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 (Fellow IEEE), Jaipal Singh (Member IEEE), Elizabeth Chang (Senior Member IEEE)

***Abstract* —** This paper proposes a novel neural network training method that employs the hybrid exponential smoothing method and the Levenberg-Marquardt algorithm, which aims to improve the generalization capabilities of previously used methods for training neural networks for short-term traffic flow forecasting. The approach uses exponential smoothing to pre-process traffic flow data, by removing the lumpiness from collected traffic flow data, before employing a variant of the Levenberg-Marquardt algorithm to train the NN weights of a neural network model. This approach aids neural network training, as the pre-processed traffic flow data is 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. In regards to the generalization capabilities for short-term traffic flow forecasting, the neural network models developed using the proposed approach outperform those that are developed based on the alternative tested algorithms, which are either designed particularly for short-term traffic flow forecasting or designed for enhancing generalization capabilities of neural networks.

***Index Terms*:** Neural networks, exponential smoothing method, short-term traffic flow forecasting, Levenberg-Marquardt algorithm

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, 57]. Short-term traffic flow forecasting models can be generated by conventional statistical methods such as filtering techniques [39, 36], autoregressive integrated moving average (ARIMA) methods [43] 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: a) it is difficult to specify the most suitable model without human expertise; b) the models generated by these methods may not be able to capture some strongly non-linear characteristics of short-term traffic flow data. In order to address these limitations, neural network (NN) approaches have commonly been used for short-term traffic flow forecasting [7, 12, 14, 15, 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 the following two classes:

a) **Hybrid NN approaches**, that incorporate other computational intelligence methods or statistical prediction methods that have been investigated recently for enhancing the generalization capabilities of NNs. A commonly used method for time series forecasting, Takagi-Sugeno fuzzy NNs [24, 25, 35, 37], which combine the mechanisms of fuzzy logic and feed-forward NNs, has been proposed for short-term traffic flow forecasting [20, 40, 49]. Stathopoulos et al. [44] 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 a NN and a Kalman filter. Srinivasan et al. [42] proposed a hybrid NN which consists of two components: a fuzzy filter and a feed-forward NN. The fuzzy filter performs the clustering operation on traffic flow data and provides a rough prediction, which is the input of the feed-forward NN. Accurate short-term traffic flow forecasting is produced by the feed-forward NN, which utilizes each cluster as input for modeling the input–output relation. Tan et al. [45] has proposed a hybrid NN, which combines the mechanism of a NN with the 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 tuned adaptively compared with the pure NNs, as more expensive processors with more memory space and computational power are required for the hybrid NNs.

b) **Preventing overfitting in neural network 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 is required due to the additional training data that is required to be fitted by the NN models. Another commonly used approach is cross validation [41], where the training data is divided into two data sets: 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. [32] 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, namely hybrid exponential smoothing and Levenberg-Marquardt (LM)

* Kit Yan Chan is the corresponding author. His email address is Kit.Chan@curtin.edu.au

algorithm (EXP-LM), is proposed to train NNs, in order to produce high generalization capability in 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, 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 [55]. In EXP-LM, the exponential smoothing method [11, 48] 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 a 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 EXP-BP compared with the other tested methods.

The rest of the 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 the EXP-LM. Section IV shows and discusses the results obtained by EXP-LM and the 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. NNs 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, … D_n), which are located between the off-ramp and on-ramp of the freeway as illustrated in Fig. 1. D_i captures two traffic flow measures, 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, 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 $s_i(t\text{-}k\cdot T_s)$ and past average headway $h_i(t\text{-}k\cdot T_s)$, which was collected by D_i at time $(t\text{-}k\cdot T_s)$ with i=1, 2, …, n and k=1, 2, …, p, while the past traffic flow data within p sampling time interval/period is 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 L-th detector station D_L at time $(t+m\cdot T_s)$, where future traffic flow with m sampling time ahead is forecasted.

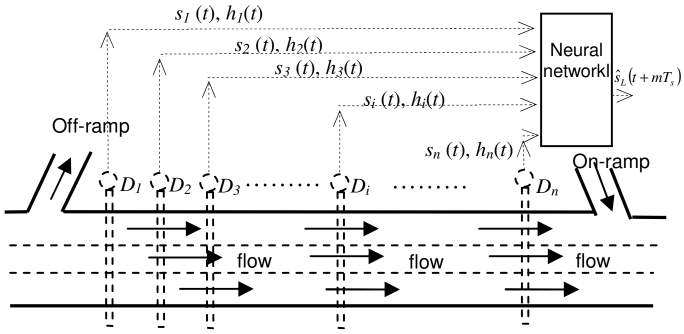


Fig. 1 Schematic of short-term traffic flow forecasting of the freeway

To predict the future traffic flow at the location of D_L, the following multilayer (3 layers) NN is implemented, where satisfactory results can be obtained for traffic flow forecasting [7, 14, 16, 22]. The NN is formulated as:

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

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, namely 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 is in the form of

$$d(l)=\left[\theta(l),\varphi(l)\right] \text{ with } l=1, 2, \dots N_D, \quad (2)$$

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

$$\theta(l)=s_L(t(l)+mT_s); \quad (3)$$

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

$$\begin{aligned}\varphi(l)=\Big[&h_1(t(l)-T_s),h_1(t(l)-2T_s),\dots,h_1(t(l)-pT_s),\\&h_2(t(l)-T_s),h_2(t(l)-2T_s),\dots,h_2(t(l)-pT_s),\dots,\\&h_n(t(l)-T_s),h_n(t(l)-2T_s),\dots,h_n(t(l)-pT_s),\\&s_1(t(l)-T_s),s_1(t(l)-2T_s),\dots,s_1(t(l)-pT_s),\\&s_2(t(l)-T_s),s_2(t(l)-2T_s),\dots,s_2(t(l)-pT_s),\dots,\\&s_n(t(l)+T_s),s_n(t(l)-2T_s),\dots,s_n(t(l)-pT_s)\Big]\end{aligned}; \quad (4)$$

$h_i(t(l))$ 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 l-th traffic flow data. Based on the collected traffic flow data $d(l)=\left[\theta(l),\varphi(l)\right]$ with l=1, 2, … 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. The e_{MARE} is formulated as:

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

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

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

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

Then, the Levenberg-Marquardt algorithm, namely LM algorithm, is a commonly used method to train NNs by minimizing the mean absolute relative error e_{MARE} [60]. It starts by randomly generating the first two initial guesses of NN weights $w(0)$ and $w(1)$ at the 0-th and the 1-st iterations, where

$$w(0) = \left[\alpha_0(0), \beta_{j,i}^h(0), \beta_{j,i}^s(0), \gamma_{0,j,i}^h(0), \gamma_{0,j,i}^s(0), \gamma_{k,j,i}^h(0), \gamma_{k,j,i}^s(0)\right] \quad (7)$$

and $w(1) = \left[\alpha_0(1), \beta_{j,i}^h(1), \beta_{j,i}^s(1), \gamma_{0,j,i}^h(1), \gamma_{0,j,i}^s(1), \gamma_{k,j,i}^h(1), \gamma_{k,j,i}^s(1)\right]$ (8)

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

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

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

The details of determination of the Jacobian matrix, $J\left(w(\varsigma)\right)$, can be referred to [60]. To forecast average headway, similar formulation can be used by replacing $\hat{s}_L\left(t(l)+mT_s\right)$ with the forecast average headway, $\hat{h}_L\left(t(l)+mT_s\right)$, in equation (1). Also, equation (3) is redefined by $\theta(l) = h_L\left(t(l)+mT_s\right)$.

III. Hybrid 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 was collected from Mitchell Freeway, which is near the on-ramp of Reid Highway, Western Australia. It was collected over the 2-hour peak traffic periods (7.30 – 9.30 am) on 18 December 2008, where the sampling time was 1 minute. A NN can be obtained by fitting all the collected traffic flow data that has 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, namely hybrid exponential smoothing and LM (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, a simple and intensively used method for pre-processing 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 pre-processed data which is denoted by $d'(l) = \left[\theta'(l), \varphi(l)\right]$ with l =1, 2, …,N_D.

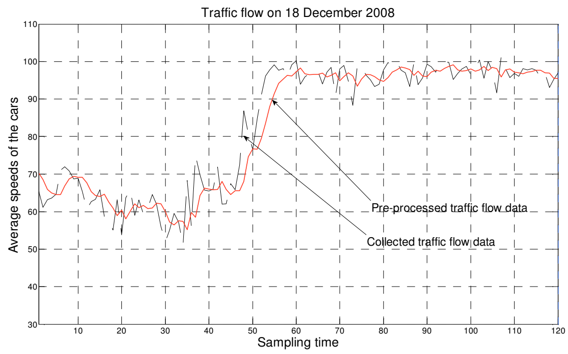


Fig. 2 Traffic flow data collected from 18 December 2008

In EXP-LM, the l-th filtered traffic flow data, $\theta'(l)$, is 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 $\left(\theta(l-1)-\theta'(l-1)\right)$, which is expected at the l-th traffic flow data. The l-th filtered traffic flow data, $\theta'(l)$, with $l \geq 3$ is defined by the following equation:

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

where α is the smoothing constant within the range, $0.1 < \alpha \leq 0.9$. The 1-st and the 2-nd filtered traffic flow data are initialized by $\theta'(1) = \theta(1)$ and $\theta'(2) = \left(\theta(1)+\theta(2)+\theta(3)\right)/3$ 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 $\left(0.8/N_G\right)$ of the parameter space between α =0.1 and α =0.9 is used, where N_G is the number of grids of the grid search. The higher N_G is, the smaller $R^2(\alpha)$ can be obtained. The best α is chosen so as to produce the smallest sum of squares for the residuals which is defined as:

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

The filtered traffic flow data which was pre-processed by the exponential smoothing method is shown in Fig. 2. It shows that the filtered traffic data seeks 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 excludes the lumpiness is used for training the NN, better generalization capability is more likely to be developed by the EXP-LM. The mechanism of EXP-LM is illustrated by the following steps:

Step 1: Collect the traffic flow data $d(l) = [\theta(l), \varphi(l)]$ in the form of equation (2) with l=1, 2, … 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, …, 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, …, N_G, based on (12).

Step 2.3: Determine the best exponential smoothing parameter, $\alpha_{best} = \alpha(i_2)$, 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 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 l-th filtered traffic flow data based on equation (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 equations (7) and (8) respectively.

Step 4.2: Update the NN weights, $w(\varsigma + 1)$, based on equation (9), where e_{MARE} and $\hat{\theta}(l)\big|_{w(\varsigma)} = \hat{s}_L(t(l) + mT_s)\big|_w$ is determined by equations (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 1, 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 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, as well as near the on-ramp of Erindale Road, respectively.

The traffic flow data sets were collected over the 2-hour peak traffic period (7.30 – 9.30 am) on the five business days of the week, Monday, Tuesday, Wednesday, Thursday and Friday. 60 seconds (1 minute) of sampling time were 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 sub-sets. The first sub-set of traffic flow data, namely training data, collected from Monday to Thursday (comprising 80% of all the observations), was used for training the NNs. The second sub-set of traffic flow data, namely test data, collected from Friday (comprising 20% of all the observations), was used to evaluate the generalization capability of the trained NNs.

Table 1 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

B. Experimental results

The EXP-LM was implemented in Matlab. Four NNs (namely 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.

1) 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 the EXP-LM.

1. **Standard LM algorithm, namely 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 pre-process data but S-LM does not.
2. **The hybrid simple moving average and LM algorithm namely 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 *l*-th filtered traffic flow data, $\theta'(l)$ is generated based on the past four traffic flow data as:

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

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

3. **The hybrid weighted moving and LM algorithm, namely WM-LM,** use the weighted moving method to filter lumpiness in the traffic flow data. In the WM-LM, the *l*-th filtered traffic flow data, $\theta'(l)$, is generated based on the past four traffic flow data as:

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

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 neural networks.

4. **The cross-validation based LM algorithm, namely LM-CROSS-(τ),** uses the mechanisms of cross-validation [2, 38] to avoid overtraining NNs. In LM-CROSS-(τ), the fitting data (comprising 60% of all the observations) was used for computing the NN weights, while the cross-validation data (comprising 20% of all the observations) was 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) stop training the NNs, when the error for the cross-validation data at the ($\varsigma+\tau$)-th iteration is higher than those at the ς-th 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 [47]; 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; termination occurs in LM-CROSS-(5) and LM-CROSS-(10), when the error for the validation data increases.

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 2 shows the mean training error and variance of training errors among the 30 runs of the algorithms on computing 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 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.

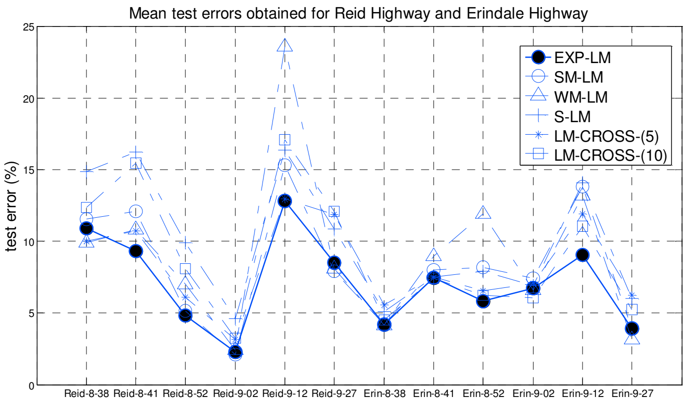


Fig. 3 Mean test errors obtained for Reid Highway and Erindale Highway using EXP-LM, SM-LM, WM-LM, S-LM, LM-CROSS(5) and LM-CROSS(10)

To evaluate the generalization capability of the NNs developed based on the algorithms, the test data, which was not involved on training the NNs, was used. The mean test errors and variances of test errors among the 30 runs were recorded in Table 3. Also, 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 the number of first ranks obtained by the algorithms are shown in the last four rows of the table. It can be found from Table 3 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 9 first ranks out of 12 tests, while SM-LM can achieve only 2 first ranks out of 12 tests and WM-LM can achieve only 1 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). Also, the average variances of test errors obtained by EXP-LM are the lowest among all algorithms. These results show that EXP-LM can produce a more stable generalization capability than the other algorithms. Also, Fig. 3 shows the mean test errors obtained by all algorithms. It shows, in general, the mean test errors obtained by EXP-LM are the smallest.

Table 2 Training error obtained for Reid Highway based on EXP-LM, SM-LM, WM-LM, S-LM, LM-CROSS-(5) , 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

Table 3 Test error obtained for Reid Highway and Erindale Highway based on EXP-LM, SM-LM, WM-LM, S-LM, LM-CROSS-(5) , 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

In addition, the *t*-test [56] 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 4. 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:

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

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; 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 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 10 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 outperform significantly on all data sets, EXP-LM in general offered much better performance than did all the other algorithms.

Table 4 T-values between EXP-LM to the other algorithms (SM-LM, WM-LM, S-LM, LM-CROSS-(5), LM-CROSS-(10)) for Reid Hwy and Erindale Hwy

		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.

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, while the fitting capability for the collected traffic flow data obtained by EXP-LM is generally poor in training the neural networks 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.

2) Incorporation with advanced NN configurations

To further evaluate the effectiveness of using exponential smoothing method in pre-processing 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 neural network, namely WNN-LM** [53], which combines the mechanisms of feed-forward NNs and the wavelet theory [9, 34]. In WNN-LM, a wavelet function is used as the transfer function in the NNs, which provides a multi-resolution approximation for the discriminate functions. Based on WNN-LM, better performance in function learning [52], including short-term traffic flow forecasting [23], can be obtained than those of the conventional feed-forward NNs.
2. **A Bayesian neural network, namely BNN-LM** [33], 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, in order 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 [54].

Table 5 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

Table 6 Test error obtained for EXP-WNN, WNN, EXP-BNN and BNN

			EXP-WNN	WNN	EXP-BNN	BNN
NN_6^{Reid}	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

Based on the configurations of BNN-LM and WNN-LM, the original collected traffic flow data was 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 Table 5 and Table 6, 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 was based on the pre-processed 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.B.1.

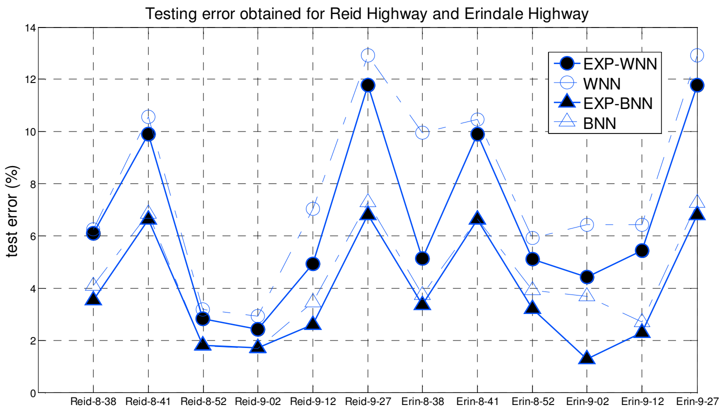


Fig. 4 Mean test errors obtained for Reid Highway and Erindale Highway based on EXP-WNN, WNN, EXP-BNN and BNN

To illustrate the capability of fitting the traffic flow data, Table 5 shows that the average of mean training errors and average of ranks of mean training errors obtained by EXP-WNN-LM are generally higher than those obtained by WNN-LM, while 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 6 shows that EXP-WNN-LM yields smaller mean test errors and better ranks of mean test errors compared with those obtained by WNN-LM, while those obtained by EXP-BNN-LM are better compared with those obtained by BNN-LM. Also, 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 7 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 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. Also, 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 is pre-processed by the exponential smoothing method. These results are not presented here due to page limitation.

Table 7 T-values between EXP-WNN to WNN, and between EXP-BNN to BNN

		WNN	BNN
NN_6^{Reid}	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**
$NN_6^{Erindal}$	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.

C. Further evaluations

In Section IV.B, the NNs were developed based on traffic flow data collected between 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 was 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 minutes to drive along Mitchell Freeway from Reid Highway to Hutton Street.

30 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-hour 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, located between the off-ramp and on ramp of Hutton Street, and located near the on-ramp of Hutton Street, respectively.

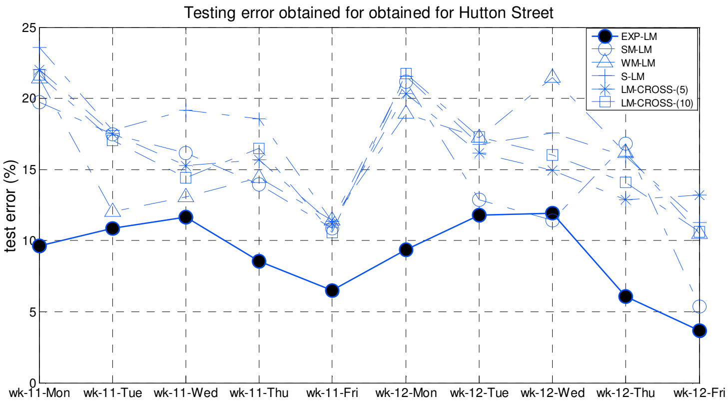


Fig. 5 Mean test errors obtained for Hutton Street for Weeks 11 and 12

The five NNs, NN_{Mon}^{Hutton}, NN_{Tue}^{Hutton}, NN_{Wed}^{Hutton}, NN_{Thu}^{Hutton} and NN_{Fri}^{Hutton}, were developed for Monday, 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, collected from Week 11 and Week 12, were used as the test data sets to evaluate the generalization capability of the NNs.

Table 8 Training error obtained for Hutton Street based on EXP-LM, SM-LM, WM-LM, S-LM, LM-CROSS-(5), LM-CROSS-(10)

		EXP-LM	SM-LM	WM-LM	S-LM	LM-CROSS -(5)	LM-CROSS -(10)
NN_{Mon}^{Hutton}	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_{Tue}^{Hutton}	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_{Wed}^{Hutton}	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_{Thu}^{Hutton}	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_{Fri}^{Hutton}	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

Table 9 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), LM-CROSS-(10)

			EXP-LM	SM-LM	WM-LM	S-LM	LM-CROSS -(5)	LM-CROSS -(10)
Week 11	NN_{Mon}^{Hutton}	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_{Tue}^{Hutton}	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_{Wed}^{Hutton}	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_{Thu}^{Hutton}	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_{Fri}^{Hutton}	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_{Mon}^{Hutton}	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_{Tue}^{Hutton}	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_{Wed}^{Hutton}	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_{Thu}^{Hutton}	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_{Fri}^{Hutton}	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 10 T-values between EXP-LM to the other algorithms (SM-LM, WM-LM, S-LM, LM-CROSS-(5), LM-CROSS-(10)) for Hutton Street

		SM-LM	WM-LM	S-LM	LM-CROSS-(5)	LM-CROSS-(10)
Week 11	NN_{Mon}^{Hutton}	**6.15**	**7.75**	**8.71**	**7.76**	**8.36**
	NN_{Tue}^{Hutton}	**4.39**	0.86	**7.49**	**6.26**	**5.67**
	NN_{Wed}^{Hutton}	**3.09**	0.92	**6.66**	**2.59**	**2.18**
	NN_{Thu}^{Hutton}	**2.89**	**3.19**	**4.03**	**3.38**	**3.91**
	NN_{Fri}^{Hutton}	**4.51**	**5.29**	**7.15**	**7.46**	**5.52**
Week 12	NN_{Mon}^{Hutton}	**7.13**	**6.45**	**7.75**	**6.86**	**8.26**
	NN_{Tue}^{Hutton}	1.25	**6.15**	**5.22**	**3.58**	**4.50**
	NN_{Wed}^{Hutton}	0.53	-2.10	**4.82**	**2.24**	**3.12**
	NN_{Thu}^{Hutton}	**5.64**	**5.21**	**3.95**	**3.08**	**3.80**
	NN_{Fri}^{Hutton}	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.

Table 8 shows that the training errors obtained by EXP-LM are generally larger than those obtained by the other algorithms, SM-LM, WM-LM, S-LM, LM-CROSS-(5) and LM-CROSS-(10). Table 9 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. Also, Fig. 5 illustrates 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 10 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 Levenberg-Marquardt algorithm, namely EXP-LM, is proposed to train NNs for short-term traffic flow forecasting. EXP-LM is 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 is 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 non-recurrent congestion conditions, to further evaluate NN models trained using the EXP-LM algorithm; 2) the proposed method will be applied to pre-process travel time data or congestion data which contain lumpiness. The pre-processed data will be applied, in order to develop travel time predictors [57, 59] or congestion predictors [58], 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] G.Ö. Ali and Y.T. Chen, "Design quality and robustness with neural networks", *IEEE Transactions on Neural Networks*, vol. 10, no. 6, pp. 1518-1527, 1999.

[2] S. Amari, N. Murata, K. Muller, M. Finke and H.H. Yang, "Asymptotic statistical theory of overtraining and cross validation", *IEEE Transactions Neural Networks*, vol. 8, no. 5, pp. 958-996, 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". *International Journal of Forecasting,* vol. 13, pp. 1-12, 1997.

[4] R.G. Brown and R.F. Meyer (1961), "The fundamental theorem of exponential smoothing", *Operations Research*, vol. 9*,* pp. 673 -685, 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 Systems with Applications*, vol. 36, 6164-6173, 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 Systems with Applications*, vol. 32, pp. 86-96, 2007.

[7] S.C. Chang, R.S. Kim, S.J. Kim and B.H. Ahn, "Traffic flow forecasting using a 3-stage model", *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 451-456, 2000.

[8] H.C. Co and R. Boosarawongse, "Forecasting Thailand's rice export: statistical techniques vs. artificial neural networks", *Computers and Industrial Engineering*, vol. 53, pp.610-627, 2007.

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

[10] G.A. Davis, and N.L. Nihan, "Nonparametric regression and short-term freeway traffic forecasting", *Journal of Transportation Engineering*, vol. 177, no. 2, pp. 178–188, 1991.

[11] S. De Lurgio, *Forecasting principles and applications*. McGraw Hill, Inc, New York, 1998.

[12] H. Dia, "An object-oriented neural network approach to short-term traffic forecasting", *European Journal of Operational Research*, vol. 131, pp. 253-261, 2001.

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

[14] M. Dougherty, "A review of neural networks applied to transport", *Transportation Research Part C: Emerging Technologies*, vol. 3, no. 4, pp. 247-260, 1995.

[15] M.S. Dougherty, H.R. Kirby, D. Boyce, "Using neural networks to recognize, predict and model traffic", *Artificial Intelligence Applications to Traffic Engineering*, pp. 235-250, 1994.

[16] M.S. Dougherty and M.R. Cobbett, "Short-term inter-urban traffic forecasts using neural networks", *International Journal of Forecasting*, vol. 13, pp. 21-31, 1997.

[17] E.L. Faria, M.P. Albuquerque, J.L. Gonzalez, J.T.P. Cavalcanta, M.P. Albuquerque, "Predicting the Brazilian stock market through neural networks and adaptive exponential smoothing methods", *Expert Systems with Applications*, vol. 36, pp. 12506-12509, 2009.

[18] F.D. Foresee and M.T. Hagan, "Gauss-Newton approximation to Bayesian regularization", *Proceedings of the 1997 International Joint Conference on Neural Networks*, pp. 1930–1935, 1997.

[19] L. Fu and L.R. Rilett, "Estimation of time-dependent, stochastic route travel times using artificial neural networks", *Transportation Planning and Technology*, vol. 24, 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 and Systems*, vol. 150, pp. 331-350, 2005.

[21] R.S. Gutierrez and O.S. Adriano, S. Mukhopadhyay, "Lumpy demand forecasting using neural networks", *International Journal of Production Economics*, vol. 111, pp. 409-420, 2008.

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

[23] X. Jiang and H. Adeli, "Dynamic wavelet neural network model for traffic flow forecasting", *Journal of Transportation Engineering*, vol. 131, no. 10, pp. 771-779, 2005.

[24] C.F. Juang, "Temporal problems solved by dynamic fuzzy network based on genetic algorithm with variable-lengthchromosomes", *Fuzzy Sets and Systems*, vol. 142, pp. 199–219, 2004.

[25] N.K. Kasabov and Q. Song, "DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction", *IEEE Transactions Fuzzy Systems*, vol. 10, pp. 144-154, 2002.

[26] G.N. Karystinos and D.A. Pados, "On overfitting, generalization, and randomly expanded training sets", *IEEE Transactions on Neural Networks*, vol. 11, no. 5, pp. 1050-1057, 2000.

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

[28] Ledoux C., "An urban traffic flow model integrating neural network", *Transportation Research 5C*, 287-300, 1997.

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

[30] F. Liang, "Bayesian neural networks for nonlinear time series forecasting", *Statistics and Computing*, vol. 15, pp. 13-29, 2005.

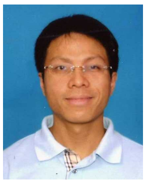
[32] Y. Liu, A.S. Janusz and Z. Zhen, "Optimized approximation algorithm in neural networks without overfitting", *IEEE Transactions on Neural Networks*, vol. 19, no. 6, pp. 983-995, 2008.

[33] D. MacKay, "A practical Bayesian framework for backpropagation networks", *Neural Computation*, vol. 4, pp. 448-472, 1992.

[34] S.G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no.7, pp. 674-693, Jul. 1989.

[35] P.A. Mastorocostas and J.B. Theocharis, "An orthogonal least-squares method for recurrent fuzzy-neural modeling", *Fuzzy Sets and Systems*, vol. 140, pp. 285-300, 2003.

[36] I. Okutani and Y.J. Stephanedes, "Dynamic prediction of traffic volume through Kalman filtering theory", *Transportation Research, Part B: Methodology,* vol. 18, no. 1, pp. 1–11, 1984.
[37] S.E. Papadakis, J.B. Theocharis, A.G. Bakirtzis, "A load curve based fuzzy modeling technique for short-term load forecasting", *Fuzzy Sets and Systems*, vol. 135, pp. 279-303, 2003.
[38] L. Prechelt, "Automatic early stopping using cross validation: quantifying the criteria", *Neural Networks*, vol. 11, no. 4, pp. 761-777, 1998.
[39] P. Ross, "Exponential filtering of traffic data", Transportation Research Record, vol. 869, *Transportation Research Board*, Washington,D.C., pp. 43–49, 1982.
[40] C. Quek, M. Pasquier and B.B.S. Lim, "POP-TRAFFIC: A Novel Fuzzy Neural Approach to Road Traffic Analysis and Prediction", *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 2, pp. 133-146, 2006.
[41] R. Setiono, "Feedforward neural network construction using cross validation", *Neural Computation*, vol. 13, no. 12, pp. 2865-2977, 2001.
[42] D. Srinivasan, C.W. Chan and P.G. Balaji, "Computational intelligence based congestion prediction for a dynamic urban street network", *Neurocomputing*, vol. 72, pp. 2710-2716, 2009.
[43] B.L. Smith, B.M. Williams and R.K. Oswald, "Comparison of parametric and nonparametric models for traffic flow forecasting", *Transportation Research Part C*, vol. 19, pp. 303-321, 2002.
[44] A. Stathopoulos, L. Dimitriou and T. Tskeris, "Fuzzy modeling approach for combined forecasting of urban traffic flow", *Computer-Aided Civil and Infrastructure Engineering*, vol. 23, pp. 521-535, 2008.
[45] 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 Transactions on Intelligent Transportation systems*, vol. 10, no. 1, pp. 60-69, 2009.
[46] 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 Transactions Industrial Electronics*, vol.50, no. 4, pp.733-748, Aug. 2003.
[47] N. Wanas, G. Auda, M. S. Kamel and F. Karray, "On the optimal number of hidden nodes in a neural network", *1998 IEEE Canadian Conference on Electrical and Computer Engineering*, vol.2, pp. 918-921, 1998.
[48] R.A. Yaffee and M. McGee, *Introduction to Time Series Analysis and Forecasting*. San Diego, CA: Academic, 2000.
[49] H. Yin, S.C. Wong, J. Xu and C.K. Wong, "Urban traffic flow prediction using a fuzzy-neural approach", *Transportation Research Part C*, vol. 10, pp. 85-98, 2002.
[50] 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 Transactions on Systems, Man and Cybernetics: Part B: Cybernetics*, vol. 36, no. 6, pp. 1342-1355, Dec. 2006.
[51] G.P. Zhang and D.M. Kline, "Quarterly time series forecasting with neural networks", *IEEE Transactions on Neural Networks*, vol. 18, no. 6, pp. 1800-1814, 2007.
[52] G.P. Zhang and M. Qi, "Neural network forecasting for seasonal and trend time series", *European Journal of Operational Research*, vol. 160, pp. 501-514, 2005.
[53] Q. Zhang and A. Benveniste, "Wavelet Networks", *IEEE Transactions on Neural Networks*, vol. 3, no. 6, pp. 889-898, 1992.
[54] W. Zheng, D.H. Lee and Q. Shi, "Short-term freeway traffic flow prediction: Bayesian combined neural network approach", *Journal of Transportation Engineering*, vol. 132, no. 2, pp. 114-121, 2006.
[55] K.Y. Chan, J. Singh, T.S. Dillon and E. Chang, Traffic flow forecasting neural networks based on exponential smoothing method, Proceeding of the 6th IEEE Conference on Industrial Electronics and Applications, pp. 376-381, 2011.
[56] G. E.-P. Box, J. S. Hunter, and W. G. Hunter, *Statistics for Experiments: Design, Innovation, and Discovery*, 2nd ed. New York: Wiley, 2005.
[57] A. Simroth and H. Zahle, "Travel time prediction using floating car data applied to logistics planning", IEEE Transactions on Intelligent Transportation Systems, vol. 12, no. 1, pp. 243-253, 2011.
[58] T. Thomas, W. Weijermars and E. Berkum, "Predictions of Urban Volumes in Single Time Series", IEEE Transactions on Intelligent Transportation Systems, vol. 11, no. 1, pp. 71-80, 2010.
[59] M. Yang, Y. Liu and Z. You, "The reliability of travel time forecasting", IEEE Transactions on Intelligent Transportation Systems, vol. 11, no. 1, pp. 162-171, 2010.
[60] M.T. Hagan and M.B. Menhaj, Training feedforward networks with the Marquardt algorithm, IEEE Transactions on Neural Networks, vol. 5, no. 6, pp. 989-993, 1994.



Kit Yan Chan received his PhD degree in Computing from London South Bank University, United Kingdom in 2006. After his PhD study, he worked as a Postdoctoral Research Fellow in the Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong, until 2009. Currently, he is a Senior Research Fellow in the Digital Ecosystems and Business Intelligence Institute, Curtin University of Technology, Australia. His research interests include computational intelligence and its applications to new product design, manufacturing process design, and traffic flow forecasting.

Tharam Dillon is a Fellow of the IEEE, ACS and IE(Aust). He is the research professor at the Digital Ecosystems and Business Intelligence Institute, Curtin University of Technology, Australia. He is head of the 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/IES Technical Committee on Industrial Informatics, and the IFIP Technical Committee 12 on Artificial Intelligence. He has published more than 750 papers published in international conferences and journals and is the author of 5 books and has another 5 edited books. His current research interests include Web semantics, ontologies, Internet computing, e-commerce, hy-brid neurosymbolic systems, neural nets, software engineering, database systems, and data mining. He is the Editor-in-Chief of the *International Journal of Computer Systems Science and Engineering* as well as the *Engineering Intelligent Systems*.

Jaipal Singh received his PhD 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 optimisation and improved quality of service. He is currently a research fellow at Curtin University working on short-term traffic forecasting, traffic visualisation and real-time congestion management. He is also developing new cyber-physical systems architectures in intelligent transportation systems.

Elizabeth Chang received a Bachelor's Degree in Computer Science in 1985 from Beijing University, China. She received her Master's Degree in Computer Science and PhD in Computer Science in 1991 and 1996 respectively from La Trobe University, Australia. Since 2006, she has been the founder and professor of Digital Ecosystems and Business Intelligence Institute, Curtin University, Australia. She has been awarded the Vice Chancellor's Outstanding Performance Award for 2005 and the Dean's Best Researcher of Year Award for 2004 and 2005. She has co-authored 3 books and has published over 350 scientific papers as book chapters, and international journals and conferences She is currently holding 6 ARC grants and a Tier 1 Centre of Excellence grant and obtained cash from ARC, industry partners as well as Research Centre of Excellence funds of over $5 million for 2002-2011. Her research interest includes: ontology and multi-agent systems, data mining for business intelligence, trust, security and risk in e-Business, XML, Web Services, P2P for collaborative environments, web engineering, IT for business and commerce, IT for health informatics, and IT for education.