Transportmetrica
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/ttra20

Short-term prediction of traffic dynamics with real-time recurrent learning algorithms

Jiuh-Biing Sheu [a], Lawrence W. Lan [b] & Yi-San Huang [a]

[a] Institute of Traffic and Transportation, National Chiao Tung University, Taiwan

[b] Department of Global Marketing and Logistics, MingDao University, Taiwan

Published online: 17 Mar 2009.

To cite this article: Jiuh-Biing Sheu, Lawrence W. Lan & Yi-San Huang (2009) Short-term prediction of traffic dynamics with real-time recurrent learning algorithms, Transportmetrica, 5:1, 59-83, DOI: 10.1080/18128600802591681

To link to this article: http://dx.doi.org/10.1080/18128600802591681

Transportmetrica
Vol. 5, No. 1, January 2009, 59–83

Short-term prediction of traffic dynamics with real-time recurrent learning algorithms

Jiuh-Biing Sheu[a], Lawrence W. Lan[b] and Yi-San Huang[a]*

[a]*Institute of Traffic and Transportation, National Chiao Tung University, Taiwan;*
[b]*Department of Global Marketing and Logistics, MingDao University, Taiwan*

(*Received 9 October 2007; final version received 18 June 2008*)

Short-term prediction of dynamic traffic states remains critical in the field of advanced traffic management systems and related areas. In this article, a novel real-time recurrent learning (RTRL) algorithm is proposed to address the above issue. We dabble in comparing pair predictability of linear method *versus* RTRL algorithms and simple non-linear method *versus* RTRL algorithms individually using a first-order autoregressive time-series AR(1) and a deterministic function. A field study tested with flow, speed and occupancy series data collected directly from dual-loop detectors on a freeway is conducted. The numerical results reveal that the performance of RTRL algorithms in predicting short-term traffic dynamics is satisfactorily accepted. Furthermore, it is found that the dynamics of short-term traffic states characterised in different time intervals, collected in diverse time lags and times of day may have significant effects on the prediction accuracy of the proposed algorithms.

Keywords: real-time recurrent learning; traffic dynamics; stochastic; deterministic

1. Introduction

Accurately characterising and predicting the traffic dynamics or travel time, especially measured in short-time intervals, has become a prerequisite in the development of advanced traffic management/transport information systems (ATMSs/ATISs) (Lam *et al.* 2005). Here, traffic dynamics (or termed as traffic time series) are regarded as temporal evolution of traffic states, such as flow, speed and occupancy, measured in a sequential (chronological) order with identical time intervals. Numerous adaptive intelligent signal control mechanisms, for instance, are established on the basis of instantaneous or predicted 5-min or shorter flow data. Smart incident detection may require 1-min or shorter traffic states as inputs. Lam *et al.* (2002) pointed out that the short-term traffic forecasting results can be used for validation of the regional and territory-wide transport models required in various transport studies, such as the freight transport study and parking demand study, and the development of traffic flow simulator to provide the off-line short-term travel time and traffic flow forecasting database. Due to the complex nature of traffic time series with considerable fluctuations and noises, accurately capturing and predicting short-term traffic dynamics is more challenging than the long-term (e.g. hourly or daily traffic) dynamics wherein conspicuous fluctuations have essentially been

*Corresponding author. Email: goodgood31@yahoo.com.tw

ISSN 1812–8602 print/ISSN 1944–0987 online

DOI: 10.1080/18128600802591681
http://www.informaworld.com

smoothed out. Lan *et al.* (2008a) revealed that different non-linear traffic patterns could emerge depending on observed time scale, history data and time-of-day. In view of traffic dynamics measured in different ways would provide more informative insights into its complex nature, developing the prediction models to better elucidate its evolution, measured in different time intervals, periods, lags and times of day, deserves in-depth exploration. And this motivates our study.

Techniques for predicting time series can be generally divided into two categories: linear and non-linear. Linear techniques, such as autoregressive integrated moving average (ARIMA) methods, aim to characterise homogeneous time-series data, either stationary, or non-stationary but can be further transformed into a stationary series (Kalman 1960, Granger and Newbold 1976, Oller 1985). Additional comparisons between linear technique (ARIMA) and other predicting methods, for instance Kalman filtering, neural network (NN), non-parametric regression (NPR) and Gaussian maximum likelihood (GML) are also conducted for extensive applications (Smith *et al.* 2002, Tam *et al.* 2004, Lam *et al.* 2006, Lee *et al.* 2006). In contrast, linear models may not be applicable in characterising inhomogeneous data due to their weakness in transforming the non-stationarity of traffic states into stationarity.

The non-linear techniques for predicting the inhomogeneous time series are in effect strongly based on the underlying postulation that different time series with equal states may have equal futures and similar states will also evolve similarly, at least in the short run. According to such postulation, Iokibe *et al.* (1995) proposed a fuzzy local reconstruction method, which was satisfactory in prediction of some experimental non-linear time series cases. Sakawa *et al.* (1998) proposed a fuzzy neighbourhood method, which proved effective in some deterministic non-linear predictions. Lan and Lin (2001) proposed a phase-space local approximation method for satisfactorily predicting short-interval flow dynamics. Afterward, Lan *et al.* (2004) presented a confined space fuzzy proportion model originating from the improved phase-space local approximation method. However, in the prediction literature, most successful modelling for non-linear time series data have been generated in laboratory experiments and rarely found outside the laboratory due to the complex fluctuations with noises of most real-time traffic series data. This has stimulated some attempts to combine non-linearity and stochasticity in modelling and making predictions (Gardiner 1997, Ragwitz and Kantz 2000, 2002).

Considerable literature has elaborated on the predicting approaches from NNs (Clark *et al.* 1993, Dochy *et al.* 1996, Dougherty and Cobbett 1997, Smith and Demetsky 1997, Kirby *et al.* 1997) to wavelet analysis (He and Ma 2002) and to hybrid method (Li 2002, Soltani 2002). Tam and Lam (2008) also presented a real-time solution algorithm to estimate the current travel time by integrating both the on-line travel time data and off-line travel time estimates. One of the neural network based approaches, real-time recurrent learning (RTRL), is noteworthy because it is not only able to manipulate the mapping of single input–output, i.e. static process, but also capable of incorporating time sequential order into operating the non-stationary process, in which the chronological order is a very important factor to accurately predict traffic dynamics (Haykin 1999, Chang *et al.* 2002). Because of the recurrent feedback loops, a recurrent neural network (RNN) is able to process temporal patterns and time-vary systems (Chang and Mak 1999). Wherein, the real-time recurrent learning algorithm applied to train RNN developed by Williams and Zipser (1989) is one of the successful learning algorithms. In particular, it is suitable for on-line training of RNN (Mak *et al.* 1999). Afterward, Mak *et al.* (1999), Chang and Mak (1999) and Goh and Mandic (2003) proposed modified learning algorithms as well as an adaptive gradient computation to improve the convergence capability of the RTRL algorithms, which have made RTRL algorithms more useful and practical in prediction. Despite the improvement in computation of algorithms, most previous literature employing the RTRL algorithms may either lack empirical analysis of the characteristics of traffic dynamics before prediction, or hastily train a network without considering the effects that influence the prediction accuracy from diverse perspectives.

Accordingly, the main purpose of this study is to propose RTRL-based algorithms to know whether or not the dynamics of short-term traffic states characterised in different time intervals, collected in diverse time lags and times of day have a significant influence on the performance of the proposed model relative to the published forecasting methods. In addition to assessing the relative performance of the proposed RTRL algorithms, we further compare the pair predictability of linear method *versus* RTRL algorithms and simple non-linear method *versus* RTRL algorithms individually using a first-order autoregressive time series AR(1) and a deterministic function to elucidate the significance that the characteristics of traffic dynamics affect the accuracy of prediction. After a well-trained network is built and various techniques are compared, the accurate understanding for traffic dynamics and reliable prediction would be anticipated.

The remaining parts of this article are organised as follows: Section 2 introduces the rationales for linear, simple non-linear algorithms, RTRL and radial basis function neural network (RBFNN) methodologies. Section 3 presents a preliminary test to compare the pair predictability between linear, simple non-linear algorithms and the proposed RTRL algorithms. Section 4 further progresses in a sensitivity analysis with various time intervals, time lags and times of day. Finally, extensive discussions are elucidated in Section 5, and concluding remarks are addressed in Section 6.

2. Methodology

This section mainly describes the methods of this study, including linear, simple non-linear algorithms, RBFNN and RTRL models. In addition, according to the comments from Tsekeris (2006) and the responses from Lin *et al.* (2006), therein the benefits and limitations of approaches are also presented in this section.

2.1. *Linear and simple non-linear algorithms*

Now, let us start out from the fundamentals of linear and non-linear algorithms. Let $x(t)$ denote the studied traffic series describing the time evolution in state space. Then it can be further expressed in a mapping form given by $x_{n+1} = f(x_n)$, $n \in Z$, in discrete time $t = n\Delta t$, where x is a state vector that is finite dimensional $x \in R^n$, and f is referred to as vector fields explicitly depending on n and t. Thus, a traffic series can be considered as a sequence of observations $\{S_t = s(x_n)\}$ performed with some measurement functions $s(\cdot)$, inferring that a one-dimensional series can be embedded into multiple-dimensional spaces denoted by $S_t = (\zeta_{t-(m-1)\tau}, \zeta_{t-(m-2)\tau}, \ldots \zeta_{t-\tau}, \zeta_t)$, $t = 1, 2, \ldots, N$, where the parameter τ is the time lag and the integer m is the dimension (Lan *et al.* 2008a).

Given a sequence of observations S_t, $t = 1, \ldots, N$, we intend to predict the outcome of the following measurements, S_{t+1}. One often wants to find the prediction $\hat{S}_{t+1}$, which minimises the expectation value of the squared prediction error $\langle(\hat{S}_{t+1} - S_{t+1})^2\rangle$. When we assume the time series is stationary, we can estimate this expectation value by its average over the available measured values. If we further restrict the minimisation to linear time-series models, which incorporate the k last measurements, we can express this by

$$\hat{S}_{t+1} = \sum_{j=1}^{k} a_j S_{t-k+j} \tag{1}$$

and minimise

$$\sum_{t=k}^{N-1} \left(\hat{S}_{t+1} - S_{t+1}\right)^2 \tag{2}$$

with respect to the parameters a_j, $j = 1, \ldots, k$. Here, we have assumed that the mean of the time series has already been subtracted from the measurements. By requiring that the derivatives with respect to all the a_js to be zero, we obtain the solution by solving the linear set of equations

$$\sum_{j=1}^{k} C_{ij} a_j = \sum_{t=k}^{N-1} S_{t+1} S_{t-k+i}, \quad i = 1, \ldots, k \tag{3}$$

Here, C_{ij} is the $k \times k$ auto-covariance matrix

$$C_{ij} = \sum_{t=k}^{N-1} S_{t-k+i} S_{t-k+j} \tag{4}$$

Note that the linear relation, Equation (1), is justified for harmonic as well as for linear stochastic processes. The most popular stochastic models for linear time series, autoregressive (AR) models and moving average (MA) models, either consisted of linear filters acting on a series of independent noise inputs as expressed in Equation (5) or on past values of the signal itself as expressed in Equation (6), while Equation (7) represents the ARMA model (Chatfield 1996)

$$x_n = \sum_{j=0}^{M_{MA}} b_j \eta_{n-j} \tag{5}$$

$$x_n = \sum_{j=1}^{M_{AR}} a_j x_{n-j} + \eta_n \tag{6}$$

$$x_n = a_0 + \sum_{i=1}^{M_{AR}} a_i x_{n-i} + \sum_{j=0}^{M_{MA}} b_j \eta_{n-j} \tag{7}$$

where x_n is a Gaussian random variable, a_j, b_j are parameters. M_{MA}, M_{AR} are the order of MA model and AR model and η_n is white Gaussian noise.

Nevertheless, most time series of traffic dynamics exhibited in the real world are non-linear and more complex than the time series formulated by linear models. A local linear method in multi-dimensional spaces was employed to predict non-linear time series if the database was large and the noise level was small (Kantz and Schreiber 2004). The original concept relevant to non-linear prediction was used in tests for determinism by Kennel and Isabelle (1992). The resulting method is very simple. Recall the time-series expression in multi-dimensional spaces: $S_t = (\zeta_{t-(m-1)\tau}, \zeta_{t-(m-2)\tau}, \ldots, \zeta_{t-\tau}, \zeta_t)$, $t = 1, \ldots, N$, and for all measurements $S_1, \ldots, S_t$, the corresponding delay vectors $(\zeta_{1-(m-1)\tau}, \ldots, \zeta_1)$, $\ldots, (\zeta_{t-(m-1)\tau}, \ldots, \zeta_t)$ in multi-dimensional spaces can be found. In order to predict a future measurement S_{t+T}, one can find the embedding vector ζ_{t0} closest to ζ_t and use ζ_{t0+T} as a predictor. However, owing to multiple dimensions, we have to choose the parameter ε of the order of the resolution of the measurements and form a neighbourhood $\Psi_\varepsilon(\zeta_t)$ of radius ε around the point ζ_t. For all points $\zeta_{t0} \in \Psi_\varepsilon(\zeta_t)$, i.e. all points closer than ε to ζ_t, look up the individual predictors ζ_{t0+T}. The prediction is then the average of all these individual predictors.

$$\hat{S}_{t+T} = \frac{1}{\left|\Psi_\varepsilon(\zeta_t)\right|} \sum_{\zeta_{t0} \in \Psi_\varepsilon(\zeta_t)} \zeta_{t0+T} \tag{8}$$

Here, $\left|\Psi_\varepsilon(\zeta_t)\right|$ denotes the number of elements of the neighbourhood $\Psi_\varepsilon(\zeta_t)$. If no neighbours closer than ε can be found, one might just increase the value of ε until some neighbours are found. The calculation of average of individual predictors can be adjusted to raise the accuracy of prediction by adopting the nearest-neighbour algorithms (Smith *et al.* 2002).

The concept of multi-dimensional spaces and simple non-linear prediction for any traffic series is demonstrated in Figure 1. In the left panel (a), the red, blue and black circles respectively represent the values of the series at time t, $t+\tau$ and $t+2\tau$; in contrast, in the middle panel (b), one can find their corresponding trajectories in the multi-dimensional spaces through reconstruction. If the time delay τ is different, the portrait in multi-dimensional spaces will change immediately. For small τ, s_t and $s_{t+\tau}$ are very close to each other. For large values of τ, s_t and $s_{t+\tau}$ can be completely independent of each other, and any connection between them is random. When a proper time delay is determined, we can map the traffic series from one dimension into three-dimensional spaces. On the other hand, in the right panel (c), if we choose one small section part of the state trajectories in the multi-dimensional spaces and enlarge it to observe the trajectory motions, it indicates

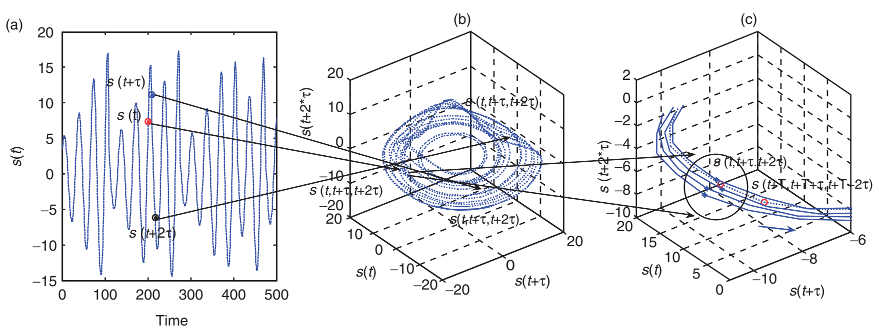

Figure 1. The concept of prediction for traffic series in multi-dimensional spaces.

that the trajectories within radius ε, i.e. in the black circle, move towards the same direction and one can predict the red circle at time $(t+T)$ by averaging the four closed points marked start at time $(t+T)$. That is, in this case we assume that the underlying relationship between the current observation and its nearest neighbours remains stationary with short-term time evolution; the points marked, their values already known, are the neighbours of the current observation (the red circle at time t) and then a prediction (the red circle at time $t+T$) can be made by using the relationship and tracking the movements of the nearest neighbours.

2.2. *Real-time recurrent learning algorithms*

Apart from the linear and simple non-linear techniques, several learning algorithms using NN theory, such as Hebbian learning (Hebb 1949), back propagation network (BPN) (Rumelhart and McClelland 1986, Werbos 1994), self-organising neural network (SOM) (Kohonen 1995) and RBFNN (Haykin 1999), have also been developed to improve the accuracy of prediction. For instance, BPN and RBFNN are two typical examples, which learn of the static non-linear relationship by mapping input–output pairs. Therein, the BPN learning begins with the feed-forward recall phase, i.e. as a single pattern vector is submitted at the input, the output of the layers is compared to the known-good output and a mean-squared error signal is calculated. The error value is then propagated backwards through the network, and small changes are made to the weights in each layer. The weight changes are calculated to reduce the error signal for the case in question. The whole process is repeated for each of the example cases, then back to the first case again, and so on. The cycle is repeated until the overall error value drops below some pre-determined threshold.

In contrast, RBFNN uses a multi-dimensional traffic series U_t as the input to the network, and outputs the corresponding output series Y_t. Theoretically, formulation of the network output response to an input multi-dimensional time series is postulated as a linear combination through the hidden layer responses, and thus can be expressed as follows:

$$y_t = w_0 + \sum_{j}^{M} w_j \cdot \phi_j\left(\left\| U_t - c_j \right\|\right), \quad j = 1, 2, \ldots, M \tag{9}$$

where $\phi(\cdot)$ is a radial basis function representing a response of the j-th hidden neuron to an input multi-dimensional time series U_t; w_j is a weight factor of the j-th hidden neuron for defining the contribution of the hidden neuron to a particular output; and w_0 is a bias term. The RBF hidden neuron responses z_j are given by

$$z_j = \phi_j\left(\left\| U_t - c_j \right\|\right) = \exp\left(-\frac{\left\| U_t - c_j \right\|}{2\sigma_j^2}\right), \quad j = 1, 2, \ldots, M \tag{10}$$

where c_j is the centre of the j-th Gaussian function and σ_j is the width of the Gaussian.

As input traffic series are presented to the RBFNN, the network iteratively creates new centre neurons to reduce its performance error (i.e. Euclidean distance). Allocation of the new hidden neurons is determined by orthogonal least squares (OLS), which employs a Gram–Schmidt algorithm and Cholesky decomposition (Chang and Chang 2005) to create new centre neurons under a given threshold. In other words, the widths and centre locations of the existing hidden neurons can be adjusted during the learning process. As to the method of adjusting the weights w_j, Broomhead and Lowe (1988) proposed a recursive least mean squares (LMS) algorithm to obtain an acceptable error as follows: if $d(p)$ is the p-th desired value, then $y(p)$ is the p-th network output. The $e(p)$ is the p-th difference between the desired value and the network output. When $e(p)$ equals zero, the p-th network output is thereby able to fit the p-th desired value entirely. As such, the total value of all $e(p)$ in the network can be a minimum:

$$E = \sum_{p=1}^{N} [e(p)]^2 = \sum_{p=1}^{N} (d(p) - y(p))^2 \tag{11}$$

When E has a minimum value, then the gradient vector $\partial E(p)/\partial w_j$ is equal to zero. Substituting $\partial E(p)/\partial w_j = 0$ into Equation (11) will obtain $W = (\phi^T \phi)^{-1} \phi^T d(p)$. The parameters w_j are iteratively updated until the learning process stably converges.

In contrast to the static learning algorithms, the real-time recurrent neural network (RTRNN) can be considered as a BPN with feedback loops connecting to every hidden node, which exhibits dynamical learning algorithms. The main difference compared with BPN is that the outputs are used as part of the next sequentially timed input, i.e. the output at time $(t+1)$ is based upon the current input and previous outputs. Furthermore, the RTRNN consists of three layers: a concatenated input–output layer with $(m+n)$ nodes, a processing (hidden) layer with n nodes and an output layer with k outputs. Let $y(t)$ denote the n-tuple of outputs of the n-processing neurons at time t and $x(t)$ the m-tuple of external inputs to the network at time t. We concatenate $y(t)$ and $x(t)$ to form the $(m+n)$-tuple $u(t)$, with B denoting the set of indices for the processing neurons and A the set of indices for the external inputs, so that

$$u_i(t) = \begin{cases} x_i(t) & \text{if} \quad i \in A \\ y_i(t) & \text{if} \quad i \in B \end{cases} \tag{12}$$

By adopting the indexing convention just described, a hidden network net_j at time t is obtained by summing up the weighted inputs with a weight matrix w. After the network is transferred by an activation function $f(\ )$, the output $y_j(t)$ is used as a feedback input in the next time step and summing up the weighted feedback inputs with a weight matrix v is repeated. Likewise, after the transformation, the network output, $z_k(t)$, is passed to an output layer. The above said procedure can be expressed as the following equations:

$$\text{net}_j(t) = \sum_{j \in A \cup B} w_{ji}(t-1) u_i(t-1) \tag{13}$$

$$y_j(t) = f(\text{net}_j(t)) \tag{14}$$

$$\text{net}_k(t) = \sum v_{kj}(t) y_j(t) \tag{15}$$

$$z_k(t) = f(\text{net}_k(t)) \tag{16}$$

With regard to the algorithms for computing the weight matrix w, v and the error function, we denote $d_k(t)$ as the desired value of the k-th neuron at time t and define $e_k(t)$ to be the difference between the desired value and the network output at time t, i.e.

$$e_k(t) = d_k(t) - z_k(t) \tag{17}$$

Then we define the error function, $E(t)$:

$$E(t) = \frac{1}{2}\sum_{k=1}^{K} e_k^2(t) \tag{18}$$

Here, the error function defined as half of the sum of the square errors is only for easy calculation on the weight adjustments. According to the steepest descent method, the amount of adjusted weight for $v_{kj}(t)$ and for $w_{mn}(t)$:

$$\Delta v_{kj}(t) = -\eta_1 \frac{\partial E(t)}{\partial v_{kj}(t)} \tag{19}$$

$$\Delta w_{mn}(t-1) = -\eta_2 \frac{\partial E(t)}{\partial w_{mn}(t-1)} \tag{20}$$

where η_1, η_2 are the learning rates.

In general, a high or low learning rate would disadvantage the training process because a higher learning rate referring to an internet with larger modified weights could quickly achieve the goal of minimising the error function, yet in contrast, the higher learning rates could lead to an over-weighted adjustment and cause an error-oscillation phenomenon. Hence, according to empirical experiments, it was decided that a higher learning rate be set up in the beginning and decreasingly adjust the values by multiplying by a constant less than 1 (e.g. 0.95) in order to take into account both the converging rate and to avoid an oscillatory situation.

And

$$\frac{\partial E(t)}{\partial v_{kj}(t)} = -e_k(t) f'(\mathrm{net}_k(t)) y_j(t) \tag{21}$$

$$\frac{\partial E(t)}{\partial w_{mn}(t-1)} = \left[\sum_{k=1}^{K} -e_k(t) f'(\mathrm{net}_k(t)) v_{kj}(t)\right] \frac{\partial y_j(t)}{\partial w_{mn}(t-1)} \tag{22}$$

According to error back propagation algorithms (Chang and Chang 2005), a new variable with three dimension can be defined as $\pi^j_{mn}(t)$, which is called a dynamic variable

$$\pi^j_{mn}(t) = \frac{\partial y_j(t)}{\partial w_{mn}(t)} \text{ for all } j \in B, \quad m \in B, \quad n \in A \cup B \tag{23}$$

Accordingly

$$\Delta w_{mn}(t-1) = \eta_2 \left[\sum e_k(t) f'(\mathrm{net}_k(t)) v_{kj}(t)\right] \pi^j_{mn}(t) \tag{24}$$

In brief, the steps involved in RTRL algorithms can be summarised as follows and depicted in Figure 2:

Step 1: Randomly initialise the weight $w_{mn}(0)$ and $v_{kj}(0)$.

Step 2: Input the $x_i(t)$ into the RTRL network and compute the $y_j(t)$, $z_k(t)$, then use $y_j(t+\tau)$ as feedback to the concatenated input–output layer together with $x_j(t+\tau)$ as new inputs.

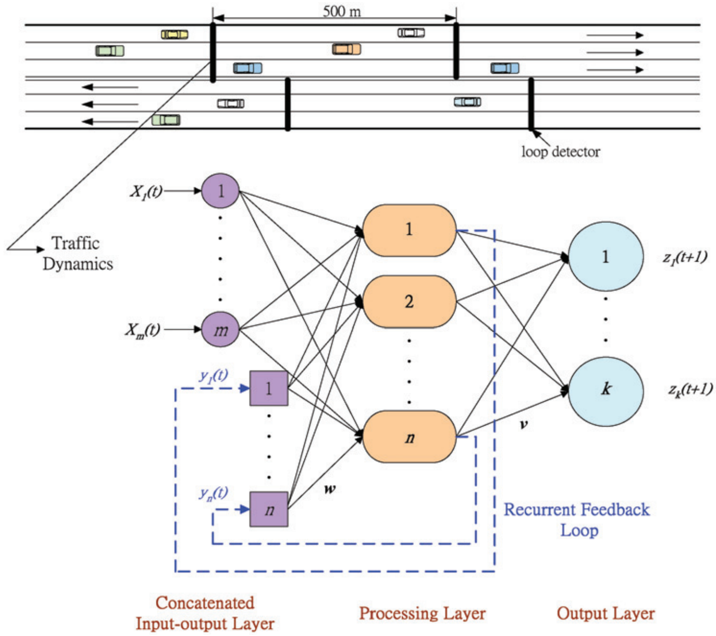

Figure 2. A typical RTRL network and traffic dynamics from loop detectors.

Step 3: Compute the difference between desired value $d_k(t)$ and network output $z_k(t)$.

Step 4: Update $\Delta v_{kj}(t)$ according to Equation (19).

Step 5: Update $\Delta w_{mn}(t-1)$ according to Equation (20).

Step 6: Increment t by 1 and go to step 2.

Note that, according to Yeh (2004), a feasible analysis encompassing several steps were suggested before one employed NN algorithms to solve problems. First, ask whether some kind of relationship exists between input and corresponding output data or not; second, consider whether or not plenty of data examples can be afforded to train the network; third, review if conventional tools cannot come up with the current requirement; and fourth, examine if similar case studies using NNs had been done successfully. Only after the above circumstances are taken into detailed account, the NN tools are recommended to solve problems and accurate prediction can be anticipated.

3. Preliminary testing

This section describes the main procedures and analytical results of the preliminary tests. Therein, we generated two time-series data categories including linear stochastic time series and non-linear deterministic time series in advance of prediction. The stochastic time series derived from a linear equation was adopted to compare the predictability between the linear autoregressive method and RTRL algorithms while the non-linear time series derived from a first-order differential-delay equation was used to compare the predictability between the simple non-linear method and RTRL algorithms. Details about the preliminary test procedures are described as follows:

First, an AR(1) time series, $(x_{t+1} - 0.4) = 0.75(x_t - 0.4) + e_t$ with $e_t \overset{i.i.d}{\sim} N(0, 1)$ was used to compare the predictability between the linear model and the RTRL algorithms. In the AR(1) linear time series, 200 independent points e_t, which conformed to Gausian distribution, were created and an initial x_t was picked to iterate 200 times together with e_t. Then, we set the order of the above model to be equal to one to compute the average prediction error and residuals for each time step. After computation, we learned that the root-mean-square error (RMSE) was equal to 1.03. Employing the same time series x_t as an input as well as x_{t+1} as output, we adopted the RTRL algorithms to train a network and calculated the RMSE, which equaled 0.979 for one trained data set and 0.93 for another test data set. In order to train the AR(1) model, in the RTRL NN, six nodes were used to process the recurrent feedbacks and the learning rate was set to 0.1. The goal of the RTRL network we set was either that the RMSE equals 0.01 or the training times reached 700 times, whereupon the training iterations would stop. In Figure 3, panel (a) represents the difference between the outputs of the AR(1) model and desired values; while panel (d) represents the difference between the outputs of the RTRL network and desired values. It is obvious to indicate that for a stochastic time series, the accuracy of prediction by adopting RTRL algorithms is superior to adopting linear prediction both from observing the difference in figure and comparing the values of RMSE.

Second, a first-order differential-delay equation, which is the famous Mackey–Glass equation: $\mathrm{d}x(t)/\mathrm{d}t = \left[0.2x(t-\tau)/1 + x^{10}(t-\tau)\right] - 0.1x(t)$, was used to compare the predictability between the simple non-linear method and the RTRL algorithms. This equation represents a physiological responsive system, which can be used as an index to examine the features of a non-linear time series (Mackey and Glass 1977). For instance, the series displays periodic motions when τ is a relatively small value, whereas for τ larger than 17, it displays a chaotic phenomenon. We employed an average mutual information (AMI) approach (Fraser and Swinney 1986) and a false nearest neighbour (FNN)

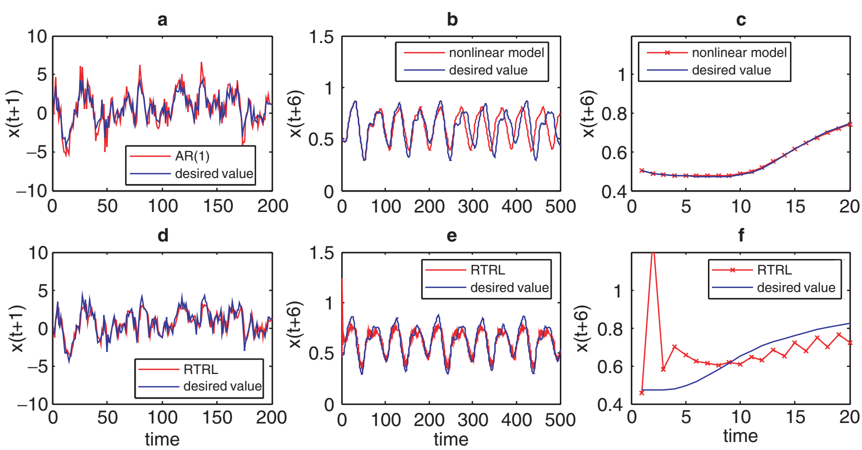

Figure 3. The difference between model output and desired values by adopting linear method, simple non-linear method and RTRL algorithms.

algorithm (Kennel *et al*. 1992) to search for the proper time delay and to determine the minimal sufficient embedding dimension. Once the appropriate time delay and the sufficient dimension were determined, we were able to map the one-dimensional differential equation into multi-dimensional spaces and make use of the neighbouring measurements in multi-dimensional spaces to predict future points. In accordance with this approach, a time series derived from Mackey–Glass equation with a time lag $\tau = 6$, which contained 500 data points, was reconstructed into multi-dimensional spaces and was forward predicted 500 time steps. Panel (b) in Figure 3 represents the predicting results showing the difference between output of the simple non-linear technique and desired values. The RMSE was equal to 0.1268. Similarly, we adopted the RTRL algorithms to train a network, in which the input and output are the same as the above time series, to predict the first-order differential-delay equation. In the RTRL network, 12 nodes were used to process the recurrent feedbacks and the learning rate was set to 0.1. The goal of the RTRL network we set was either that the RMSE equals 0.07 or the training times reached 2000 times, whereupon the training iterations would stop. Panel (e) in Figure 3 represents the prediction results, showing the difference between output of RTRL algorithms and desired values. The RMSE was equal to 0.07. From Figure 3 and RMSE, again we learned that for a deterministic equation, the accuracy of prediction by adopting RTRL algorithms is superior to adopting the simple non-linear technique.

Note that if further observing the top panel in Figure 3(b), we will find that the errors of prediction by using the simple non-linear technique are not the same as time evolves, but rather the differences are getting larger, i.e. the accuracy of prediction is getting low as time evolves. By contrast, the errors of prediction by using the RTRL algorithms do not exhibit such a situation, but show large differences during the first steps. The right panels of Figure 3 display the difference between the model output and desired values by adopting the simple non-linear technique (panel (c)) and adopting the RTRL network (panel (d)) in the first 20 time steps. It can be clearly seen that in panel (c), the curve depicting the simple non-linear technique and the curve depicting the desired values match quite closely. The RMSE of short time steps (e.g. 20 steps) is equal to 0.0035, which is greatly superior to the average RMSE of whole steps (e.g. the average RMSE of 500 time steps is equal to 0.1268). In other words, if one would like to forward predict a non-linear time series resulting from a deterministic function in short steps, then the simple non-linear technique is quite a good method to adopt. By contrast, in panel (f), the curve depicting RTRL algorithms and the curve depicting the desired values do not match well in the beginning but converge gradually. It indicates that the RTRL algorithms is suitable to train a network to predict, and eventually the average predictability compared with other techniques is satisfied; however, training time of the NN is comparably long and is a factor that should be taken into account in practice. Consequentially, in accordance with the different purposes to be achieved, the predicting techniques with their traits serve various functions. In terms of the purpose of improving the accuracy of prediction in this case study, the RTRL network is a suitable technique because the training and testing results have revealed that the method can not only successfully simulate a linear time series with stochastic characteristics, but also can capture the non-linear dynamic trajectories resulting from a deterministic function.

A further description regarding how the numbers of hidden nodes used to process the recurrent feedbacks were determined. Similar to the learning rates, the number of hidden nodes was determined by a trial and error method or from empirical experiments.

Normally, the higher the number of hidden nodes used to process the recurrent feedbacks, the slower the speed a network converges. In other words, an appropriate number of nodes used to process the recurrent feedbacks helped reduce the value of the error function, whereas a large number of nodes did not benefit the process but rather disadvantaged the convergence of a network. From the above we can infer that, when training a network, an insufficient number of hidden nodes results in the relationship between input elements and observed data not being successfully constructed, which consequently leads to a large error gap. In contrast to using an insufficient number of hidden nodes, too many nodes could cause a network with high freedom to over-fit the training sets, and, sometimes, too many nodes, involving over description with regard to noise existing in data sets, would not only delay the convergent time but also fail to train the network. However, according to Dawson and Wilby (2001), the methods for deciding on the appropriate number of hidden nodes for a network can be divided into two types: pruning algorithms (Abrahart *et al.* 1998) and constructive algorithms (Kwork and Yeung 1997). The pruning algorithms set up a loose number of hidden nodes to train a network and then the number of nodes are gradually reduced until an adequate number of nodes are found, whereas the constructive algorithms predetermine a small number of nodes and then the number of nodes are increased one by one until a threshold error value is obtained. In addition, Yeh (2004) suggested that around 6–8 hidden nodes were adequate to process networks in case of normal problems while 12–16 hidden nodes were recommended to train networks provided that complex problems were appropriately tackled. Hence, in this article, we adopted constructive algorithms to determine the number of nodes and eventually concluded that six nodes be used in the RTRL NN for training the AR(1) model and that 12 nodes be used for training the first-order differential equation.

4. Sensitivity analysis

After explaining the rationales for various prediction models and comparing the predictability of different techniques, we would like to further learn what kind of factors affect the accuracy of prediction. The following is the sensitivity analysis, where we adopt the proposed RTRL algorithms and RBFNN to train and to compare the networks by inputting real traffic-series measurements with various time intervals, time lags and times of day to explore the factors affecting the predictability. Note that the reason we implemented the sensitivity analysis using RTRL and RBFNN models is to support the fact that the accuracy of prediction will be influenced by some factors, wherein the influences are not only justified by the proposed RTRL but also RBFNN, rather than comparing the predictability between RBFNN and RTRL algorithms.

4.1. *Data*

In this section, the real traffic time-series data was directly extracted from two nearby dual-loop detectors, station 433 and station N27.9, both inbound to Taipei City, on the mainline of the Sun Yat-Sen Freeway of Taiwan. In order to investigate the training results of traffic dynamics under different situations, the raw data was divided into two groups. Raw data in the first group contains lane-base traffic series from five selected workdays, including flow, time-mean-speed and percent occupancy, which were reported once for

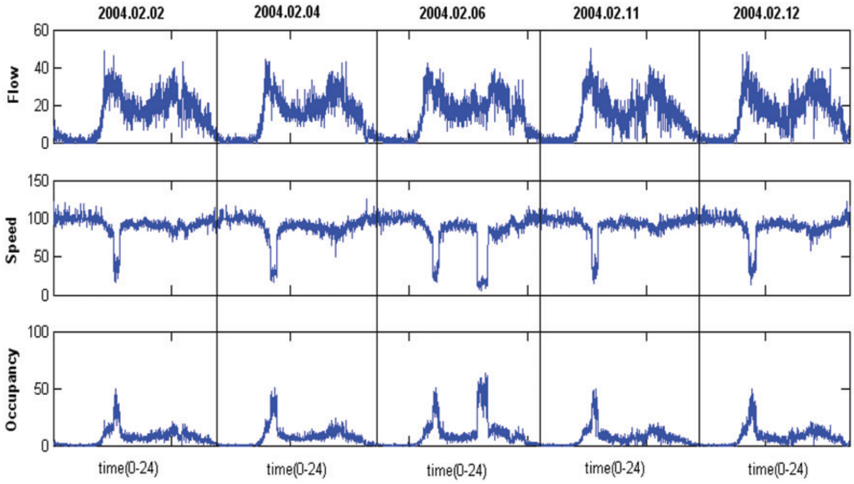

Figure 4. An illustration of 1 min traffic series for five workdays (station 433).

every 20 s by the median-lane detector at station 433. For analysis purpose, we accumulated the 20-s traffic series into 1- and 3-min lane-base series via the following calculations: flows were directly summed up from 20-s flow; speeds were the weighted average of 20-s time-mean-speed multiplied by its corresponding flow; and occupancies were the arithmetic mean of 20-s occupancy. Figure 4 demonstrates the 1-min flow, speed and occupancy series. Note that very similar traffic patterns emerge on different workdays, but they never exactly repeat.

Raw data in the second group contains approach-base flow, time-mean-speed and percent occupancy from 10 selected workdays, which were reported once for every 5 min at station N27.9. This 5-min approach-base raw data, together with a conversion into 15-min approach-base time series data, was used for analysis. The conversion of the 5-min approach-base flow, speed and occupancy series to 15-min ones follows the same calculation as that at station 433.

Apart from the above traffic time series extracted from station 433 and station N27.9, there are some noteworthy invariant features. According to Lan *et al.* (2008a) who also analysed the traffic features at the same stations, the results have indicated that the degrees of variation of traffic series depend on times of day, the early hours (00:00 am–03:00 am) having the largest coefficient of variation (CV) while the evening peak hours (18:00 pm–21:00 pm) having the smallest CV. The degrees of variation will decline with the length of measured time interval. Traffic series measured at 20-s intervals have the largest CV, followed by 1- and 3-min intervals. Furthermore, traffic time-series data at station 433 and station N27.9 have revealed different non-linear features in multi-dimensional spaces, including fixed point, periodic-like, deterministic-like, stochastic and random features. Specifically, the flow and occupancy state trajectories can shrink gradually to fixed points in the early hours. If measured at shorter intervals, such as one-minute interval at station 433, the state trajectories in multi-dimensional spaces will repel randomly like a ‘random walk’. If measured in longer intervals, such as 15-min intervals at station N27.9, deterministic-like features are likely to emerge. Apart from the above non-linear features, stochastic features of these traffic series prevail.

Since the criterion of *RMSE* is influenced on the value of data and the flow, speed and occupancy time series carry different units and cover a diverse range in this article, it is necessary to process the original data with various units before comparison of predictability. For comparison purposes, therefore, all the traffic data studied were standardised using Equation (25).

$$\tilde{x}_i = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}, \quad i = 1, 2, \ldots, N \tag{25}$$

where x_i is the i-th observed data point; $x_{\min}$ is the minimum in observed points; $x_{\max}$ is the maximum in observed points; $\tilde{x}_i$ is the i-th standardised data point. The prediction results for flow, speed and occupancy series measured in different time intervals, time lags and times of day, are detailed as follows.

4.2. *Prediction accuracy with various intervals*

Before a network is trained, it is necessary to clarify the input and output of the NN. For RBFNN, according to Lan *et al.* (2007) the traffic series measured with time lag (=1) in three-dimensional state spaces provides a satisfying training effect. Namely, the input vector is $[\tilde{x}(t-3), \tilde{x}(t-2), \tilde{x}(t-1)]$; output vector is $\tilde{x}(t)$, where $\tilde{x}(t)$ represents the standardised traffic data at time t. For RTRLNN, the input vector is $\tilde{x}(t-1)$ and the output vector is $\tilde{x}(t)$, wherein the network output at time t consists of the current input vector $\tilde{x}(t-1)$ and network outputs of the previous layer. At station 433, the number of lane-base data points to be analysed were 1440 and 480 respectively, for 1- and 3-min traffic series, thus a 24-h workday (2004.02.04) data set was selected for training and another 24-h workday (2004.02.12) data set for testing. At station N27.9, we also used 1440 and 480 approach-base data points for 5- and 15-min traffic series, respectively, thus a consecutive five-workday (2004.02.09–2004.02.13) data set was selected for training and another consecutive five-workday (2004.02.16–2004.02.20) data set for testing.

The results of prediction are summarised in Tables 1 and 2. From the tables, all of the RMSEs are sufficiently small to show that both the RTRL and RBFNN model are highly

Table 1. Prediction results of traffic series measured in different time intervals (station 433).

			RTRL-RMSE		RBF-RMSE	
Time interval	Traffic variable	Time lag τ	Train (2004.02.04)	Test (2004.02.12)	Train (2004.02.04)	Test (2004.02.12)
1 min	Flow	1	0.0943	0.1049	0.0851	0.0907
	Speed	1	0.0510	0.0600	0.0556	0.0678
	Occupancy	1	0.0566	0.0600	0.0433	0.0477
3 min	Flow	1	0.0787	0.0800	0.0593	0.0734
	Speed	1	0.0500	0.0600	0.0547	0.0555
	Occupancy	1	0.0510	0.0557	0.0392	0.0458

satisfactory in predicting the real-world short-term traffic series. Figures 5 and 6 depict the difference between network outputs and observed values by adopting RTRL algorithms, in which a portion of the data points are picked deliberately to clearly depict the differences in the lower panel. However, it is noted that in Figure 7 the curve of difference using the RTRL model oscillates up and down more significantly than the curve of difference using the RBF model for the first 15 steps or even longer period. Such oscillations are similar to our case study demonstrated in the above section. In addition,

Table 2. Prediction results of traffic series measured in different time intervals (station N27.9).

			RTRL-RMSE		RBF-RMSE	
Time interval	Traffic variable	Time lag τ	Train (2004.02.09–2004.02.13)	Test (2004.02.16–2004.02.20)	Train (2004.02.09–2004.02.13)	Test (2004.02.16–2004.02.20)
5 min	Flow	1	0.0671	0.0686	0.0819	0.0623
	Speed	1	0.0574	0.0640	0.0624	0.0587
	Occupancy	1	0.0806	0.0500	0.0494	0.0549
15 min	Flow	1	0.0663	0.0648	0.0620	0.0602
	Speed	1	0.0449	0.0574	0.0590	0.0575
	Occupancy	1	0.0755	0.0475	0.0428	0.0513

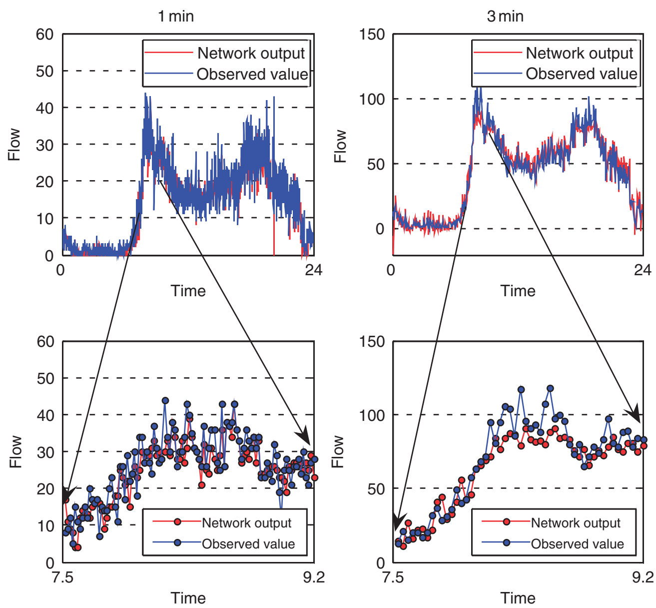

Figure 5. The RTRL network outputs and observed values of flows measured in different time intervals (station 433).

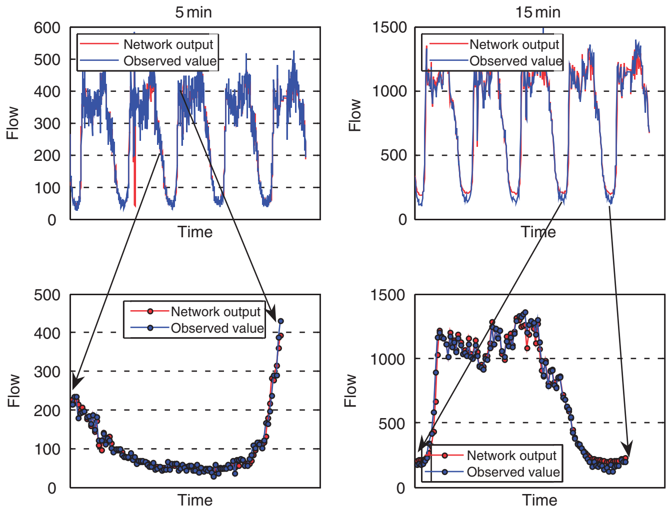

Figure 6. The RTRL network outputs and observed values of flows measured in different time intervals (station N27.9).

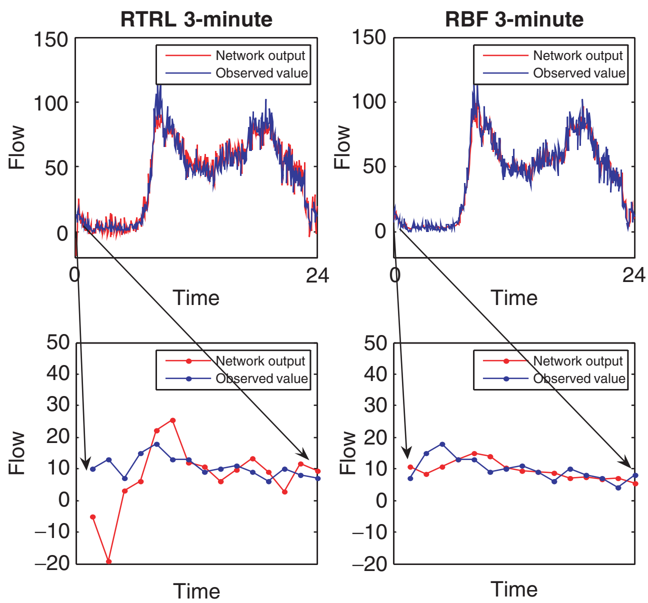

Figure 7. The RTRL and RBF network outputs and observed values of flows measured in 3 min intervals for the first 15 steps.

due to the convergent ability of the RTRL network, the average predictability for RTRL networks and for RBFNN is about the same. Further comparing the RMSEs in more detail, we find that for both RTRL networks and RBF networks, the RMSEs$_{(3\,\text{min})}$ are smaller than the RMSEs$_{(1\,\text{min})}$; similarly, the RMSEs$_{(15\,\text{min})}$ are smaller than the RMSEs$_{(5\,\text{min})}$. The findings suggest that the predictive accuracy for traffic dynamics measured in longer time intervals is better than those measured in shorter intervals.

4.3. *Prediction accuracy with various lags*

As mentioned previously, it is important to determine a proper time lag τ when analysing time series, especially when the time series in one dimension is mapped into multi-dimensional spaces. For the RTRL algorithms and RBFNN, it is postulated that the training process is a sequential learning scheme and that the traffic time series at time t and at time $(t+1)$ have relevant dependence, i.e. the time series is a first-order process or Markov process. Therefore, in this study, when considering the vector $[x(t-(q-1)\tau), \ldots, x(t-\tau)]$ as inputs and using the input vector to predict the desired value, $x(t)$, we set $\tau = 1$. Nevertheless, would the accuracy of prediction be better if we used other time lags? For instance, for a deterministic function, using the Mackey–Glass equation with a time lag $\tau = 6$ would produce the best accuracy of prediction compared to adopting other time lags.

Accordingly, Table 3 shows the prediction results for various time lags. For traffic data sets with 1- and 3-min intervals, we find that the prediction accuracy declines with increasing time lags for both RTRL algorithms and RBFNN, i.e. $\text{RMSE}_{(\tau=1)} < \text{RMSE}_{(\tau=2)} < \text{RMSE}_{(\tau=3)}$. Likewise, for traffic data sets with 5- and 15-min intervals, the prediction accuracy also declines with increasing time lags, i.e. $\text{RMSE}_{(\tau=1)} < \text{RMSE}_{(\tau=1/2\,\text{time delay})} < \text{RMSE}_{(\tau=\text{time delay})}$[1], except for one RBF case marked in gray (RMSE_flow_15 min $_{(\tau=\text{time delay})}$ < RMSE_flow_15 min $_{(\tau=1/2\,\text{time delay})}$).

Table 3. Prediction results of traffic dynamics for various time lags using RTRL and RBF.

Time interval	Traffic variable	Time lag	RMSE		Time lag	RMSE		Time lag	RMSE	
			RTRL	RBF		RTRL	RBF		RTRL	RBF
1 min (one workday, station 433)	Flow	1	0.0943	0.0851	2	0.0968	0.0960	3	0.0979	0.1074
	Speed	1	0.0510	0.0556	2	0.0632	0.0606	3	0.0669	0.0655
	Occupancy	1	0.0566	0.0433	2	0.0612	0.0458	3	0.0677	0.0466
3 min (one workday, station 433)	Flow	1	0.0787	0.0593	2	0.0790	0.0659	3	0.0792	0.0679
	Speed	1	0.0500	0.0547	2	0.0547	0.0666	3	0.0727	0.0754
	Occupancy	1	0.0510	0.0392	2	0.0599	0.0471	3	0.0662	0.0576
5 min (five workdays, station N27.9)	Flow	1	0.0686	0.0623	30	0.1897	0.0911	60	0.1947	0.1023
	Speed	1	0.0640	0.0587	42	0.0984	0.1484	84	0.0998	0.1718
	Occupancy	1	0.0500	0.0549	30	0.0870	0.0951	60	0.0895	0.1009
15 min (five workdays, station N27.9)	Flow	1	0.0648	0.0602	10	0.2011	0.093	20	0.2829	0.0898
	Speed	1	0.0574	0.0575	13	0.1024	0.0903	27	0.1062	0.1266
	Occupancy	1	0.0557	0.0513	10	0.1171	0.0748	20	0.1384	0.0763

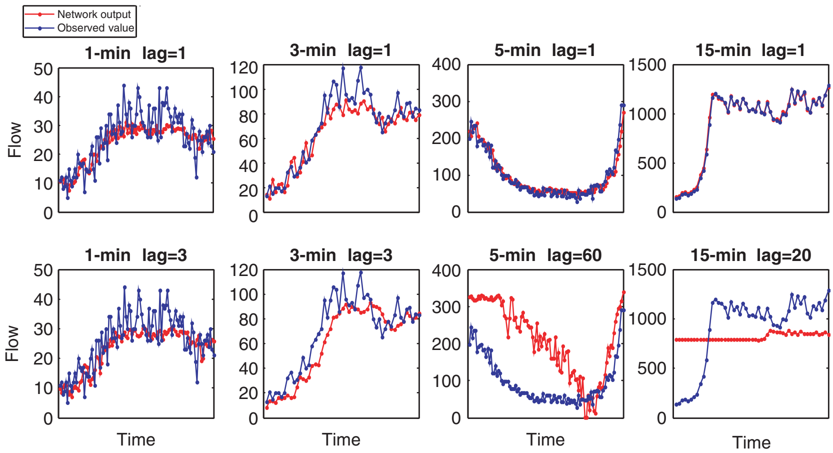

Figure 8. The RTRL network outputs and observed values of flows for various time lags.

Compared with the findings of Lan *et al.* (2008a), the above results seem again to indicate that the characteristics of short-interval traffic dynamics extracted from real world detectors measured within 15-min intervals and involving noises are more stochastic than deterministic; therefore, in the prediction of non-linear short-interval traffic dynamics, stochastic characteristics can be stronger than deterministic characteristics similar to the famous Mackey–Glass equation. Nevertheless, the only one exception for RBF model in Table 3 reveals that the 15-min flows have shown a slight tendency towards deterministic characteristics, so a better accuracy of prediction for 15-min flows using a proper time lag (i.e. time delay) occurs, compared to using half time delay. However, with regard to RTRL, owing to its real-time recurrent algorithms, the prediction accuracy significantly declines with increasing time lags. Figure 8 presents the differences between RTRL network outputs and observed values of flows for various time lags. In this figure, the same training data as Section 4.2 was employed; but only a portion of the data points are picked deliberately to clearly depict the differences.

4.4. *Prediction accuracy with various times of day*

In this scenario, we mainly aim at the time-of-day effect on the system performance. Therein, we attempt to identify the most critical time-of-day for prediction and how to improve the prediction accuracy. Accordingly, we tested the proposed algorithms using different data sets collected in times of day. Table 4 provides the corresponding prediction results. Observed from this table, we find that the values of RMSE in four time periods are different. In terms of 1-min flow, for RBFNN, the results are $\mathrm{RMSE}_{(18:00–21:00)} > \mathrm{RMSE}_{(06:00–09:00)} > \mathrm{RMSE}_{(12:00–15:00)} > \mathrm{RMSE}_{(00:00–03:00)}$. Likewise, other 3-min traffic variables, speed and occupancy, also have different RMSE values, depending on various time periods. Corresponding results for an RTRL network are $\mathrm{RMSE}_{(18:00–21:00)} > \mathrm{RMSE}_{(06:00–09:00)} > \mathrm{RMSE}_{(00:00–03:00)} > \mathrm{RMSE}_{(12:00–15:00)}$, which compared to the values of *RMSE*, it is noted that the order of $\mathrm{RMSE}_{(00:00–03:00)}$ and

Table 4. Prediction results of traffic dynamics during different time periods (station 433).

Time period	Traffic variable	Time lag	RMSE(1 min)		RMSE(3 min)	
			RTRL	RBF	RTRL	RBF
00:00–03:00	Flow	1	0.0653	0.0322	0.0548	0.0213
	Speed	1	0.0510	0.0479	0.0493	0.0395
	occupancy	1	0.0411	0.0101	0.0338	0.0082
06:00–09:00	Flow	1	0.1187	0.0954	0.1022	0.0755
	Speed	1	0.0533	0.0673	0.069	0.0603
	Occupancy	1	0.0864	0.0538	0.0851	0.0499
12:00–15:00	Flow	1	0.0693	0.0941	0.0532	0.0582
	Speed	1	0.0405	0.0391	0.0286	0.0305
	Occupancy	1	0.0401	0.0414	0.0315	0.0298
18:00–21:00	Flow	1	0.1258	0.1142	0.1059	0.0812
	Speed	1	0.0638	0.059	0.0558	0.0586
	Occupancy	1	0.0732	0.0647	0.0647	0.0527

$\text{RMSE}_{(12:00\text{–}15:00)}$ are reversed. This is because an oscillation often occurs during the beginning steps whenever one adopts the RTRL algorithms to train a network, hence the $\text{RMSE}_{(00:00\text{–}03:00)} > \text{RMSE}_{(12:00\text{–}15:00)}$. Such results may reveal that in general the morning and evening peak-hour periods remain the most critical for accurate prediction compared to other periods because serious jams are constantly incurred during such periods. Figure 9 illustrates the difference between RTRL outputs and observed values of flows during various periods of time in a workday (2004.02.04) at station 433.

To improve the accuracy of prediction during peak hours, a feasible method is to train a network that only consists of historical data for a specific time period, e.g. 06:00–09:00 or 18:00–21:00, in other words, to predict traffic dynamics at the same time period rather than to train a whole-day network to predict a specific time period of traffic dynamics. Table 5 illustrates the improved prediction results using this feasible method. It indicates that the prediction performance obtained from the network of historical data at specific time periods is better than that obtained from a whole-day network, that is $\text{RMSE}_{(8\ \text{days\_1-min\_06:00–09:00})} < \text{RMSE}_{(1\ \text{day\_1-min}\ 00:00\text{–}24:00)}$ and $\text{RMSE}_{(8\ \text{days\_3-min\_06:00–09:00})} < \text{RMSE}_{(1\ \text{day\_3-min}\ 00:00\text{–}24:00)}$.

5. Discussions

In this study, various techniques, including a linear method, simple non-linear prediction and RTRL algorithms were employed to compare their predictability. Wherein, a first-order AR model and a first-order differential-delay equation were used to test the predictability between the linear method, simple non-linear prediction and RTRL algorithms. After validating the prediction power of RTRL algorithms, we further implemented the sensitivity analysis by employing the short-term (within 15 min) traffic series, including flow, speed and occupancy measured with various time intervals, time lags

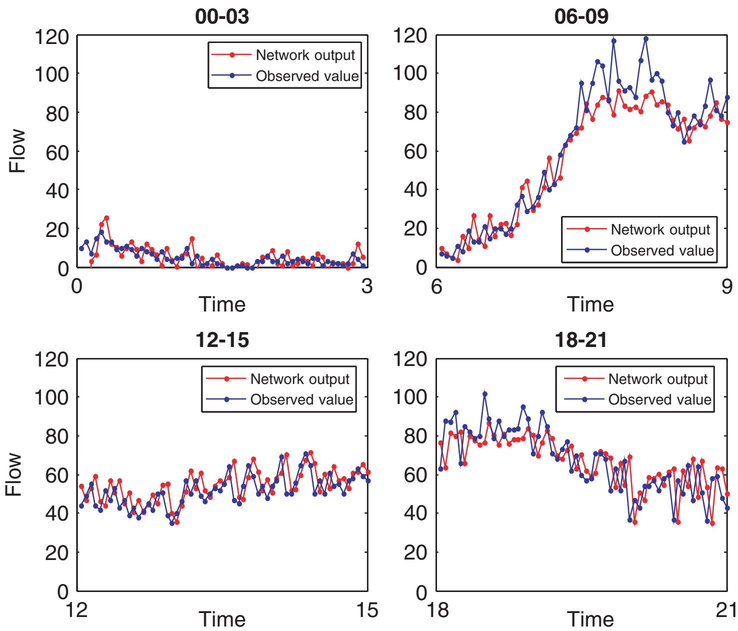

Figure 9. The difference between RTRL network outputs and observed values of flows during various time periods (3 min, station 433).

Table 5. Prediction results based only on peak-hours traffic data for network training.

Time period	Traffic variable	Time lag	RMSE (1 day, 1 min)		RMSE (1 day, 3 min)		RMSE (8 days, 1 min)		RMSE (8 days, 3 min)	
			RTRL	RBF	RTRL	RBF	RTRL	RBF	RTRL	RBF
06:00–09:00	Flow	1	0.1187	0.0954	0.1022	0.0755	0.0908	0.0742	0.0845	0.0645
	Speed	1	0.0533	0.0673	0.0690	0.0603	0.0501	0.0667	0.0461	0.0568
	Occupancy	1	0.0864	0.0538	0.0851	0.0499	0.0694	0.0531	0.0674	0.0414
18:00–21:00	Flow	1	0.1258	0.1142	0.1059	0.0812	0.1006	0.0812	0.0882	0.0711
	Speed	1	0.0638	0.0590	0.0558	0.0586	0.0594	0.0511	0.0501	0.0485
	Occupancy	1	0.0732	0.0647	0.0647	0.0527	0.0562	0.0366	0.0510	0.0334

and times of day. In accordance with the above investigation, we present the advantages and limitations of proposed RTRL model and summarise some important findings, and explain their nature here.

Except for the merits we portrayed in the above sections, the main reason for choosing RTRL algorithms in this article to train a non-linear traffic dynamics is that RTRL algorithms can iteratively modify performance errors and update its weighted parameters to meet the characteristics of traffic dynamics, which are neither deterministic nor completely random series but instead exhibit various features with times of day. In addition, in this study, the results have indicated that the RTRL algorithms have efficiently captured the trend and variation of non-linear traffic series after several training circles because the RTRL algorithms can be considered as BPN with feedback loops connecting to every hidden node, which exhibits dynamical learning algorithms. Nevertheless, one shortcoming of RTRL algorithms is that an oscillation often occurs during the initial training, and at the present time, the problems of deciding on the proper hidden neurons, hidden layers and training time can only be solved by a trial and error method. For instance, a high or low learning rate disadvantaged the training process because a higher learning rate referring to a network with larger modified weights could quickly achieve the goal of minimising the error function, yet in contrast, the higher learning rates could lead to an over-weighted adjustment and cause an error-oscillation phenomenon. Furthermore, if the number of hidden nodes are not sufficient, the relationship between input elements and observed data cannot be successfully constructed, which consequently leads to a large error gap. In contrast, too many nodes could cause a network with high freedom to over-fit the training sets, and, sometimes, too many nodes involving over description regarding noise existing in data sets would not only delay the convergent time but also fail to train the network.

In addition, from the comparison between different techniques, we have learned that it is very important to take into account the characteristics of traffic series before prediction. Without the prerequisite analysis, it is hasty to claim or determine which technique is the best to predict or is able to precisely predict a non-linear time series because different characteristics of the time series could greatly affect the accuracy of prediction. Moreover, different methods of prediction may only provide a certain function for a specific purpose rather than being capable of error-free predicting including all aspects. For instance, the simple non-linear technique can immediately learn the intrinsic rules of the dynamics to precisely catch the trajectories in multi-dimensional spaces within a few time steps. However, the requirement is that the underlying dynamics be deterministic or a time series with slight noises. Likewise, we successfully predict the short-term non-linear traffic dynamics extracted from the dual-loop detectors by employing RTRL algorithms as well as RBFNN. Nevertheless, as mentioned above, the problems of how to decide the proper hidden neurons, hidden layers and training time still remain crucial consideration prior to manipulating their algorithms. Consequently, in terms of prediction, characteristic analysis of a time series is important and is a prerequisite for prediction. Furthermore, what we would like to do is to select a technique that permits predicting the short-term traffic dynamics to meet the requirements of ATMS rather than arbitrarily searching for a method for perfect prediction without any errors.

Apart from the above findings, the traffic time series measured in different time intervals (1, 3, 5 and 15 min), with different time lags (time lag = 1, one-half time delay, 1-*time delay*) and during different times of day (00:00–03:00, 06:00–09:00, 12:00–15:00, 18:00–21:00) have been trained to predict traffic dynamics. According to our field study, several findings have been illustrated to support the accuracy of prediction when influenced by various time intervals, time lags and time periods. We have found that the accuracy of predicting traffic dynamics for longer intervals (15 min) is better than for shorter intervals (5 min); likewise, 3 min is better than 1 min. The findings may shed light on criteria that traffic dynamics with longer intervals (e.g. over 15 min) can be deemed as a simple problem, and around 6–8 hidden nodes are suggested to process the network while traffic dynamics with shorter intervals (e.g. less than 5 min) can be regarded as a complex problem and around 12–16 hidden nodes are recommended to tackle the network. Such criteria may be useful in practice to decide the appropriate number of hidden nodes, since determining the number of hidden nodes by using trial and error methods is sometimes time consuming. If a prediction model is slow to respond to changing parameters, then practicability of the model is extremely limited, and this point is one of the contributions of this article.

Moreover, a deterministic model with a proper time delay can precisely predict the dynamic state, for instance, $\tau = 6$ is a good time delay for the Mackey–Glass equation. In contrast, a stochastic time series for short time lag (e.g. $\tau = 1$) will produce a better prediction than adopting other time lags. Short-interval traffic dynamics extracted from detectors are very likely close to stochastic patterns, thus the training results of adopting time lag being equal to one produce optimum prediction compared to adopting other time lags. The findings that good prediction originates from a proper time delay seems to imply that searching for an appropriate time lag is not only very important for mapping a time series from one dimension to multi-dimensions, but is also a preliminary testing if one would like to select a NN as a tool to predict a time series, which does not originate from a given equation. Namely, pre-searching for the time delay of a time series helps one correctly determine input data and corresponding output data for training a network, with which fast and high prediction accuracy can be anticipated. The above findings can be regarded as the second distinguishing contribution of this article compared to previous studies.

On the other hand, we have also found that traffic dynamics in the morning and evening peak hours are the most difficult to predict compared to other time periods, but this situation can be improved by training a historical network using the traffic data composed of only the same time periods, i.e. gathering several historical data at the same time periods will produce a better training network to predict traffic dynamics. Nevertheless, it is noticed that one has to carefully select proper historical data when adopting the above approach to train a NN, wherein the 'proper historical data' means to pick similar historical data that emulates the trend and variance as the future traffic dynamics. Improper historical data (e.g. that involving serious incidents, bad weather, etc.) may contribute to unexpected inaccuracy in prediction. Such findings provide encouragement to tackle field problems like recurrent congestions that exist ubiquitously in transport systems rather than spend a vast amount of money to predict the slight fluctuations in the early hours. Theoretically, for recurrent congestions, we can alter the service process to more closely match the arrival patterns; making the arrival process more closely match the service capacity; or impose proper service disciplines to cut down the overall delay costs or the size of delays. The main challenge is to determine the proper times and intensity for actuating the control mechanism. Lan *et al.* (2008b) devoted a study to traffic dynamics and presented some tactics to deal with recurrent congestions. In this article, we adopted an improved method by training a historical network using the traffic data drawn from the same time periods to increase the prediction accuracy, which helped precisely capture the trend and fluctuation of traffic series in the morning and evening peak hours, and this will greatly benefit traffic system managers to determine the proper times and intensity for actuating the control mechanism. This is the third more meaningful and practical contribution of this article.

6. Concluding remarks

In this article, various techniques were conducted to predict traffic dynamics and the results bring to a conclusion that before selecting a proper technique to predict the diversity of dynamical systems, we should deliberately consider the characteristics of the dynamics and the purpose we would like to accomplish. The RTRL algorithms have also been configured to train non-linear traffic dynamics measured in different aspects. According to our results, the accuracy of prediction is influenced by time intervals, time lags and time periods. In addition, it is a prerequisite to discriminate from various features of traffic dynamics because the accuracy of prediction is also influenced by the characteristics of traffic dynamics. In brief, this study will be most valuable not only in presenting a feasible approach to predict the short-term traffic dynamics, but also in emphasising the significance that the characteristics of traffic dynamics affect the accuracy of prediction. Finally, our results have provided supportive evidence showing the value of training a NN in various aspects so as to provide more useful information for advanced traffic control and management practices.

Acknowledgements

This study is part of the research projects granted by the ROC National Science Council (NSC-91-2211-E-009-048, NSC-92-2211-E-009-058 and NSC 96-2416-H-009-011-MY3).

Note

1. According to the calculation of nonlinear autocorrelation function, using average mutual information θ_{ij}, one can find a time delay to represent the proper time lag. $\theta_{ij} = -\sum_{i,j} p_{ij}(\tau)\ln(p_{ij}(\tau)/p_i p_j)$, where p_i is the probability to find a time series value in the i-th interval, and $p_{ij}(\tau)$ is the joint probability that an observation falls into the i-th interval and the observation time τ later falls into the j-th interval.

References

Abrahart, R.J., See, L., and Kneale, P.E. 1998. New tools for neurohydrologists: using network pruning and model breeding algorithms to discover optimum inputs and architectures. *In*: *Proceedings of the 3rd international conference on geocomputation*, University of Bristol.

Broomhead, D.S. and Lowe, D., 1988. Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2, 321–355.

Chang, F.J., Chang, L.C., and Huang, H.L., 2002. Real time recurrent neural network for streamflow forecasting. *Hydrological Processes*, 16 (13), 2577–2588.

Chang, F.J. and Chang, L.C., 2005. *Neural network*. Taipei: Tung Hua.

Chang, W.F. and Mak, M.W., 1999. A conjugate gradient learning algorithm for recurrent neural networks. *Neurocomputing*, 24 (1–3), 173–189.

Chatfield, C., 1996. *The analysis of time series*. 15th ed. London, UK: Chapman & Hall.

Clark, S.D., Dougherty, M.S., and Kirby, H.R. 1993. The use of neural networks and time series models for short-term traffic forecasting: a comparative study. *In*: *Transportation planning methods*, *Proceedings of the PTRC 21st annual summer meeting*, Manchester.

Dawson, C.W. and Wilby, R.L., 2001. Hydrological modeling using artificial neural networks. *Progress in Physical Geography*, 25 (1), 80–108.

Dochy, T., Danech-Pajouh, M., and Lechevallier, Y., 1995. Short-term road traffic forecasting using neural network. *RTS English Issue*, 11, 73–82.

Dougherty, M. and Cobbett, M., 1997. Short-term inter-urban traffic forecasts using neural networks. *International Journal of Forecasting*, 13 (1), 21–31.

Fraser, A.M. and Swinney, H.L., 1986. Independent coordinates for strange attractors from mutual information. *Physical Review A*, 33 (2), 1134–1140.

Gardiner, C.W., 1997. *Handbook of stochastic methods for physics, chemistry and the natural sciences*. 2nd ed. Berlin: Springer.

Goh, S.L. and Mandic, D.P., 2003. Recurrent neural networks with trainable amplitude of activation functions. *Neural Networks*, 16 (8), 1095–1100.

Granger, C.W.J. and Newbold, P., 1976. Forecasting transformed series. *Journal Royal Statistical Society, Series B*, 38 (2), 189–203.

Haykin, S., 1999. *Neural networks: a comprehensive foundation*. NJ: Englewood Cliffs, Prentice-Hall.

He, G. and Ma, S., 2002. A study on the short-term prediction of traffic volume based on wavelet analysis. *In*: *The IEEE 5th international conference on ITS*, Singapore.

Hebb, D.O., 1949. *The organization of behavior. A neuropsychological theory*. New York: Wiley.

Iokibe, T., Kanke, M., and Yasunari, F., 1995. Local fuzzy reconstruction model for short-term prediction on chaotic time series. *Fuzzy Set*, 7 (1), 186–194.

Kalman, R.E., 1960. A new approach to linear filtering and prediction problems. *Tranasactions of the ASME-Journal of Basic Engineering, Series D*, 82, 34–45.

Kants, H. and Schreiber, T., 2004. *Nonlinear time series analysis*. Cambridge, UK: Cambridge University Press.

Kennel, M.B. and Isabelle, S., 1992. Method to distinguish possible chaos from colored noise and to determine embedding parameters. *Physical Review A*, 46 (6), 3111–3118.

Kennel, M.B., Brown, R., and Abarbanel, H.D.I., 1992. Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Physical Review A*, 45 (6), 3403–3411.

Kirby, H., Dougherty, M., and Watson, S., 1997. Should we use neural networks or statistical models for short term motorway traffic forecasting? *International Journal of Forecasting*, 13 (1), 43–50.

Kohonen, T., 1995. *Self-organizing maps*. 1st ed. New York: Springer.

Kwok, T.Y. and Yeung, D.Y., 1997. Constructive algorithms for structure learning in feed forward neural networks for regression problems. *IEEE Transactions on Neural Networks*, 8 (3), 630–645.

Lam, W.H.K., Chan, K.S., and Shi, J.W.Z., 2002. A traffic flow simulator for short-term travel time forecasting. *Journal of Advanced Transportation*, 36 (3), 265–291.

Lam, W.H.K., *et al*., 2005. Short-term travel time forecasts for transport information system in Hong Kong. *Journal of Advanced Transportation*, 39 (3), 289–306.

Lam, W.H.K., Tang, Y.F., and Tam, M.L., 2006. Comparison of two non-parametric models for daily traffic forecasting in Hong Kong. *Journal of Forecasting*, 25 (3), 173–192.

Lan, L.W. and Lin, S.Y., 2001. Prediction of short-term traffic dynamics–phase space local approximation method. *In*: *Proceedings of the 16th annual conference for the Chinese institute of transportation*, Taipei, 477–486.

Lan, L.W., Lin, F.Y. and Huang, Y.S., 2004. Confined space fuzzy proportion model for short-term traffic flow prediction. *In*: *Proceedings of the 9th conference of Hong Kong society for transportation studies*, Hong Kong, 370–379.

Lan, L.W., Sheu, J.B., and Huang, Y.S., 2007. Prediction of short-interval traffic dynamics in multidimensional spaces. *Journal of the Eastern Asia Society for Transportation Studies*, 7, 2353–2367.

Lan, L.W., Sheu, J.B., and Huang, Y.S., 2008a. Investigation of temporal freeway traffic patterns in reconstructed state spaces. *Transportation Research Part C*, 16 (1), 116–136.

Lan, L.W., Sheu, J.B., and Huang, Y.S., 2008b. Features of traffic time-series on multi-dimensional spaces. *Journal of the Chinese Institute of Transportation*, 20 (1), 91–118.

Lee, S., Lee, Y.I., and Cho, B., 2006. Short-term travel speed prediction models in car navigation systems. *Journal of Advanced Transportation*, 40 (2), 123–139.

Li, S., 2002. Nonlinear combination of travel-time prediction model based on wavelet network. *In*: *The IEEE 5th international conference on ITS*, Singapore.

Lin, W.H., Kulkarni, A., and Mirchandani, P., 2006. Response to ''Comment on 'Short-term arterial travel time prediction for advanced traveler information systems' by Wei-Hua Lin, Amit Kulkarni, and Pitu Mirchandani''. *Journal of Intelligent Transportation Systems*, 10 (1), 45–47.

Mackey, M.C. and Glass, L., 1977. Oscillation and chaos in physiological control systems. *Science*, 197, 287–289.

Mak, M.W., Ku, K.W., and Lu, Y.L., 1999. On the improvement of the real time recurrent learning algorithm for recurrent neural networks. *Neurocomputing*, 24 (1–3), 13–36.

Oller, L.E., 1985. Macroeconomic forecasting with a vector ARIMA model: a case study of the Finnish economy. *International Journal of Forecasting*, 1 (2), 143–150.

Ragwitz, M. and Kantz, H., 2000. Detecting nonlinear structure and predicting turbulent gusts in surface wind velocities. *Europhysics Letters*, 51 (6), 595–601.

Ragwitz, M. and Kantz, H., 2002. Markov models from data by simple nonlinear time series predictors in delay embedding spaces. *Physical Review E*, 65 (5), 056201(12 pages).

Rumelhart, D.E. and McClelland, J.L., 1986. *Parallel distributed processing: explorations in the microstructure of cognition*. Cambridge, MA: MIT Press, 1.

Sakawa, M., Kosuke, K., and Ooura, K., 1998. A deterministic nonlinear prediction model through fuzzy reasoning using neighborhoods' difference and its application to actual time series data. *Fuzzy Set*, 10 (2), 381–386.

Smith, B.L. and Demetsky, M.J., 1997. Traffic flow forecasting: comparison of modeling approaches. *Journal of Transportation Engineering*, 123 (4), 261–266.

Smith, B.L., Williams, B.M., and Oswald, R.K., 2002. Comparison of parametric and nonparametric models for traffic flow forecasting. *Transportation Research Part C*, 10 (4), 303–321.

Soltani, S., 2002. On the use of the wavelet decomposition for time series prediction. *Neurocomputing*, 48 (1–4), 267–277.

Tam, M.L., Lam, W.H.K., and Tang, Y.F., 2004. Short-term prediction of hourly traffic flows in Hong Kong. *In*: *Proceedings of the 9th conference of Hong Kong society for transportation studies*, Hong Kong, 360–369.

Tam, M.L. and Lam, W.H.K., 2008. Using automatic vehicle identification data for travel time estimation in Hong Kong. *Transportmetrica*, 4 (3), 179–194.

Tsekeris, T., 2006. Comment on ''Short-term arterial travel time prediction for advanced traveler information systems'' by Wei-Hua Lin, Amit Kulkarni, and Pitu Mirchandani. *Journal of Intelligent Transportation Systems*, 10 (1), 41–43.

Williams, R.J. and Zipser, D., 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1 (2), 270–280.

Werbos, P.J., 1994. *The roots of backpropagation: from ordered derivatives to neural networks and political forecasting*. New York: Wiley.

Yeh, I.C., 2004. *Applied neural network,* 3rd ed. Taipei: Ru-Lin Co.