSS-(5), AND LM-CROSS-(10)

			EXP-LM	SM-LM	WM-LM	S-LM	LM-CROSS -(5)	LM-CROSS -(10)
NN_6^{Reid}	Reid-2008 -38	mean	8.73	10.36	8.60	7.34	5.46	6.43
		var	1.97	0.69	0.96	3.23	2.58	4.21
	Reid-2008 -41	mean	7.93	4.91	5.47	3.27	3.01	5.28
		var	0.797	1.64	1.29	4.40	2.52	4.34
	Reid-2008 -52	mean	5.03	6.06	6.19	3.71	3.74	3.87
		Var	0.31	0.53	0.64	0.32	1.25	0.59
	Reid-2009 -02	mean	3.72	5.08	3.85	3.69	3.42	3.99
		Var	0.28	0.46	0.39	3.69	0.93	0.77
	Reid-2009 -12	mean	11.03	10.23	9.96	8.27	6.10	7.05
		Var	0.33	0.82	0.65	3.14	4.33	1.83
	Reid-2009 -27	mean	8.23	8.42	7.70	7.51	6.97	6.48
		Var	1.22	0.84	1.86	1.48	2.83	1.80
$NN_6^{Erindale}$	Erindale-2008-38	mean	10.38	10.44	9.30	6.60	8.32	8.21
		Var	1.26	0.64	1.09	3.65	2.30	4.21
	Erindale-2008-41	mean	5.55	6.83	5.19	4.79	4.04	5.48
		Var	0.66	1.18	1.31	4.59	1.66	4.18
	Erindale-2008-52	mean	5.27	6.09	5.17	3.76	4.21	3.41
		Var	0.28	0.66	0.64	0.54	1.72	0.57
	Erindale-2009-02	mean	4.93	4.39	4.26	4.30	4.01	3.93
		Var	0.28	0.29	0.54	2.84	1.23	0.83
	Erindale-2009-12	mean	9.01	7.66	11.97	6.83	7.06	8.26
		Var	0.29	0.76	0.80	2.33	5.68	3.20
	Erindale-2009-27	mean	12.51	9.48	7.44	8.86	8.03	6.64
		Var	1.22	1.00	1.30	1.66	3.48	2.91
Average mean			6.22	6.57	6.56	4.76	4.97	4.77
Average variance			0.19	0.34	0.34	1.01	2.28	2.04
Average rank			4	6	5	1	3	2

All these algorithms were run for 30 times with different initial guesses of NN weights, and the results for the 30 runs were recorded. Table II shows the mean training error and variance of training errors among the 30 runs of the algorithms on computing the NN weight of NN_6^{Reid} and NN_6^{Erindale} regarding all data sets. The ranks of mean training errors among the algorithms are also shown. The results show that the averages of mean training errors obtained by S-LM, LM-CROSS-(2), and LM-CROSS-(5), which do not involve filtering of lumpiness in traffic flow data, are smaller than those obtained by the EXP-LM, SM-LM, and WM-LM, which do involve filtering of lumpiness in traffic flow data. In other words, the poorer fitting capability for the collected traffic flow data was obtained by the EXP-LM, SM-LM, and WM-LM.

To evaluate the generalization capability of the NNs developed based on the algorithms, the test data, which were not involved on training the NNs, were used. The mean test errors and variances of test errors among the 30 runs were recorded in Table III. In addition, the results obtained are in bold font, when they achieve the smallest test errors among the others. The average mean test errors, average variance of test errors, average ranks, and number of first ranks obtained by the algorithms are shown in the last four rows of the table. It can be found from Table III that the NNs trained by EXP-LM yield the smallest average of mean test errors and the best average rank, compared with those obtained by the other five algorithms. EXP-LM can achieve nine first ranks out of 12 tests, whereas SM-LM can achieve only two first ranks out of 12 tests and WM-LM can achieve only one first rank out of 12 tests. S-LM, LM-CROSS-(5) and LM-CROSS-(10) cannot achieve any first rank for the 12 tests. Therefore, these results indicate that, in general, EXP-LM can find the NNs with the best generalization capability when compared with those obtained by the other tested algorithms (S-LM, LM-CROSS-(5), LM-CROSS-(5), SM-LM, and WM-LM). In addition, the average variances of test errors obtained by EXP-LM are the

TABLE III
TEST ERROR OBTAINED FOR REID HIGHWAY AND ERINDALE HIGHWAY BASED ON EXP-LM, SM-LM, WM-LM, S-LM, LM-CROSS-(5), AND LM-CROSS-(10)

			EXP-LM	SM-LM	WM-LM	S-LM	LM-CROSS -(5)	LM-CROSS -(10)
NN_6^{Reid}	Reid-2008 -38	mean	10.93	11.54	**9.90**	14.86	10.01	12.35
		Var	53.00	42.28	**37.17**	46.34	19.80	36.22
	Reid-2008 -41	mean	**9.34**	12.08	10.81	16.24	10.73	15.44
		Var	**4.28**	8.85	7.63	14.02	10.42	8.36
	Reid-2008 -52	mean	**4.82**	5.14	6.99	9.90	6.11	8.11
		Var	**9.90**	11.09	7.85	76.10	4.32	91.41
	Reid-2009 -02	mean	2.31	**2.09**	2.44	4.61	3.15	3.26
		Var	0.07	**0.30**	0.25	1.49	1.33	1.54
	Reid-2009 -12	mean	**12.80**	15.31	23.57	16.36	12.95	17.07
		Var	**57.11**	235.3	170.8	34.28	44.09	67.58
	Reid-2009 -27	mean	8.52	**7.94**	8.10	10.86	11.88	12.08
		Var	1.66	**0.51**	1.84	6.52	3.44	3.76
$NN_6^{Erindale}$	Erindale-2008-38	mean	**4.21**	4.21	4.16	5.12	5.54	4.53
		Var	**2.18**	2.12	3.53	2.12	1.42	1.59
	Erindale-2008-41	mean	**7.48**	7.99	8.97	7.49	7.53	7.41
		Var	**1.01**	1.25	1.51	1.86	1.24	2.26
	Erindale-2008-52	mean	**5.84**	8.19	11.90	7.92	6.56	6.24
		Var	**1.31**	5.67	5.03	11.25	3.86	4.00
	Erindale-2009-02	mean	**6.75**	7.44	6.62	6.93	7.06	6.04
		Var	**1.91**	3.04	2.89	1.13	1.12	2.54
	Erindale-2009-12	mean	**9.04**	13.83	13.16	14.09	11.90	11.07
		Var	**14.99**	19.10	25.91	18.72	8.08	28.76
	Erindale-2009-27	mean	**3.93**	3.94	3.15	6.03	6.25	5.26
		Var	**0.43**	2.23	2.04	1.69	1.84	1.50
Average mean			6.25	6.96	7.50	7.97	7.43	7.95
Average variance			8.28	12.60	15.81	12.96	16.39	13.56
Average rank			1	2	4	6	3	5
Number of first ranks			9	2	1	0	0	0

The results obtained are in bold font, when they achieve the smallest test errors among the others

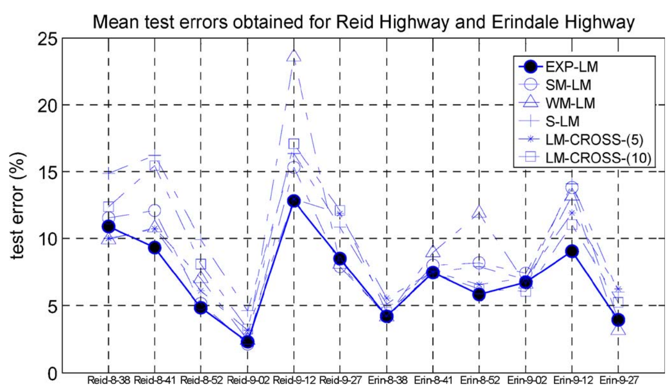


Fig. 3. Mean test errors obtained for Reid Highway and Erindale Highway using EXP-LM, SM-LM, WM-LM, S-LM, LM-CROSS(5), and LM-CROSS(10).

lowest among all algorithms. These results show that EXP-LM can produce a more stable generalization capability than the other algorithms. In addition, Fig. 3 shows the mean test errors obtained by all algorithms. It shows, in general, that the mean test errors obtained by EXP-LM are the smallest.

In addition, the t-test [55] was used to evaluate the significance of the hypothesis that the sample mean of the test errors of the NNs trained by EXP-LM are smaller than those trained by the other algorithms [S-LM, LM-CROSS-(5), LM-CROSS-(5), SM-LM, or WM-LM]. The t-values between EXP-LM and the other algorithms are shown in Table IV. Based on the t-distribution table, if the t-value is higher than 1.699, the significance is 95% confidence level, which means that the test errors of the NNs trained by EXP-LM are smaller than those trained by the other algorithm with 95% confidence level. The t-value can be determined by

$$t - \text{value} = \frac{\mu_2 - \mu_1}{\sqrt{\sigma_2^2/N_2 + \sigma_1^2/N_1}}$$

TABLE IV
T-VALUES BETWEEN EXP-LM TO THE OTHER ALGORITHMS [SM-LM, WM-LM, S-LM, LM-CROSS-(5), AND LM-CROSS-(10)] FOR REID HIGHWAY AND ERINDALE HIGHWAY

		SM-LM	WM-LM	S-LM	LM-CROSS-(5)	LM-CROSS-(10)
NN_6^{Reid}	Reid-2008-38	0.34	**2.97**	**2.16**	0.59	0.82
	Reid-2008-41	**4.14**	**6.39**	**8.84**	**1.99**	**9.40**
	Reid-2008-52	0.38	**1.74**	**3.00**	**1.88**	**1.79**
	Reid-2009-02	**1.91**	**9.03**	**10.10**	**3.92**	**4.13**
	Reid-2009-12	0.81	**2.76**	**2.04**	0.085	**2.10**
	Reid-2009-27	**2.12**	**5.23**	**4.49**	**8.17**	**8.39**
NN_6^{Erindale}	Erindale-2008-38	0.00	0.11	**2.40**	**3.83**	0.90
	Erindale-2008-41	**1.86**	**5.16**	0.04	0.19	0.20
	Erindale-2008-52	**4.88**	**13.20**	**3.22**	**1.73**	0.95
	Erindale-2009-02	**1.71**	0.32	**5.72**	0.97	**1.85**
	Erindale-2009-12	**4.49**	**3.53**	0.99	**3.26**	**1.68**
	Erindale-2009-27	0.03	**2.72**	**7.89**	**8.43**	**5.24**
Number of data sets that EXP-LM can obtain significantly better results		7	10	10	8	7

Bolded t-values indicate that significance is 95% confidence level.

where μ_1 is the mean test error of the NNs trained by the EXP-LM, and μ_2 is the one for the other compared algorithm; σ_1^2 is the variance of test errors of the NNs trained by the EXP-LM and σ_2^2 is the one for the other compared algorithm; and N_1 and N_2 are the number of tests performed by EXP-LM and the other compared algorithm, respectively. The t-values are in bold fonts when the confidence level is above 95%. In the last row of the table, the number of data sets that the EXP-LM performs significantly better than the other algorithms is shown. For example, comparing EXP-LM and S-LM, EXP-LM is significantly better than S-LM on ten data sets (NN_6^{Reid} for Reid-2008-38, Reid-2008-41, Reid-2008-51, Reid-2009-02, Reid-2009-12, and Reid-2009-27, as well as NN_6^{Erindale} for Erindale-2008-38, Erindale-2008-52, Erindale-2009-02, and Erindale-2009-27). Although EXP-LM cannot significantly outperform on all data sets, EXP-LM, in general, offered much better performance than did all the other algorithms.

These results show that the average of mean training errors obtained by EXP-LM are larger than those obtained by the other five algorithms on training NN_6^{Reid} and NN_6^{Erindale}, but the average of mean test errors and average of variances of mean test errors obtained by EXP-LM are smaller than those obtained by the other five algorithms. In other words, the better generalization capability of the six algorithms can generally be achieved by EXP-LM, whereas the fitting capability for the collected traffic flow data obtained by EXP-LM is generally poor in training the NNs for forecasting traffic flow. The results show that, in general, NNs with better generalization capability can be developed by using the exponential smoothing method to remove lumpiness from traffic flow data.

Incorporation With Advanced NN Configurations: To further evaluate the effectiveness of using exponential smoothing method in preprocessing the collected traffic flow data, the approach is incorporated with other advanced NN configurations. It aims to further evaluate whether the exponential smoothing method can help to improve the effectiveness with more advanced NN configurations for short-term traffic flow forecasting. The following two advanced NN configurations have been considered:

1) *A wavelet NN (WNN-LM)* [52], which combines the mechanisms of feedforward NNs and the wavelet theory [9], [33]. In WNN-LM, a wavelet function is used as the transfer function in the NNs, which provides a multiresolution approximation for the discriminate functions. Based on WNN-LM, better performance in function learning [51], including short-term traffic flow forecasting [23], can be obtained than those of the conventional feedforward NNs.
2) *A Bayesian NN (BNN-LM)* [32], which combines the mechanisms of Bayesian regularization and the LM method [18]. Based on the Bayesian regularization, the insignificant hidden nodes in the NN are removed to avoid developing an overtrained NN. Results show that generalization abilities of NNs are better than those obtained by the standard LM algorithm for time series forecasting [30], including short-term traffic flow forecasting [53].

TABLE V
TRAINING ERROR FOR EXP-WNN, WNN, EXP-BNN, AND BNN

			EXP-WNN	WNN	EXP-BNN	BNN
NN_6^{Reid}	Reid-2008 -38	mean	14.12	14.67	13.73	13.48
		var	0.65	20.79	1.11	0.93
	Reid-2008 -41	mean	8.94	8.19	7.66	6.90
		var	1.19	3.95	3.98	0.28
	Reid-2008 -52	mean	5.16	4.78	4.12	3.93
		var	1.05	2.41	0.15	0.14
	Reid-2009 -02	mean	1.44	1.56	1.81	1.15
		var	0.03	0.20	0.01	0.00
	Reid-2009 -12	mean	10.18	10.57	9.62	9.37
		var	1.42	3.34	1.09	1.04
	Reid-2009 -27	mean	17.93	16.01	13.61	13.17
		var	6.31	25.68	0.17	1.45
NN_6^{Erindale}	Erindale-2008-38	mean	14.12	14.67	13.73	13.48
		var	0.65	20.79	0.11	0.93
	Erindale-2008-41	mean	10.01	6.55	9.13	8.58
		var	0.34	1.18	3.95	0.14
	Erindale-2008-52	mean	6.89	4.21	5.44	4.59
		var	0.39	0.88	0.26	0.45
	Erindale-2009-02	mean	6.65	4.11	5.00	4.50
		var	0.32	0.87	0.02	0.03
	Erindale-2009-12	mean	11.38	7.95	9.37	8.27
		var	0.40	1.77	0.11	0.86
	Erindale-2009-27	mean	12.87	8.89	10.93	13.67
		var	1.42	2.29	0.59	55.20
Average mean			8.36	6.48	7.35	7.13
Average variance			1.60	2.58	0.49	1.41
Average rank			4	1	3	2

Based on the configurations of BNN-LM and WNN-LM, the original collected traffic flow data were used for training the NNs for short-term traffic flow forecasting. The results for NN_6^{Reid} and $NN_6^{Erindale}$ regarding training errors and test errors are shown in Tables V and VI, respectively. The results of the NNs trained based on the mechanisms of BNN-LM and WNN-LM are labeled as BNN-LM and WNN-LM, respectively. Other NNs were trained using the same mechanisms of BNN-LM and WNN-LM, but the data used for developing the NNs were based on the preprocessed traffic flow data in which the exponential smoothing method was used to remove the lumpiness in the original traffic flow data. These results are labeled as EXP-BNN-LM and EXP-WNN-LM, respectively. The parameters used in EXP-WNN-LM, WNN-LM, EXP-BNN-LM, and BNN-LM are the same as those used in the EXP-LM, as shown in Section IV-B1.

To illustrate the capability of fitting the traffic flow data, Table V shows that the average of mean training errors and average of ranks of mean training errors obtained by

TABLE VI
TEST ERROR OBTAINED FOR EXP-WNN, WNN, EXP-BNN, AND BNN

			EXP-WNN	WNN	EXP-BNN	BNN
$NN_6^{\text{Re}id}$	Reid-2008-38	mean	**6.11**	6.26	**3.54**	4.09
		var	**7.16**	8.39	**2.16**	0.66
	Reid-2008-41	mean	**9.90**	10.56	**6.63**	6.86
		var	**1.36**	2.39	**0.06**	0.12
	Reid-2008-52	mean	**2.82**	3.18	**1.81**	1.79
		var	**0.44**	0.80	**0.28**	0.03
	Reid-2009-02	mean	**2.43**	2.94	**1.71**	1.69
		var	**0.11**	0.20	**0.00**	0.00
	Reid-2009-12	mean	**4.94**	7.04	**2.60**	3.47
		var	**2.76**	3.23	**0.10**	0.42
	Reid-2009-27	mean	**11.77**	12.92	**6.80**	7.29
		var	**15.32**	32.20	**0.09**	1.03
NN_6^{Erindale}	Erindale-2008-38	mean	**5.14**	9.95	**3.37**	3.75
		var	**0.56**	22.27	**0.19**	0.68
	Erindale-2008-41	mean	**9.90**	10.46	**6.63**	6.66
		var	**1.36**	2.39	**0.06**	0.12
	Erindale-2008-52	mean	**5.11**	5.93	**3.21**	3.91
		var	**2.63**	1.10	**0.18**	0.55
	Erindale-2009-02	mean	**4.43**	6.42	**1.29**	3.69
		var	**1.06**	1.96	**0.16**	0.97
	Erindale-2009-12	mean	**5.43**	6.42	**2.29**	2.69
		var	**3.02**	2.22	**0.18**	1.21
	Erindale-2009-27	mean	**11.77**	12.92	**6.80**	7.27
		var	**15.32**	32.20	**0.09**	1.03
Average mean			5.41	7.26	3.90	4.14
Average variance			0.97	7.10	0.70	0.25
Average rank			1	2	1	2
Number of first ranks			12	0	12	0

The results obtained are in bold font, when they achieve the smallest test errors among the others

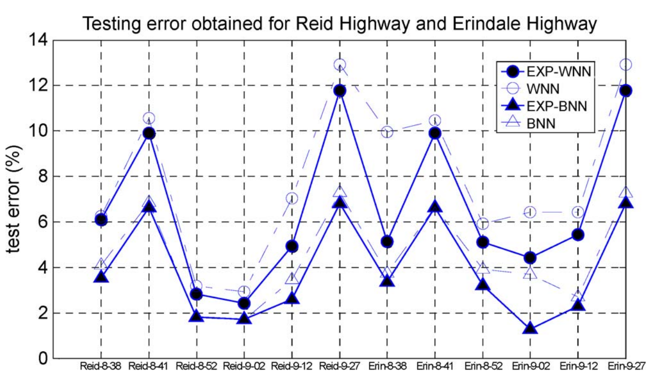


Fig. 4. Mean test errors obtained for Reid Highway and Erindale Highway based on EXP-WNN, WNN, EXP-BNN, and BNN.

EXP-WNN-LM are generally higher than those obtained by WNN-LM, whereas those obtained by EXP-BNN-LM are also generally higher than those obtained by BNN-LM. To illustrate the generalization capability of the NNs developed based on these algorithms, Table VI shows that EXP-WNN-LM yields smaller mean test errors and better ranks of mean test errors compared with those obtained by WNN-LM, whereas those obtained by EXP-BNN-LM are better compared with those obtained by BNN-LM. In addition, the mean test errors obtained by each algorithm are shown in Fig. 4, indicating that the mean test errors obtained by EXP-WNN-LM and EXP-BNN-LM are smaller than those obtained by WNN-LM and BNN-LM, respectively. The t-values in Table VII show that, in general, the test errors obtained by EXP-WNN-LM are significantly smaller than those obtained by WNN-LM, and the test errors obtained by EXP-BNN-LM are significantly smaller than those obtained by BNN-LM. Therefore, these results indicate that NN models with better generalization capability can be obtained

TABLE VII
T-VALUES BETWEEN EXP-WNN AND WNN, AND BETWEEN EXP-BNN AND BNN

		WNN	BNN
$NN_6^{\text{Re}id}$	Reid-2008-38	0.22	**1.79**
	Reid-2008-41	**1.86**	**2.91**
	Reid-2008-52	**1.77**	0.20
	Reid-2009-02	**4.93**	**5.37**
	Reid-2009-12	**4.71**	**6.63**
	Reid-2009-27	0.92	**2.40**
NN_6^{Erindale}	Erindale-2008-38	**5.52**	**2.12**
	Erindale-2008-41	**1.86**	0.38
	Erindale-2008-52	**2.30**	**4.45**
	Erindale-2009-02	**6.27**	**12.37**
	Erindale-2009-12	**2.37**	**1.86**
	Erindale-2009-27	0.92	**2.40**
Number of data sets that EXP-LM can obtain significantly better results		9	10

Bolded t-values indicate that significance is 95% confidence level.

by the algorithms involving exponential smoothing methods: EXP-BNN-LM and EXP-WNN-LM.

Similar results were found on training NNs for forecasting traffic flow in two sampling periods ahead by using EXP-LM, EXP-BNN-LM, and EXP-WNN-LM. In addition, similar results were obtained when using the back-propagation algorithm, indicating that better generalization capability can be obtained by training with traffic flow data, which are pre-processed by the exponential smoothing method. These results are not presented here due to page limitation.

C. Further Evaluations

In Section IV-B, the NNs were developed based on traffic flow data collected between the off-ramp and on-ramp of a particular location. To further evaluate the effectiveness of the proposed approach, traffic flow data collected from different locations were used to develop NNs. These traffic flow data were collected from the location in the intersection of Reid Highway and Mitchell Freeway, as well as from the location in the intersection of Hutton Street and Mitchell Freeway, Western Australia. The distance between the two locations is about 6 km. If the traffic flow condition is smooth, drivers usually take 6 min to drive along Mitchell Freeway from Reid Highway to Hutton Street.

Thirty traffic flow data sets were collected from Week 6, Week 7, Week 8, Week 9, Week 11, and Week 12 in 2009, and they were collected over the 2-h peak traffic period (7.30–9.30 am). All these data sets were collected by two detection stations located near the off-ramp and on-ramp of Reid Highway, as well as the three detection stations located near the off-ramp of Hutton Street, between the off-ramp and on ramp of Hutton Street, and near the on-ramp of Hutton Street, respectively.

The five NNs NN_{Mon}^{Hutton}, NN_{Tue}^{Hutton}, NN_{Wed}^{Hutton}, NN_{Thu}^{Hutton}, and NN_{Fri}^{Hutton} were developed for Monday,

TABLE VIII
TRAINING ERROR OBTAINED FOR HUTTON STREET BASED ON EXP-LM, SM-LM, WM-LM, S-LM, LM-CROSS-(5), AND LM-CROSS-(10)

		EXP-LM	SM-LM	WM-LM	S-LM	LM-CROSS-(5)	LM-CROSS-(10)
NN^{Hutton}_{Mon}	mean	11.39	11.25	10.66	9.04	9.40	10.05
	var	6.57	7.17	5.95	6.00	6.48	8.25
NN^{Hutton}_{Tue}	mean	11.34	11.24	10.87	9.10	8.38	10.59
	var	7.71	8.06	8.46	7.49	6.97	11.56
NN^{Hutton}_{Wed}	mean	10.72	8.68	11.53	11.09	12.38	13.40
	var	6.70	4.99	6.85	6.71	5.97	8.10
NN^{Hutton}_{Thu}	mean	10.79	10.42	10.25	8.17	7.122	7.44
	var	11.83	11.61	11.05	9.12	11.16	10.51
NN^{Hutton}_{Fri}	mean	10.80	10.61	10.33	8.24	9.20	10.41
	var	11.61	12.02	12.10	7.79	6.87	9.60
Average mean		11.15	10.39	11.02	9.74	10.05	11.35
Average variance		7.14	7.61	7.21	6.74	6.72	9.91
Average rank		5	3	4	1	2	6

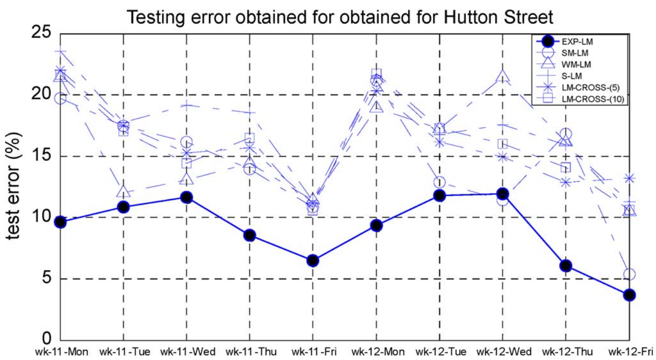


Fig. 5. Mean test errors obtained for Hutton Street for Weeks 11 and 12.

Tuesday, Wednesday, Thursday, and Friday, respectively. They were developed to forecast six sampling periods ahead the traffic flow conditions near the on-ramp of Hutton Street by using the last six sampling periods of past traffic flow conditions. The traffic flow data sets collected from Week 6 to Week 9 were used as the training data sets to train the five NNs. The rest of the two traffic flow data sets, which were collected from Week 11 and Week 12, were used as the test data sets to evaluate the generalization capability of the NNs.

Table VIII shows that the training errors obtained by EXP-LM are generally larger than those obtained by the other algorithms, i.e., SM-LM, WM-LM, S-LM, LM-CROSS-(5), and LM-CROSS-(10). Table IX shows that the mean of test errors, variance of test errors, and rank of test errors obtained by EXP-LM are generally smaller than those obtained by the other algorithms. In addition, Fig. 5 shows further that the mean test errors obtained by EXP-LM are generally smaller than those obtained by the other algorithms. The t-values in Table X show that, in general, the test errors obtained by EXP-LM are significantly smaller than those obtained by the other algorithms. Therefore, these results further evaluate that NNs with better generalization capability can be obtained by EXP-LM involving exponential smoothing methods.

V. CONCLUSION

In this paper, a hybrid exponential smoothing method and LM algorithm called EXP-LM has been proposed to train NNs for short-term traffic flow forecasting. EXP-LM has been developed based on the observation that the landscape of traffic flow data is highly lumpy. When lumpiness is included in the training of NN models, training errors of NN models can be decreased to a small value by fitting all of the lumpiness, but it may degrade the generalization capability, as the lumpiness may not be helpful in training NNs. In the proposed EXP-LM, the exponential smoothing method has been employed to remove the lumpiness from traffic flow data before employing LM for training purposes. Results indicate that, in general, test errors obtained by EXP-LM are smaller than those obtained by the other tested algorithms. Therefore, in general, NNs with superior generalization capabilities for traffic flow forecasting can be obtained by using EXP-LM.

Future research will be focused on three areas: 1) Work is currently under way to build a prototype to capture real-time traffic flow data from a number of different freeway locations, under recurrent and nonrecurrent congestion conditions, to further evaluate NN models trained using the EXP-LM algorithm. 2) The proposed method will be applied to preprocess travel time data or congestion data, which contain lumpiness.

TABLE IX
TEST ERRORS OBTAINED FOR HUTTON STREET FOR WEEK 11 AND WEEK 12, BASED ON EXP-LM, SM-LM, WM-LM, S-LM, LM-CROSS-(5), AND LM-CROSS-(10)

			EXP-LM	SM-LM	WM-LM	S-LM	LM-CROSS-(5)	LM-CROSS-(10)
Week 11	NN^{Hutton}_{Mon}	mean	**9.63**	19.74	21.40	23.58	22.03	21.66
		var	**11.79**	69.34	57.36	65.24	64.90	50.26
	NN^{Hutton}_{Tue}	mean	**10.86**	17.49	12.04	17.77	17.53	17.07
		var	**2.44**	65.82	53.75	23.10	31.55	33.50
	NN^{Hutton}_{Wed}	mean	**11.67**	16.14	13.06	19.16	15.29	14.44
		var	**3.94**	58.77	63.65	34.06	54.54	44.30
	NN^{Hutton}_{Thu}	mean	**8.57**	13.97	14.44	18.53	15.70	16.50
		var	**25.05**	79.45	76.42	157.8	108.0	98.21
	NN^{Hutton}_{Fri}	mean	**6.49**	10.86	11.44	11.37	11.18	10.60
		var	**2.47**	25.73	23.82	11.48	9.41	14.15
Week 12	NN^{Hutton}_{Mon}	mean	**9.37**	21.20	18.93	21.55	20.34	21.73
		var	**9.30**	73.31	56.65	64.74	67.49	57.82
	NN^{Hutton}_{Tue}	mean	**11.78**	12.87	17.24	16.78	16.17	17.30
		var	**6.39**	16.39	56.95	21.14	38.66	38.80
	NN^{Hutton}_{Wed}	mean	11.95	**11.41**	21.46	17.56	14.95	16.03
		var	4.42	**26.09**	67.23	36.34	49.51	46.62
	NN^{Hutton}_{Thu}	mean	**6.09**	16.80	16.16	15.93	12.86	14.11
		var	**30.86**	77.28	81.24	155.2	113.4	102.2
	NN^{Hutton}_{Fri}	mean	**3.70**	5.36	10.48	11.30	13.20	10.62
		var	**1.13**	48.75	27.16	16.01	12.59	15.32
Average mean			9.01	14.59	16.61	17.35	15.92	16.00
Average variance			9.78	54.09	56.42	58.50	55.00	50.12
Average rank			1	2	5	6	3	4
Number of first ranks			9	1	0	0	0	0

The results obtained are in bold font, when they achieve the smallest test errors among the others

TABLE X
T-VALUES BETWEEN EXP-LM TO THE OTHER ALGORITHMS [SM-LM, WM-LM, S-LM, LM-CROSS-(5), AND LM-CROSS-(10)] FOR HUTTON STREET

		SM-LM	WM-LM	S-LM	LM-CROSS-(5)	LM-CROSS-(10)
Week 11	NN^{Hutton}_{Mon}	**6.15**	**7.75**	**8.71**	**7.76**	**8.36**
	NN^{Hutton}_{Tue}	**4.39**	0.86	**7.49**	**6.26**	**5.67**
	NN^{Hutton}_{Wed}	**3.09**	0.92	**6.66**	**2.59**	**2.18**
	NN^{Hutton}_{Thu}	**2.89**	**3.19**	**4.03**	**3.38**	**3.91**
	NN^{Hutton}_{Fri}	**4.51**	**5.29**	**7.15**	**7.46**	**5.52**
Week 12	NN^{Hutton}_{Mon}	**7.13**	**6.45**	**7.75**	**6.86**	**8.26**
	NN^{Hutton}_{Tue}	1.25	**6.15**	**5.22**	**3.58**	**4.50**
	NN^{Hutton}_{Wed}	0.53	-2.10	**4.82**	**2.24**	**3.12**
	NN^{Hutton}_{Thu}	**5.64**	**5.21**	**3.95**	**3.08**	**3.80**
	NN^{Hutton}_{Fri}	1.29	**6.98**	**10.78**	**15.37**	**10.06**
Number of data sets that EXP-LM can obtain significantly better results		7	7	10	10	10

Bolded t-values indicate that significance is 95% confidence level.

The preprocessed data will be applied to develop travel time predictors [56], [58] or congestion predictors [57], which is an important issue of intelligent transportation systems. 3) We will develop a methodology to determine the optimal numbers of hidden nodes and input nodes, which are significant to prevent overfitting.

REFERENCES

[1] I. G. Ali and Y. T. Chen, "Design quality and robustness with neural networks," *IEEE Trans. Neural Netw.*, vol. 10, no. 6, pp. 1518–1527, Nov. 1999.

[2] S. Amari, N. Murata, K. Muller, M. Finke, and H. H. Yang, "Asymptotic statistical theory of overtraining and cross validation," *IEEE Trans. Neural Netw.*, vol. 8, no. 5, pp. 985–996, Sep. 1997.

[3] V. B. Arem, H. R. Kirby, M. J. M. Van Der Vlist, and J. C. Whittaker, "Recent advances and applications in the field of short-term traffic forecasting," *Int. J. Forecasting*, vol. 13, no. 1, pp. 1–12, Mar. 1997.

[4] R. G. Brown and R. F. Meyer, "The fundamental theorem of exponential smoothing," *Oper. Res.*, vol. 9, no. 5, pp. 673–685, Sep./Oct. 1961.

[5] M. Castro-Neto, Y. S. Jeong, M. K. Jeong, and L. D. Han, "Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 6164–6173, Apr. 2009.

[6] P. C. Chang, Y. W. Wang, and C. H. Liu, "The development of a weighted evolving fuzzy neural network for PCB sales forecasting," *Expert Syst. Appl.*, vol. 32, no. 1, pp. 86–96, 2007.

[7] S. C. Chang, R. S. Kim, S. J. Kim, and M. H. Ahn, "Traffic flow forecasting using a 3-stage model," in *Proc. IEEE Intell. Veh. Symp.*, 2000, pp. 451–456.

[8] H. C. Co and R. Boosarawongse, "Forecasting Thailand's rice export: Statistical techniques vs. artificial neural networks," *Comput. Ind. Eng.*, vol. 53, no. 4, pp. 610–627, Nov. 2007.

[9] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis," *IEEE Trans. Inf. Theory*, vol. 36, no. 5, pp. 961–1005, Sep. 1990.

[10] G. A. Davis and N. L. Nihan, "Nonparametric regression and short-term freeway traffic forecasting," *J. Transp. Eng.*, vol. 177, no. 2, pp. 178–188, Apr. 1991.

[11] S. De Lurgio, *Forecasting Principles and Applications.*. New York: McGraw-Hill, 1998.

[12] H. Dia, "An object-oriented neural network approach to short-term traffic forecasting," *Eur. J. Oper. Res.*, vol. 131, no. 2, pp. 253–261, Jun. 2001.

[13] T. S. Dillon, S. Sestito, and S. Leung, "Short term load forecasting using an adaptive neural network," *Elect. Power Energy Syst.*, vol. 13, no. 4, pp. 186–192, 1991.

[14] M. Dougherty, "A review of neural networks applied to transport," *Transp. Res. Part C: Emerging Technol.*, vol. 3, no. 4, pp. 247–260, 1995.

[15] M. S. Dougherty, H. R. Kirby, and R. D. Boyle, "Using neural networks to recognize, predict and model traffic," in *Artificial Intelligence Applications to Traffic Engineering*. Utrecht, The Netherlands: VSP, 1994, pp. 235–250.

[16] M. S. Dougherty and M. R. Cobbett, "Short-term inter-urban traffic forecasts using neural networks," *Int. J. Forecasting*, vol. 13, no. 1, pp. 21–31, Mar. 1997.

[17] E. L. Faria, M. P. Albuquerque, J. L. Gonzalez, J. T. P. Cavalcanta, and M. P. Albuquerque, "Predicting the Brazilian stock market through neural networks and adaptive exponential smoothing methods," *Expert Syst. Appl.*, vol. 36, no. 10, pp. 12 506–12 509, Dec. 2009.

[18] F. D. Foresee and M. T. Hagan, "Gauss-Newton approximation to Bayesian regularization," in *Proc. Int. Joint Conf. Neural Netw.*, 1997, pp. 1930–1935.

[19] L. Fu and L. R. Rilett, "Estimation of time-dependent, stochastic route travel times using artificial neural networks," *Transp. Planning Technol.*, vol. 24, no. 1, pp. 25–48, 2000.

[20] Y. Gao and M. J. Er, "NARMAX time series model prediction: Feedforward and recurrent fuzzy neural network approaches," *Fuzzy Sets Syst.*, vol. 150, no. 2, pp. 331–350, Mar. 2005.

[21] R. S. Gutierrez, O. S. Adriano, and S. Mukhopadhyay, "Lumpy demand forecasting using neural networks," *Int. J. Prod. Econ.*, vol. 111, no. 2, pp. 409–420, Feb. 2008.

[22] S. Innamaa, "Effect of monitoring system structure on short-term prediction of highway travel time," *Transp. Planning Technol.*, vol. 29, no. 2, pp. 125–140, Apr. 2006.

[23] X. Jiang and H. Adeli, "Dynamic wavelet neural network model for traffic flow forecasting," *J. Transp. Eng.*, vol. 131, no. 10, pp. 771–779, Oct. 2005.

[24] C. F. Juang, "Temporal problems solved by dynamic fuzzy network based on genetic algorithm with variable-length chromosomes," *Fuzzy Sets Syst.*, vol. 142, no. 2, pp. 199–219, Mar. 2004.

[25] N. K. Kasabov and Q. Song, "DENFIS: Dynamic evolving neural-fuzzy inference system and its application for time-series prediction," *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 2, pp. 144–154, Apr. 2002.

[26] G. N. Karystinos and D. A. Pados, "On overfitting, generalization, and randomly expanded training sets," *IEEE Trans. Neural Netw.*, vol. 11, no. 5, pp. 1050–1057, Sep. 2000.

[27] S. Lawrence, C. L. Giles, and A. C. Tsoi, "What size neural network gives optimal generalization? Convergence properties of backpropagation," Instrum. Adv. Comput. Studies, Univ. Maryland, College Park, MD, Tech. Rep., UMIACS-TR-96-22 and CS-TR-3617, Jun. 1996.

[28] C. Ledoux, "An urban traffic flow model integrating neural network," *Transp. Res. Part C, Emerging Technol.*, vol. 5, no. 5, pp. 287–300, Oct. 1997.

[29] G. L. Lilien and P. Kotler, *Marketing Decision Making: A Model Building Approach*. New York: Harper & Row, 1983.

[30] F. Liang, "Bayesian neural networks for nonlinear time series forecasting," *Statist. Comput.*, vol. 15, no. 1, pp. 13–29, Jan. 2005.

[31] Y. Liu, J. A. Starzyk, and Z. Zhu, "Optimized approximation algorithm in neural networks without overfitting," *IEEE Trans. Neural Netw.*, vol. 19, no. 6, pp. 983–995, Jun. 2008.

[32] D. MacKay, "A practical Bayesian framework for backpropagation networks," *Neural Comput.*, vol. 4, no. 3, pp. 448–472, May 1992.

[33] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 7, pp. 674–693, Jul. 1989.

[34] P. A. Mastorocostas and J. B. Theocharis, "An orthogonal least-squares method for recurrent fuzzy-neural modeling," *Fuzzy Sets Syst.*, vol. 140, no. 2, pp. 285–300, Dec. 2003.

[35] I. Okutani and Y. J. Stephanedes, "Dynamic prediction of traffic volume through Kalman filtering theory," *Transp. Res. Part B: Methodology*, vol. 18, no. 1, pp. 1–11, Feb. 1984.

[36] S. E. Papadakis, J. B. Theocharis, and A. G. Bakirtzis, "A load curve based fuzzy modeling technique for short-term load forecasting," *Fuzzy Sets Syst.*, vol. 135, no. 2, pp. 279–303, Apr. 2003.

[37] L. Prechelt, "Automatic early stopping using cross validation: Quantifying the criteria," *Neural Netw.*, vol. 11, no. 4, pp. 761–767, Jun. 1998.

[38] P. Ross, "Exponential filtering of traffic data," in *Proc. Transp. Res. Board*, Washington, DC, 1982, vol. 869, pp. 43–49.

[39] C. Quek, M. Pasquier, and B. B. S. Lim, "POP-TRAFFIC: A novel fuzzy neural approach to road traffic analysis and prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 2, pp. 133–146, Jun. 2006.

[40] R. Setiono, "Feedforward neural network construction using cross validation," *Neural Comput.*, vol. 13, no. 12, pp. 2865–2877, Dec. 2001.

[41] D. Srinivasan, C. W. Chan, and P. G. Balaji, "Computational intelligence based congestion prediction for a dynamic urban street network," *Neurocomputing*, vol. 72, no. 10–12, pp. 2710–2716, 2009.

[42] B. L. Smith, B. M. Williams, and R. K. Oswald, "Comparison of parametric and nonparametric models for traffic flow forecasting," *Transp. Res. Part C, Emerging Technol.*, vol. 19, pp. 303–321, 2002.

[43] A. Stathopoulos, L. Dimitriou, and T. Tsekeris, "Fuzzy modeling approach for combined forecasting of urban traffic flow," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 23, no. 7, pp. 521–535, Oct. 2008.

[44] M. C. Tan, S. C. Wong, J. M. Xu, Z. R. Guan, and P. Zhang, "An aggregation approach to short term traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 1, pp. 60–69, Mar. 2009.

[45] R.-J. Wai, R.-Y. Duen, J.-D. Lee, and H.-H. Chang, "Wavelet neural network control for induction motor drive using sliding-mode design technique," *IEEE Trans. Ind. Electron.*, vol. 50, no. 4, pp. 733–748, Aug. 2003.

[46] N. Wanas, G. Auda, M. S. Kamel, and F. Karray, "On the optimal number of hidden nodes in a neural network," in *Proc. IEEE Can. Conf. Elect. Comput. Eng.*, 1998, vol. 2, pp. 918–921.

[47] R. A. Yaffee and M. McGee, *Introduction to Time Series Analysis and Forecasting*. San Diego, CA: Academic, 2000.

[48] H. Yin, S. C. Wong, J. Xu, and C. K. Wong, "Urban traffic flow prediction using a fuzzy-neural approach," *Transp. Res. Part C, Emerging Technol.*, vol. 10, no. 2, pp. 85–98, Apr. 2002.

[49] S. J. Yoo, J. B. Park, and Y. H. Choi, "Adaptive dynamic surface control of flexible-joint robots using self-recurrent wavelet neural networks," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 36, no. 6, pp. 1342–1355, Dec. 2006.

[50] G. P. Zhang and D. M. Kline, "Quarterly time series forecasting with neural networks," *IEEE Trans. Neural Netw.*, vol. 18, no. 6, pp. 1800–1814, Nov. 2007.

[51] G. P. Zhang and M. Qi, "Neural network forecasting for seasonal and trend time series," *Eur. J. Oper. Res.*, vol. 160, no. 2, pp. 501–514, Jan. 2005.

[52] Q. Zhang and A. Benveniste, "Wavelet networks," *IEEE Trans. Neural Netw.*, vol. 3, no. 6, pp. 889–898, Nov. 1992.

[53] W. Zheng, D. H. Lee, and Q. Shi, "Short-term freeway traffic flow prediction: Bayesian combined neural network approach," *J. Transp. Eng.*, vol. 132, no. 2, pp. 114–121, Feb. 2006.

[54] K. Y. Chan, J. Singh, T. S. Dillon, and E. Chang, "Traffic flow forecasting neural networks based on exponential smoothing method," in *Proc. 6th IEEE Conf. Ind. Electron. Appl.*, 2011, pp. 376–381.

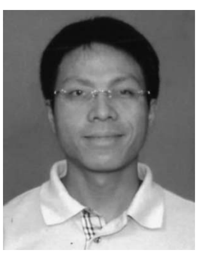
[55] G. E.-P. Box, J. S. Hunter, and W. G. Hunter, *Statistics for Experiments: Design, Innovation, and Discovery,* 2nd ed. New York: Wiley, 2005.

[56] A. Simroth and H. Zähle, "Travel time prediction using floating car data applied to logistics planning," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 1, pp. 243–253, Mar. 2011.

[57] T. Thomas, W. Weijermars, and E. Berkum, "Predictions of urban volumes in single time series," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 1, pp. 71–80, Mar. 2010.

[58] M. Yang, Y. Liu, and Z. You, "The reliability of travel time forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 1, pp. 162–171, Mar. 2010.

[59] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Trans. Neural Netw.*, vol. 5, no. 6, pp. 989–993, Nov. 1994.



Kit Yan Chan (M'11) received the Ph.D. degree in computing from London South Bank University, London, U.K., in 2006.

After his Ph.D. study, he was a Postdoctoral Research Fellow with the Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hung Hom, Hong Kong, until 2009. He is currently a Senior Research Fellow with the Digital Ecosystems and Business Intelligence Institute, Curtin University of Technology, Perth, Australia. His research interests include computational intelligence and its applications to new product design, manufacturing process design, and traffic flow forecasting.

Tharam S. Dillon (M'74–SM'87–F'98–LF'10) received the Ph.D. degree in electrical and computer systems engineering from Monash University, Australia, in 1974.

He is currently a Research Professor with the Digital Ecosystems and Business Intelligence Institute, Curtin University of Technology, Perth, Australia. He is the Editor-in-Chief of the *International Journal of Computer Systems Science and Engineering* and *Engineering Intelligent Systems*. He has published more than 750 papers published in international conferences and journals. He is the author of five books and the editor of five books. His current research interests include Web semantics, ontologies, Internet computing, e-commerce, hybrid neurosymbolic systems, neural nets, software engineering, database systems, and data mining.

Prof. Dillon is a Fellow of the Australian Computer Society and Institution of Engineers (Australia). He is the Head of the International Federation for Information Processing (IFIP) International Task Force WG2.12/24 on Semantic Web and Web Semantics, the Chairman of the IFIP WG12.9 on computational intelligence, the IEEE Industrial Electronics Society Technical Committee on Industrial Informatics, and the IFIP Technical Committee 12 on Artificial Intelligence.

Jaipal Singh (M'09) received the Ph.D. degree in computer science from La Trobe University, Melbourne, Australia in 2007.

He has been doing research on network algorithms for more efficient routing, resource optimization, and improved quality of service. He is currently a Research Fellow with Curtin University, Perth, Australia, working on short-term traffic forecasting, traffic visualization, and real-time congestion management. He is also developing new cyberphysical systems architectures in intelligent transportation systems.

Elizabeth Chang (M'02–SM'07) received the Bachelor's degree from Beijing University, Beijing, China, in 1985 and the Master's and Ph.D. degrees from La Trobe University, Melbourne, Australia, in 1991 and 1996, respectively, all in computer science.

Since 2006, she has been the Founder and Professor of the Digital Ecosystems and Business Intelligence Institute, Curtin University, Perth, Australia. She has coauthored three books and has published more than 350 scientific papers as book chapters and in international journals and conference proceedings. She is currently the holder of six Australian Research Council (ARC) grants and a Tier 1 Centre of Excellence grant and obtained cash from ARC, industry partners, and the Research Centre of Excellence funds of more than $5 million for 2002–2011. Her research interests include ontology and multiagent systems, data mining for business intelligence, trust, security and risk in e-Business, XML, Web Services, peer-to-peer for collaborative environments, web engineering, information technology (IT) for business and commerce, IT for health informatics, and IT for education.

Dr. Chang received of the Vice Chancellor's Outstanding Performance Award in 2005 and the Dean's Best Researcher of Year Award in both 2004 and 2005.