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 lumpsiness 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)
algorithm (EXP-LM), is proposed to train NNs, in order to produce high generalization capability in short-term traffic flow forecasting. 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 short-term traffic 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 (D1, D2, ..., Dn), which are located between the off-ramp and on-ramp of the freeway as illustrated in Fig. 1. D1 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 \( s_i(t-kT_s) \) and past average headway \( h_i(t-kT_s) \), which was collected by \( D_i \) at time \( (t-kT_s) \) with \( k=1, 2, ..., n \) 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 \( \hat{s}_i(t+mT_s) \) passing through the \( L \)-th detector station \( D_L \) at time \( (t+mT_s) \), where future traffic flow with \( m \) sampling time ahead is forecasted.

![Fig. 1 Schematic of short-term traffic flow forecasting of the freeway](image)

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 short-term traffic flow forecasting [7, 14, 16, 22]. The NN is formulated as:

\[
\hat{s}_i(t+mT_s) = \sum_{j=1}^{M} \sum_{l=1}^{p} \left[ \beta_j^l \Psi \left( \gamma_j^l s_i(t-lT_s) + \sum_{k=1}^{K} \gamma_{ij}^k h_i(t-kT_s) \right) \right] + \alpha_0, \tag{1}
\]

where \( M \) is the number of nodes in the hidden layer; \( \alpha_0, \beta_j^l, \gamma_{ij}^k, \gamma_j^l \), and \( \Psi(\cdot) \) are the parameters of the NN, namely NN weights; and \( \Psi(\cdot) \) 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

\[
d(l) = [\theta(l), \phi(l)] \quad \text{with } l=1, 2, ..., N_D, \tag{2}
\]

where \( N_D \) is the number of collected traffic flow data for training and \( \theta(l) \) 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 \( (t(l)+mT_s) \); \( \phi(l) \) is denoted by

\[
\theta(l) = s_i(t(l)+mT_s); \tag{3}
\]

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

\[
\phi(l) = [h_i(t(l)-T_s), h_i(t(l)-2T_s), ..., h_i(t(l)-pT_s), h_i(t(l)-T_s), h_i(t(l)-2T_s), ..., h_i(t(l)-pT_s), s_i(t(l)-T_s), s_i(t(l)-2T_s), ..., s_i(t(l)-pT_s), s_i(t(l)+T_s), s_i(t(l)+2T_s), ..., s_i(t(l)+pT_s)]. \tag{4}
\]

\( h_i(t(l)) \) 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 \( l \)-th traffic flow data. Based on the collected traffic flow data \( d(l) = [\theta(l), \phi(l)] \) 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:

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

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 $\hat{\theta}(l) = \hat{s}_l(\{1\} + mT)$;

and $\hat{s}_l(\{1\} + mT)$ is determined based on equation (1) to forecast the average future traffic speed at the location of $D_k$.

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

$$w(0) = [\alpha_0(0), \beta_0(0), \beta_0(0), \gamma_0, \gamma_0, 0(0)],$$

$$w(1) = [\alpha_1(1), \beta_1(1), \beta_1(1), \gamma_1, \gamma_1, 0(1)],$$

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:

$$w(\varsigma+1) = w(\varsigma) + J^T(w(\varsigma)) J(w(\varsigma)) + \mu I)^{-1} J^T(w(\varsigma)) R$$

where $R = \left[\theta(1) - \hat{\theta}(1), \theta(2) - \hat{\theta}(2), ..., \theta(N_p) - \hat{\theta}(N_p)\right]$. The details of determination of the Jacobian matrix, $J[w(\varsigma)]$, can be referred to [60]. To forecast average headway, similar formulation can be used by replacing $\hat{s}_l(\{1\} + mT)$ with the forecast average headway, $\hat{h}_l(\{1\} + mT)$, in equation (1). Also, equation (3) is redefined by $\theta(l) = h_l(\{1\} + mT)$.

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 $d'(l) = [\theta'(l), \phi(l)]$ 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 of 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 $\geq 3$ is defined by the following equation:

$$\theta'(l) = \theta'(l-1) + \alpha(\theta(l-1) - \theta'(l-1))$$

where $\alpha$ 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 $\theta'(2) = [\theta(1) + \theta(2) + \theta(3)]/3$ respectively.

If the value of the exponential smoothing parameter $\alpha$ 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 $\alpha$ 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 $\alpha$ 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 $N_G$ is, the smaller $R^2(\alpha)$ can be obtained. The best $\alpha$ is chosen so as to produce the smallest sum of squares for the residuals which is defined as:

$$R^2(\alpha) = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2.$$
\[ R^2(\alpha) = \frac{\sum_{i=1}^{N} \left( \theta(i) - \theta(i) \right)^2}{\sum_{i=1}^{N} \left( \theta(i) - \theta(i) \right)^2} \quad (12) \]

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 \( l = 1, 2, \ldots, N_D \).

**Step 2:** Select the best exponential smoothing parameter \( \alpha \) by using the grid search where \( 0.1 < \alpha \leq 0.9 \):

**Step 2.1:** Initialize \( N_t \) exponential smoothing parameters as

\[ \alpha(i) = 0.8 \cdot \frac{N_t}{N_t - 1} - 0.1, \]

with \( i = 1, 2, \ldots, N_t \), where \( N_t \) is a constant.

**Step 2.2:** Evaluate the sum of squares for the residuals, \( R^2(\alpha(i)) \), for all \( \alpha(i) \) with \( i = 1, 2, \ldots, N_t \), based on (12).

**Step 2.3:** Determine the best exponential smoothing parameter, \( \alpha_{best} = \alpha(i) \), where \( R^2(\alpha(i)_i) < R^2(\alpha(i)) \) with \( \forall i_i, i \), but \( i \neq i_i \).

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

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

**Step 3.2:** Generate the \( i \)-th filtered traffic flow data based on equation (11), where \( i \geq 3 \).

**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, \( w(\varphi + 1) \), based on equation (9), where \( e_{state} \) and \( \delta(l) = \delta_1 l(l) + m T_l \) is determined by equations (5) and (1) respectively.

**Step 4.3:** Goto Step 2 until the termination iteration is reached or \( e_{state} \) 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 the 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.

### B. Experimental results

The EXP-LM was implemented in Matlab. Four NNs (namely \( NN^\text{Reid}_2 \), \( NN^\text{Erindale}_6 \), \( NN^\text{Erindale}_4 \) and \( NN^\text{Erindale}_6 \)) were developed to forecast short-term traffic flow regarding Reid Highway and Erindale Road. For Reid Highway, \( NN^\text{Reid}_2 \) and \( NN^\text{Reid}_6 \) 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, \( NN^\text{Erindale}_4 \) and \( NN^\text{Erindale}_6 \) 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:

\[
\theta'(l) = \frac{1}{4} (\theta(l-1) + \theta(l-2) + \theta(l-3) + \theta(l-4))
\]

with \( l \leq 4 \), 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**, uses the weighted moving average 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:

\[
\theta'(l) = \frac{4 \cdot \theta(l-1) + 3 \cdot \theta(l-2) + 2 \cdot \theta(l-3) + \theta(l-4)}{10}
\]

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-(\( \tau \))**, uses the mechanisms of cross-validation [2, 38] to avoid overtraining NNs. In LM-CROSS-(\( \tau \)), 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-(\( \tau \)) stop training the NNs, when the error for the cross-validation data at the \( \tau \)-th iteration is higher than those at the \( \zeta \)-th iteration. LM-CROSS-(5) and LM-CROSS-(10) were implemented. As LM-CROSS-(\( \tau \)) is a commonly used method for avoiding overfitting, the results obtained by LM-CROSS-(\( \tau \)) 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 \( \log_e(480) = 9 \), in which the number of training data, \( N_p \), is 480 and \( \log_e(N_p) \) 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 \( \varepsilon_{\text{max}} \) 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 \( NN_{Reid}^{6e} \) and \( NN_{Erindale}^{6e} \) 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.
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.

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), 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 algorithms with 95% confidence level. The t-value can be determined by:

\[
t-value = \frac{\mu_{EXP-LM} - \mu_{other}}{\sigma_{\mu} / \sqrt{N} + \sigma_{\mu} / \sqrt{N}}
\]

where \(\mu_{EXP-LM}\) is the mean test error of the NNs trained by the EXP-LM and \(\mu_{other}\) is the one for the other compared algorithm; \(\sigma_{\mu}\) is the variance of test errors of the NNs trained by the EXP-LM and \(\sigma_{\mu}\) is the one for the other compared algorithm; \(N_{\text{EXP-LM}}\) and \(N_{\text{other}}\) 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 better performance than did all 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 (\(N_{\text{S-LM}}\) for Reid-2008-38, Reid-2008-41, Reid-2008-51, Reid-2009-02, Reid-2009-12 and Reid-2009-27, as well as \(N_{\text{EXP-LM}}\) 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.

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 \(N_{\text{Exp-LM}}\) and \(N_{\text{other}}\), 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

| Table 2 Training error obtained for Reid Highway based on EXP-LM, SM-LM, WM-LM, S-LM, LM-CROSS-(5), LM-CROSS-(10) |
|-----------------|-----------------|-----------------|-----------------|-----------------|
| Erindale-2008-41 | 1.79 0.69 0.96 2.73 2.58 4.21 | 0.31 0.53 0.64 0.32 1.25 0.59 |
| Reindale-2008-52 | 7.93 4.31 3.71 2.31 3.81 3.42 | 0.28 0.46 0.39 0.49 0.77 |
| Erindale-2009-02 | 0.72 0.56 0.63 0.69 1.32 0.98 | 1.10 10.21 9.96 8.27 6.10 7.05 |
| Reindale-2009-12 | 6.23 8.42 7.70 7.51 6.94 6.48 | 1.22 0.84 1.86 1.48 2.83 1.89 |

| 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) |
|-----------------|-----------------|-----------------|-----------------|-----------------|
| Erindale-2008-41 | 5.55 6.83 5.19 4.79 4.04 5.48 | 0.66 1.16 1.31 0.59 1.66 0.41 |
| Reindale-2008-52 | 5.27 6.69 5.17 5.76 4.21 3.41 | 0.72 0.96 0.84 0.74 0.82 0.73 |
| Erindale-2009-02 | 4.93 4.39 4.26 3.80 4.01 3.93 | 0.28 0.59 0.34 0.28 0.83 0.39 |
| Reindale-2009-12 | 12.21 11.65 9.66 11.09 9.54 10.07 | 0.97 0.90 0.80 0.53 0.70 0.38 |
| Erindale-2009-27 | 12.51 9.48 7.44 8.86 8.03 6.64 | 1.12 1.06 1.05 0.88 0.87 |

The results obtained are in bold font, when they achieve the smallest test errors among the others.
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 feature learning [52], including short-term traffic flow forecasting [23], can be obtained than those of the conventional feed-forward NNs.

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

<table>
<thead>
<tr>
<th></th>
<th>EXP-WNN</th>
<th>WNN</th>
<th>EXP-BNN</th>
<th>BNN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Var</td>
<td>Mean</td>
<td>Var</td>
</tr>
<tr>
<td>Red-2008-08</td>
<td>14.12</td>
<td>6.26</td>
<td>14.73</td>
<td>13.64</td>
</tr>
<tr>
<td>Red-2008-12</td>
<td>14.12</td>
<td>6.26</td>
<td>14.73</td>
<td>13.64</td>
</tr>
<tr>
<td>Red-2009-08</td>
<td>14.12</td>
<td>6.26</td>
<td>14.73</td>
<td>13.64</td>
</tr>
<tr>
<td>Red-2009-12</td>
<td>14.12</td>
<td>6.26</td>
<td>14.73</td>
<td>13.64</td>
</tr>
<tr>
<td>Average</td>
<td>14.12</td>
<td>6.26</td>
<td>14.73</td>
<td>13.64</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th></th>
<th>EXP-WNN</th>
<th>WNN</th>
<th>EXP-BNN</th>
<th>BNN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Var</td>
<td>Mean</td>
<td>Var</td>
</tr>
<tr>
<td>Reid-2008-08</td>
<td>6.11</td>
<td>0.28</td>
<td>6.43</td>
<td>0.43</td>
</tr>
<tr>
<td>Reid-2008-12</td>
<td>6.11</td>
<td>0.28</td>
<td>6.43</td>
<td>0.43</td>
</tr>
<tr>
<td>Reid-2009-08</td>
<td>6.11</td>
<td>0.28</td>
<td>6.43</td>
<td>0.43</td>
</tr>
<tr>
<td>Reid-2009-12</td>
<td>6.11</td>
<td>0.28</td>
<td>6.43</td>
<td>0.43</td>
</tr>
<tr>
<td>Average</td>
<td>6.11</td>
<td>0.28</td>
<td>6.43</td>
<td>0.43</td>
</tr>
</tbody>
</table>

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 $\mathbb{N}_6^{\text{Red}}$ and $\mathbb{N}_6^{\text{Erin}}$ 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 and BNN are the same as those used in the EXP-LM, as shown in Section IV.B.1.
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

<table>
<thead>
<tr>
<th>T-values</th>
<th>EXP-WNN</th>
<th>WNN</th>
<th>BNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>0.86</td>
<td>2.91</td>
<td></td>
</tr>
<tr>
<td>var</td>
<td>4.53</td>
<td>6.63</td>
<td></td>
</tr>
<tr>
<td>mean</td>
<td>1.86</td>
<td>2.91</td>
<td></td>
</tr>
<tr>
<td>var</td>
<td>4.53</td>
<td>6.63</td>
<td></td>
</tr>
<tr>
<td>mean</td>
<td>0.86</td>
<td>2.91</td>
<td></td>
</tr>
<tr>
<td>var</td>
<td>4.53</td>
<td>6.63</td>
<td></td>
</tr>
</tbody>
</table>

Number of data sets that EXP-WNN can obtain significantly better results: 9

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.


<table>
<thead>
<tr>
<th>Week 11</th>
<th>EXP-LM</th>
<th>SM-LM</th>
<th>LM-CROSS-(5)</th>
<th>LM-CROSS-(10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>0.23</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
</tr>
<tr>
<td>var</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
</tr>
<tr>
<td>mean</td>
<td>0.23</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
</tr>
<tr>
<td>var</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
</tr>
<tr>
<td>mean</td>
<td>0.23</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
</tr>
<tr>
<td>var</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
</tr>
</tbody>
</table>

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)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>0.58</td>
<td>0.54</td>
<td>0.54</td>
<td>0.54</td>
<td>0.54</td>
<td>0.54</td>
</tr>
<tr>
<td>var</td>
<td>0.54</td>
<td>0.54</td>
<td>0.54</td>
<td>0.54</td>
<td>0.54</td>
<td>0.54</td>
</tr>
<tr>
<td>mean</td>
<td>0.58</td>
<td>0.54</td>
<td>0.54</td>
<td>0.54</td>
<td>0.54</td>
<td>0.54</td>
</tr>
<tr>
<td>var</td>
<td>0.54</td>
<td>0.54</td>
<td>0.54</td>
<td>0.54</td>
<td>0.54</td>
<td>0.54</td>
</tr>
<tr>
<td>mean</td>
<td>0.58</td>
<td>0.54</td>
<td>0.54</td>
<td>0.54</td>
<td>0.54</td>
<td>0.54</td>
</tr>
<tr>
<td>var</td>
<td>0.54</td>
<td>0.54</td>
<td>0.54</td>
<td>0.54</td>
<td>0.54</td>
<td>0.54</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>0.31</td>
<td>0.31</td>
<td>0.31</td>
<td>0.31</td>
<td>0.31</td>
<td>0.31</td>
</tr>
<tr>
<td>var</td>
<td>0.31</td>
<td>0.31</td>
<td>0.31</td>
<td>0.31</td>
<td>0.31</td>
<td>0.31</td>
</tr>
<tr>
<td>mean</td>
<td>0.31</td>
<td>0.31</td>
<td>0.31</td>
<td>0.31</td>
<td>0.31</td>
<td>0.31</td>
</tr>
<tr>
<td>var</td>
<td>0.31</td>
<td>0.31</td>
<td>0.31</td>
<td>0.31</td>
<td>0.31</td>
<td>0.31</td>
</tr>
<tr>
<td>mean</td>
<td>0.31</td>
<td>0.31</td>
<td>0.31</td>
<td>0.31</td>
<td>0.31</td>
<td>0.31</td>
</tr>
<tr>
<td>var</td>
<td>0.31</td>
<td>0.31</td>
<td>0.31</td>
<td>0.31</td>
<td>0.31</td>
<td>0.31</td>
</tr>
</tbody>
</table>

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

The five NNs, NN_Hutton Mon, NN_Hutton Tue, NN_Hutton Wed, NN_Hutton Thu and NN_Hutton Fri, 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 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


IEEE Transactions on Intelligent Transportation Systems <


