Transportation Planning and Technology, June 2005
Vol. 28, No. 3, pp. 213–236

ARTICLE

Refining Genetically Designed Models for Improved Traffic Prediction on Rural Roads

MING ZHONG*, SATISH SHARMA* & PAWAN LINGRAS

*Faculty of Engineering, University of Regina, Canada, **Department of Mathematics and Computing Science, Saint Mary's University, Canada*

(Received 12 August 2004; Revised 10 March 2005; In final form 3 April 2005)

ABSTRACT Research into advanced traveler information systems (ATIS) for rural roads is limited. However, highway agencies expect to implement intelligent transportation systems (ITS) in both urban and rural areas. In this paper, genetic algorithms (GAs) are used to design both time delay neural network (TDNN) models as well as locally weighted regression (LWR) models to predict short-term traffic for two rural roads in Alberta, Canada. A top-down refinement was used to study the interactions between modeling techniques and underlying data sets for obtaining highly accurate models. It is found that LWR models achieve faster accuracy improvement than TDNN models over the refinement process. Compared with previous research, the models proposed here show higher accuracy. The average errors for the best LWR models obtained through the model-refining process are less than 2% in most cases. For refined TDNN models, the average errors are usually less than 6–7%. The resulting models indicate a level of high robustness over different types of roads, and thus may be considered desirable for real-world statewide ITS implementations.

KEY WORDS: Short-term traffic prediction; Model refinement; Time delay neural network (TDNN); Locally weighted regression (LWR); Genetic algorithm (GA)

Correspondence Address: Ming Zhong, Faculty of Engineering, University of Regina, Regina, SK, Canada, S4S 0A2.

ISSN 0308-1060 print: ISSN 1029-0354 online
DOI: 10.1080/03081060500120340

Introduction

Previous research into short-term traffic prediction has focused primarily on urban roads. The methodologies used can be broadly categorized into five basic approaches: the historical average approach [1–3] the neural network model [4–7], regression analysis [8,9], time series approach [10–12] and adaptive filtering techniques [13,14]. Little research has been undertaken specifically into short-term traffic forecasting on rural highways [15–18]. However, highway agencies expect to implement intelligent transportation systems (ITS) on all categories of highways, in both urban and rural areas. Demands for accurate short-term traffic prediction on rural roads already exist. For example, both the US Minnesota Department of Transportation [19] and the Washington State Transportation Center [20] have acknowledged the need for advanced traveler information systems (ATIS) for rural as well as urban highway networks. Washington State, for example, has even developed a taxonomy and specified evaluation mechanisms for both highway types.

It might be argued that it is not necessary to predict short-term traffic for rural roads on which traffic volumes are not sufficient to cause congestion. However, there are many interesting applications for ATIS in rural areas. For example, when some roads or lanes are closed due to accidents or maintenance, predicting short-term traffic is necessary since rural freeways are usually connected with various other types of road. ATIS applications on these freeways may rely on accurate short-term predictions from these connected roads so that proper detour or ramp controls can be implemented to avoid congestion. Toll authorities are also particularly interested in the ability of forecasting short-term traffic, because forecasts affect the number of toll collectors assigned to work on specific days or during specific periods [21].

The development of ATIS for different categories of highway will require different types of analysis. Yun *et al*. [16] applied three neural network models (namely the back-propagation model, the finite impulse response [FIR] model, and the time-delayed recurrent model) to three different data sets collected from interstate highways, intercity highways, and urban intersections. It was found that the time-delayed recurrent model outperformed other models in forecasting randomly moving data collected from urban intersections, and the FIR model showed better prediction accuracy for relatively regular periodic data collected from interstate and inter-city roads. In their study, the authors only tested the techniques on one type of data structure and suggested the best-fit technique for each set of data. However, it is possible that better results can be achieved by feeding refined patterns (e.g. patterns from a subset of data, instead of considering all available data at the same time) to these models. Moreover, they only tested non-linear models (e.g., neural networks) in their study. Previous research [18] indicates that it is possible that a linear model (e.g. regression) can achieve better performance based on refined datasets.

Choosing input and output parameters, as well as the projection technique, are important for successful short-term traffic forecasting. Among them, selecting input variables from a large number of historical data is a particularly difficult problem. Kalyar [22] experimented with a large set of input variables and used a subjective elimination process to find a smaller input subset. Dougherty and Cobbett [15] used a stepwise method to reduce network size by testing the elasticity of large neural networks. The general assumption in such elimination processes is that the smaller subset of input variables would guarantee little or no loss of accuracy. This paper uses a more objective technique based on genetic algorithms (GAs) for selecting the input variables.

It is also important to provide a generic method for obtaining appropriate models for different types of roads. In this paper, we use GAs to identify final input variables from a large number of candidate inputs for models with different data structures. The final selected input variables are then used to develop LWR and TDNN models. Their ability to produce short-term traffic predictions is tested on two distinct rural roads in Alberta, Canada. The model refinement process is illustrated, the interaction between modeling techniques and underlying data structures is explored, and highly accurate models are obtained for the roads under study. It is considered that such a method could be used by highway agencies to set up high-accuracy models for rural ATIS applications.

Applied Techniques

Three techniques are used in this paper. LWR analysis and TDNN are utilized as forecasting tools. GAs are used to select final inputs from a large number of candidate variables for both regression and neural network models. This section gives a brief review of these techniques.

Locally Weighted Regression Analysis

LWR is a form of instance-based (or memory-based) algorithm for learning continuous mappings from real-value input vectors to real-value output vectors. In locally weighted regression, points are weighted by proximity to the current x in question using a kernel, as shown in Figure 1. Local methods assign a weight to each training observation that regulates its influence on the training process. The

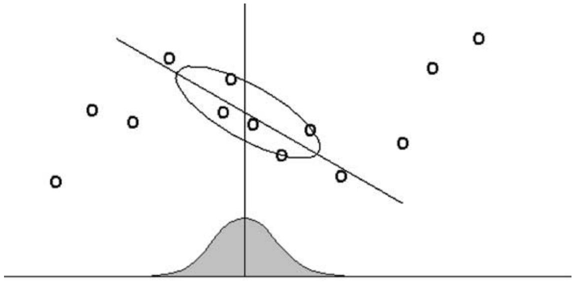

Figure 1. Locally weighted regression.

weight depends upon the location of the training point in the input variable space relative to that of the point to be predicted. Training observations closer to the prediction point generally receive higher weights [23]. This is achieved by weighting each data point according to its distance to the query point: a point very close to the query is given a weight of one, and a point far away a weight of zero. A common weighting function is Gaussian:

$$w_k = weight\ of\ data\ point\ x_k = \exp(-Distance^2(x_k, x_{query})/2K^2) \quad (1)$$

where the K parameter (called the kernel width) determines how quickly weights decline in value away from the query. Then instead of finding the line parameters a_i to minimize the global sum of squared residuals, the aim is to minimize the locally weighted sum of squared residuals [24]:

$$\sum_{i=1}^{N} w_k^2[y_k - \hat{y}(x_k)] \quad (2)$$

Model-based methods, such as neural networks and general linear regression, use data to build a parameterized model. After training, the model is used for predictions and the data are generally discarded. In contrast, 'memory-based' methods are non-parametric approaches that explicitly retain the training data, and use it each time a prediction needs to be made. Locally weighted regression is a memory-based method that performs regression around a point of interest using only training data that are 'local' to that point. One benefit of local modeling is that it avoids the difficulty of finding an appropriate structure for a global model. A key idea of local learning is to form a training set for the local model after a query is given. This approach permits the selection of only relevant samples and to weight them for the resulting models. After answering the query, the local model is discarded. A new local model is created to answer each query. The LWR program used in this study was downloaded from *Autonlab* [25].

Locally weighted regression has been used increasingly in control and prediction. Zografski and Durrani [26] explored the use of LWR in robot control and modeling time series, and also compared it to neural networks and other methods. Gorinevsky and Connolly [27] compared several different approximation schemes, such as neural networks, Kohonen maps, radial basis functions, to local polynomical fits on simulated robot inverse kinematics with added noise. They found that local polynomical fits were more accurate than all other methods. One recent study has demonstrated that LWR was suitable for real-time control by constructing a LWR-based system that learned a difficult juggling task [28].

Time Delay Neural Networks

A variant of neural networks used in this study is called the TDNN [29]. Figure 2 shows an example of a TDNN, which are particularly useful for time series analysis. The neurons in a given layer can receive delayed inputs from other neurons in the same layer. For example, the network in Figure 2 receives a single input from the external environment. The remaining nodes in the input layer obtain their input from the neuron on the left delayed by one time interval. The input layer at any time will hold a part of the time series. Such delays can also be incorporated in other layers.

The neural networks used in this study consist of three layers: input, hidden and output. The input layer receives data from the outside world; the input layer neurons send information to the hidden layer neurons; and the hidden neurons are all the neurons between the input and output layers – they are part of the internal abstract pattern, which

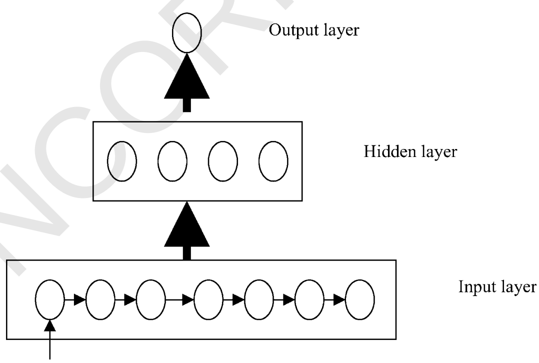

Figure 2. Time delay neural network design.

represents the neural network's solution to the problem. The hidden layer neurons feed their output to the output layer neurons, which provide the neural network's response to the input data.

Neurons process inputs and produce outputs. Each neuron takes in the output from many other neurons. The actual output from a neuron is calculated using a transfer function. In this study, a sigmoid transfer function is chosen because it produces a continuous value in the range [0,1] to normalize the data and facilitate the training and testing process. A neuron in a given layer is connected to neurons (n_1, $n_2, \ldots, n_m$) in the previous layer. The connection from n_j to n_i has the weight w_{ji}. The weights of the connections are initially assigned an arbitrary value between 0 and 1. The appropriate weights are determined during the training phase. Input to n_i is obtained using the following equation:

$$input_i = \sum_{j=1}^{n} w_{ji} \times output_j \tag{3}$$

Output from n_i is calculated using a sigmoid transfer function as:

$$output_i = f(input_i) \frac{1}{1 + e^{gain \times input_i}} \tag{4}$$

It is necessary to train a neural network model on a set of examples called the training set so that it adapts to the system it is trying to simulate. Supervised learning is the most common form of adaptation. In supervised learning, the correct output for the output layer is known. Output neurons are told what the ideal response to input signals should be. In the training phase, the network constructs an internal representation that captures the regularities of the data in a distributed and generalized way. The network attempts to adjust the weights of connections between neurons to produce the desired output. The back-propagation method is used to adjust the weights, in which errors from the output are fed back through the network, altering weights as it goes, to prevent the repetition of the error. Following training, an independent data set is used to test if the neural network is well trained. If testing results are satisfactory, the neural network is ready to be used for prediction purposes [29].

Genetic Algorithms

GAs and simulated annealing (SA) are stochastic search algorithms that attempt to strike a balance between the need to explore the solution space of a problem and the need to focus on the most promising parts of that space. The origin of GAs is attributed to Holland [30]. There has been significant interest in GAs over the last two decades [31]. The range of applications includes such diverse areas as: job shop scheduling, training neural networks, image feature extraction, and image feature identification. Previous research has shown that GAs consistently outperform both classical gradient search techniques and various forms of random search on more difficult problems, such as optimizations involving discontinuous, noisy, high-dimensional, and multimodal objective functions [32].

The GA is a model of machine learning, which derives its behavior from a metaphor of the processes of evolution in nature. In practice, the genetic model of computation can be implemented by having arrays of bits or characters to represent chromosomes, $c = (c_i | 1 \leq i \leq n)$, where c_i is a *gene*. Simple bit manipulation operations allow the implementation of crossover, mutation and other operations. The crossover operator creates new individuals called *offspring*, by recombining the genetic material of two individuals, deemed the *parents*. Individuals with higher fitness scores are selected with greater probability to be parents and 'pass on' their genes to the next generation. The mutation operator randomly alters one or more genes in an individual. Mutations add genetic diversity to the population.

GAs attempt to construct a superior individual by starting with a population of randomly generated individuals. From there on, the genetic operations, in concert with the fitness measure, operate to improve the population.

In this study, each candidate input was encoded as a gene. Individuals having 24 genes were entered into the evolution process for finding the best one whose genes – selected inputs – have maximum correlation with output.

Genetic Algorithms for Designing Neural Network and Regression Models

Many researchers have used GAs to determine neural network architectures. For example, Harp *et al.* [33] and Miller *et al.* [34] used GAs to determine the best connections among network units; Montana and Davis [35] used GAs for training neural networks; and Chalmers [36] developed learning rules for neural networks using GAs.

Hansen *et al.* [37] used GAs to design TDNN, which included the determination of important features such as the number of inputs, the number of hidden layers, and the number of hidden neurons in each hidden layer. Hansen *et al.* [37] clearly showed the advantages of using TDNN configured with GAs over other techniques including conventional autoregressive integrated moving average (ARIMA) methodology [38].

Hansen *et al.*'s [37] approach consisted of building neural networks based on the architectures indicated by the fittest chromosome. The objective of the evolution was to minimize training error. Such an approach is, however, computationally expensive. Another possibility used in this study is to choose the architecture of the input layer using GAs.

In traffic modeling time series, a large number of historical values usually exist. If all the historical values in the input layer were used in the TDNN calculations, it would lead to a complex and unwieldy network which may hinder the training process. A solution to such a problem is selective pruning of network connections. Lingras and Mountford [39] proposed the maximization of linear correlation between input variables and the output variable as the objective for selecting the connections between input and hidden layers. Since such an optimization is not computationally feasible with classical search techniques for large input layers, GAs were used to search for a near optimal solution. Compared with Hansen *et al*.'s [37] method, such an approach is simpler and easy to understand and implement. It should be noted here that since the input layer has a section of time series, it is not possible to eliminate intermediate input neurons. They are necessary to preserve their time delay connections. However, it is possible to eliminate their feedforward connections. Lingras and Mountford [39] achieved superior performance using the GA-designed neural networks for the prediction of inter-city traffic.

Selecting a subset, instead of using all historical values as inputs, can reduce the amount of data needed for developing the regression model. For example, at least 169 observations are needed for developing a regression model with 168 independent variables; whereas for a regression model with 24 independent variables, 25 observations are enough. Lingras *et al.* [18] used GAs to select final independent variables from a large number of candidate input variables for regression models. It was found that GA designed regression models achieved higher accuracy than subjectively designed models [18]. This paper uses the same techniques to develop various types of LWR and TDNN models. These models are then tested for short-term traffic predictions on two rural roads in Alberta, Canada.

Study Data Models

Study Data

Currently, Alberta Transportation employs about 350 automatic traffic recorders (ATRs) to monitor its highway networks. A hierarchical grouping method proposed by Sharma and Werner [40] was used to cluster road sections monitored by these ATRs into groups. After studying group patterns, five groups were obtained: commuter; regional commuter; rural long-distance; summer recreational; and winter recreational groups. Two study sites were selected for this study (Table 1). The first study site is on a collector road from the regional commuter (RC) group in West Alberta. The annual average daily traffic flow (AADT) of this road section was 3905 vehicles/day over the five study years (1996–2000). The second is an arterial road section belonging to the rural long-distance (RLD) group. It is located on Trans-Canada Highway 1 between the cities of Canmore and Calgary, Alberta. The average AADT over the study period was 13,627 vehicles/day.

Figure 3 shows the seasonal and hourly patterns of the study sites. It can be seen that they have remarkable peaks in the summer, as shown in Figure 3(a). A significant portion of traffic takes place from June to September, especially in July and August. Because RC carries a portion of commuter traffic, its seasonal variation is less than that of RLD. Figure 3(b) shows the hourly patterns on a typical work day (Wednesday). Both roads show two peaks in the day. However, the morning peak of RC is a few hours ahead of RLD. Moreover, the evening peak of RC is also more notable than RLD. Most traffic passes through the study sites in the afternoon and early evening (from 12:00 p.m. to 7:00 p.m.). There is a significant traffic increase in the morning and a significant decrease in the evening. The pattern indicates that during these hours in July and August, they are likely to have high traffic volumes and may experience congestion.

Data from 1996 to 1999 were used as the training set, and data from 2000 were used as the test set. Since these two roads carried different traffic volumes and showed different travel patterns, they should thus be able to test the robustness of the resulting models.

Table 1. Study roads and experimental data

Road section	Trip pattern group	Functional class	Location of monitoring counter	AADT	Training set	Testing set
RC	Regional commuter	Collector	2.4 km north of 22 and 567, Cochrane	3905	1996–1999	2000
RLD	Rural long-distance	Arterial	9.62 km east of 1 and old 1A, Canmore	13627	1996–1999	2000

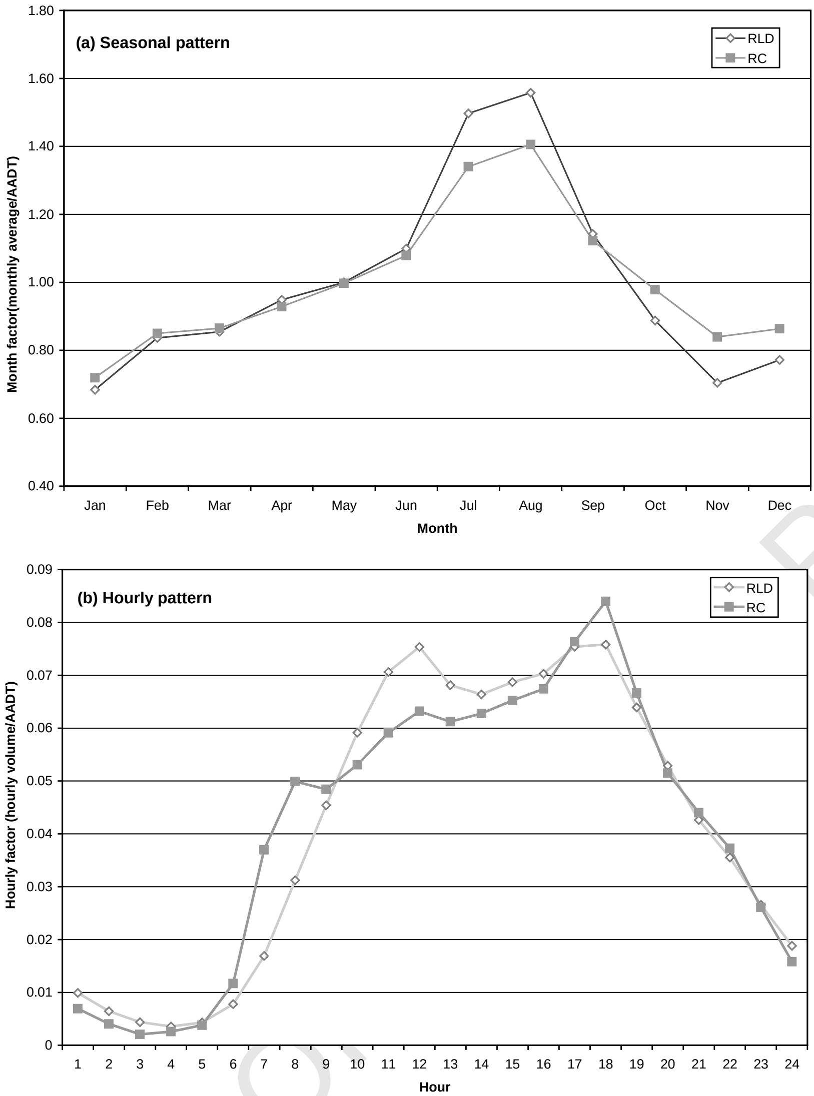

Figure 3. Seasonal and hourly patterns of the study sites.

Study Models

GAs were used to design both LWR and TDNN models. Figure 4 shows the prototype of the prediction models used in this study. First, candidate input patterns containing a section of historical hourly volumes before the predicted hour are presented to GAs for selecting a number of final inputs. The objective for GAs was to find a subset that

Figure 4. Forecasting model prototype.

had the maximum linear correlation with the outputs – predicted hourly volumes – among all possible combinations. The final selected input data were then used to develop both LWR and TDNN models. The resulting models are then applied to predict traffic volumes of interest.

Case study

In this study, week-long hourly volume patterns ($7 \times 24 = 168$ hourly volumes) before the predicted hours were used as candidate input patterns. It was assumed that these candidate input variables can provide all the necessary information for forecasting the hourly volume in the following hours. GAs were used to identify 24 variables from candidate input sets that had the highest correlation with the output variable, which was the observed traffic volume of the predicted hour. The number of final inputs was 24 because experiments indicated that increasing the number of final inputs led to little or no improvements in model accuracy. Moreover, it should be noted that using a small input set can reduce the size of training data, and hence decrease data constraints for the resulting models.

For the GAs, each chromosome consisted of 24 genes and represented a selection choice. The candidate input variables were labeled 1 to 168 depending upon their position in the input time series. The hourly volume immediately before the predicted hour was numbered 168, and that from a week ago with the predicted hour was numbered 1. Each gene was allowed to take a value from 1 to 168. The chromosomes with higher values of correlation were selected for creating the next generation and the linear ranking was used to evaluate the fitness. The population size was set to 110. The GAs were allowed to evolve for 1000 generations. A single point crossover operator was used and the crossover rate was set at 90%. For mutation, a random replacement operator was used and the probability of mutation was set to 1%. These parameters were selected since they usually lead to optimal results. The best chromosome selected after 1000 generations evolution was used as the final solution of the search. The connections selected by the GAs were used to design and implement the neural network and regression models. The resulting models were then applied to forecast traffic volumes over 12 hours (from 8:00 a.m. to 8:00 p.m.) for the study sites.

Different models were developed and then tested for the RC (collector) and the RLD (arterial) roads under study. A top-down model design method was used in this study. The method refines the underlying observations based on their temporal features (e.g. season, day of the week and hour). The purpose is to discover the interaction between the modeling techniques (regression, neural network and genetic algorithms) and the temporal homogeneity of the underlying observation sets. First, universal models were established to test their ability, and then they were further divided into single-hour models based on different hours. The observations of single-hour models were partitioned into groups based on seasons to develop seasonal single-hour models, and groups from individual hours and days of the week to develop day-hour models. Finally, both seasonal single-hour models and day-hour models were divided into seasonal day-hour models. The characteristics of these models were as follows:

(1) Universal models: this approach involved a single TDNN and a single LWR model for all 12 daytime hours across the year. The advantage of such models is the simplicity of implementation during the operational phase.
(2) Single-hour models: universal models were further divided into 12 single-hour models – that is, every hour had a separate model.
(3) Seasonal single-hour models: seasons have a definite impact on travel, so further dividing single-hour models into seasonal models may improve model accuracy. Universal single-hour models were further divided into summer (from May to August) single-hour models, winter (from November to February) single-hour models, and July–August single-hour models.
(4) Day-hour models: travel patterns also vary by days of the week. Further dividing observations into groups for each hour of different days (e.g. Wednesday or Saturday) may improve model accuracy.
(5) Seasonal day-hour models. day-hour models were further refined into July–August day-hour models to explore the models with higher accuracy.

Although all models developed in this study have the same prototype, their training and testing data sets differ. Data sets consist of different observation compositions, which show different degrees of homogeneity. For example, universal models are based on data sets consisting of observations from all 12 daytime hours throughout the year, but data sets for July–August day-hour models only include observations from the same hours (e.g. 7:00–8:00 a.m.) of the same days of the week (e.g. Wednesdays) in July and August.

There are in total two universal models, 24 single-hour models, 72 seasonal single-hour models, 48 day-hour models and 24 July–August day-hour models developed for this study.

Each model was trained and then tested. Depending on the model, the number of patterns or observations varied. The absolute percentage errors (APE) were calculated and the key evaluation parameters consisted of the average, 85th and 95th percentile errors. These parameters showed a clear profile of each model's error distribution.

Results and Discussion

Results

For comparison purposes, the average, 85th and 95th percentile errors of both LWR and TDNN models based on the same data sets are presented in one table (Table 2). Because short-term traffic prediction in real-world implementation is wholly a testing process, only the results based on test sets are included here.

First, universal models were used to predict hourly volumes between 8:00 a.m. and 8:00 p.m. of any day in the year. As noted previously, the benefit of such models is their simplicity for development and implementation. However, the shortcoming is high prediction errors. Table 2 shows the prediction errors of the universal models. The model's performance was evaluated based on the average, the 85th percentile and the 95th percentile errors for all hours combined and each individual hour. It can be seen from all hours combined that the errors for the RLD road are lower than those of the (RC) road. The reason may be that RLD has a higher traffic volume level and hence resulting in more stable traffic patterns. The mean average error for the universal LWR model was 9.73% for RC and 7.81% for RLD. The universal TDNN models outperformed the universal LWR models in both cases. The errors of the TDNN models were lower than the LWR models by 1%.

For combined hours, universal models seemed to perform well. Average errors for the LWR and the TDNN models were 7–9%. Even the 95th percentile errors were less than 25%. However, breaking

Table 2. Errors and breakdown of errors of traffic prediction using universal models

Hour	RC						RLD					
	Average		85th percentile		95^{th}%		Average		85th percentile		95^{th}%	
	LWR	TDNN	LWR	TDNN	LWR	TDNN	LWR	TDNN	LWR	TDNN	LWR	TDNN
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)
All hours	9.73	8.23	17.06	14.39	25.96	21.53	7.81	6.99	13.73	11.89	22.08	18.61
7–8	13.91	11.00	20.66	18.91	50.63	33.4	10.52	8.83	17.78	15.22	27.89	22.58
8–9	10.39	9.68	19.23	17.39	27.2	24.35	9.31	7.46	16.44	12.58	23.32	18.24
9–10	12.66	9.68	22.12	16.78	30.32	27.03	8.33	7.52	15.28	13.75	21.20	20.38
10–11	10.09	8.83	18.19	15.02	24.96	22.37	7.72	5.97	14.13	9.87	18.97	15.81
11–12	9.61	8.25	17.08	14.19	25.19	19.98	6.08	5.77	10.74	10.36	16.47	15.39
12–13	9.45	7.98	17.16	13.16	24.22	20.19	5.98	5.62	11.09	10.42	15.34	14.37
13–14	8.78	7.52	15.81	13.43	21.52	18.99	6.03	5.26	9.99	9.48	14.90	12.67
14–15	8.41	7.63	15.8	14.64	22.27	19.03	5.96	5.34	10.49	9.81	15.33	13.17
15–16	8.42	7.64	14.87	13.64	21.63	18.6	5.83	4.79	10.16	8.74	15.66	11.66
16–17	7.13	6.02	12.27	10.49	18.29	14.88	5.92	5.26	10.12	9.47	15.78	14.64
17–18	8.3	6.51	14.25	11.19	19.66	16.58	9.14	6.58	16.76	11.79	27.96	18.56
18–19	9.66	7.96	17.22	13.87	25.68	22.9	12.83	15.74	24.19	23.91	38.56	42.39

down the errors for individual hours reveal that errors for some hours are quite high. For example, when applied to RLD, the 95th percentile errors of the universal TDNN and LWR model are as high as 40% for the 6:00–7:00 p.m. hour. These large errors did not appear in the statistics for all hours combined because of the large number of observations included in the universal models.

The universal TDNN models showed superior performance to the LWR models for both roads. The errors of the TDNN models were less than those of the LWR models by about 1–3%. The study results clearly show that neural network models are better than regression models for modeling non-linear patterns.

The universal models were refined further for more accuracy. The refining process involved dividing a model's observations into more homogeneous groups, based on their temporal characteristics (e.g. season, day of the week and hour). Universal models were further divided into single-hour models, seasonal single-hour models, day-hour models, and seasonal day-hour models. Table 3 shows the mean average errors of short-term traffic predictions over 12 daytime hours (from 8:00 a.m. to 8:00 p.m.) resulting from these models, when applied to the two study roads. From Table 3, it can be seen that, as models were refined, errors usually decreased. The results indicate that dividing observations into groups for individual hours, days of the week, and seasons can significantly improve the accuracy of LWR models. In contrast, TDNN models are not sensitive to this process and do not result in much accuracy improvement. For example, for these two study roads, the mean average errors for the universal LWR models were 8–9%, whereas the mean average error for 12 daytime July–August day-hour LWR models was only 1–2%. However, for TDNN models, the mean average errors of July-August Day-hour models were reduced to 4–5% from 7–8% of the universal models. The improvement is only about 2–3%.

Figure 5 provides a visual comparison of the mean average errors of the different models. It can be found that the errors for RLD are generally lower than those for RC. It is interesting to note that the accuracy of LWR models was significantly improved through this process. However, TDNN models did not show much improvement. The error differences between different models are within 2%. Detailed pattern analysis indicates that the linearity of the underlying observations increases as the models are refined. LWR models achieved better results as the refinement process continued. It seems that neural networks could not capture the improved linear trend in the underlying data sets, and failed to provide more accurate predictions.

Table 4 shows the prediction errors of individual hours for July–August day-hour models. The LWR models resulted in highly accurate

Table 3. Comparison of model performance based on mean errors for 12 daytime hours

Model	RC						RLD					
	Average		85th percentile		95th percentile		Average		85th percentile		95th percentile	
	LWR	TDNN	LWR	TDNN	LWR	TDNN	LWR	TDNN	LWR	TDNN	LWR	TDNN
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)
Universal	9.73	8.23	17.06	14.39	25.96	21.53	7.81	6.99	13.73	11.89	22.08	18.61
Single-hour	7.65	7.78	13.40	13.85	20.35	20.42	8.12	6.97	16.70	12.24	22.21	18.53
Winter single-hour	8.01	9.10	13.87	16.14	20.68	23.13	7.42	7.27	12.92	12.74	18.51	19.08
Summer single-hour	6.67	7.09	12.01	12.53	16.40	17.15	4.67	4.85	8.37	8.43	11.84	12.34
July–August single-hour	5.99	6.59	10.73	11.31	15.00	15.22	4.30	4.87	7.66	8.69	10.74	12.04
Saturday day-hour	6.72	7.93	11.59	13.96	16.69	19.81	5.22	5.96	9.38	10.28	13.21	14.27
Wednesday day-hour	6.50	7.53	11.36	12.92	16.97	19.26	4.78	5.82	8.42	10.24	12.39	15.42
July–August day-hour	1.74	5.61	2.79	9.03	3.78	12.28	0.87	4.83	1.30	8.25	1.88	9.83

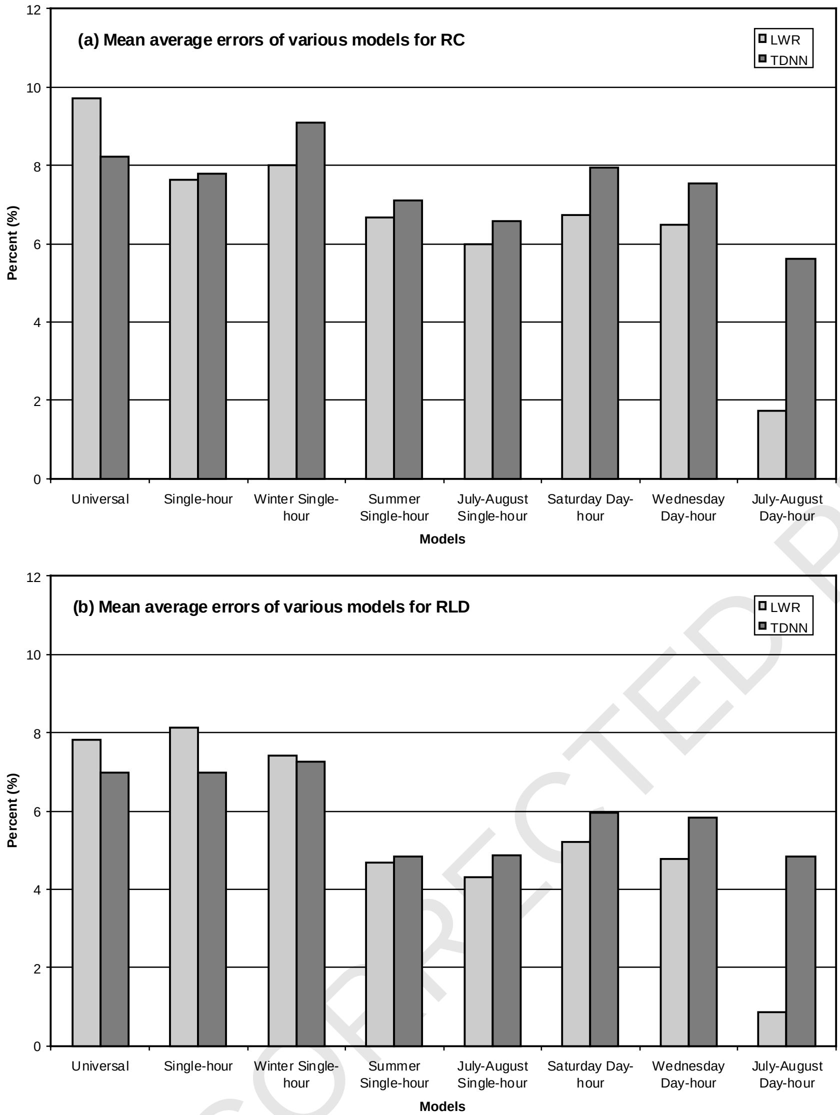

Figure 5. Comparison of mean average errors for different models.

predictions for both study roads. For the RC road average errors were usually less than 2% and the 95th percentile errors were usually lower than 5%. For the RLD road average errors were usually less than 1% and the 95th percentile errors were usually lower than 3%. The results clearly reinforce the robustness of refined LWR models over different types of roads. For TDNN models, the errors for RLD are slightly

Table 4. Errors of traffic prediction using July–August day-hour models

Hour	RC						RLD					
	Average		85th percentile		95th percentile		Average		85th percentile		95th percentile	
(1)	LWR (2)	TDNN (3)	LWR (4)	TDNN (5)	LWR (6)	TDNN (7)	LWR (8)	TDNN (9)	LWR (10)	TDNN (11)	LWR (12)	TDNN (13)
7–8	1.66	6.46	2.52	11.46	4.18	11.74	1.33	6.17	2.18	11.20	3.53	13.07
8–9	3.27	2.64	5.33	4.62	9.06	4.93	0.90	3.77	1.42	6.59	2.14	7.11
9–10	2.10	7.80	2.61	12.00	2.99	18.31	1.30	3.02	2.05	5.56	2.56	5.85
10–11	1.60	3.95	2.75	3.72	3.02	13.75	1.12	7.98	1.71	13.55	2.33	18.20
11–12	1.36	4.97	2.28	8.18	3.17	10.14	0.78	5.42	0.98	9.63	1.89	10.17
12–13	1.81	7.02	3.11	12.72	4.19	15.22	0.53	3.19	0.74	5.40	1.04	6.41
13–14	1.26	7.43	2.02	13.35	2.57	16.51	0.77	6.87	1.45	11.60	1.92	16.18
14–15	1.36	6.01	2.33	7.32	2.62	12.22	1.20	4.98	1.56	9.70	2.10	11.13
15–16	0.73	4.29	1.37	6.84	1.55	9.19	0.40	3.56	0.60	4.71	0.62	7.40
16–17	2.06	7.99	3.08	12.35	4.44	16.11	0.52	4.88	0.84	9.02	1.06	10.17
17–18	1.32	4.59	2.18	8.27	2.51	8.96	0.59	5.09	1.00	7.28	1.10	7.34
18–19	2.34	4.14	3.88	7.57	5.09	10.29	0.94	3.07	1.09	4.76	2.26	4.91

$$y = -47.68 + 0.122x_1 - 0.124x_2 + 0.904x_3 - 0.056x_4 - 0.173x_5 - 0.169x_6 + 0.434x_7 + 0.05x_8 + 0.38x_9 - 0.064x_{10} + 0.058x_{11} + 0.086x_{12} + 0.186x_{13} - 0.126x_{14} + 0.159x_{15} - 0.215x_{16} - 0.11x_{17} - 0.325x_{18} + 0.348x_{19} + 0.016x_{20} - 0.062x_{21} - 0.076x_{22} + 0.52x_{23} + 0.46x_{24}$$

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.999	.998	.995	3.19505

Analysis of Variance

ANOVA(b)

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	67139.347	24	2797.473	274.038	.000(a)
	Residual	112.292	11	10.208		
	Total	67251.639	35			

Figure 6. Regression analysis for the first-hour (7:00–8:00 a.m.) July–August day-hour model of RLD.

lower than those for RC. Average errors for the two roads were usually between 4% and 7%, and the 95th percentile errors usually range from 8% to 13%.

Figure 6 shows the July–August day-hour regression model based on the first hour (7:00–8:00 a.m.) data from RLD. It is evident that high linearity exists in the data ($R^2 = 0.998$). Regression models achieved high accuracy – the standard error of the estimate was only 3.19. That is, we would expect most estimates to have errors of only about 6 vehicles per hour (two standard deviations from the mean). Similar accuracy was obtained for the RC road.

It should be noted that the 95th percentile errors for some hours are more than 16% for the refined TDNN models (Table 4). A possible reason for their inferior performance is the small number of observations in the training and test sets – only 36 in the training sets and nine in the test sets. The results indicate that more data is needed for developing more accurate TDNN models. Another solution would be to expose the trained TDNN models to more field observations, as they can continuously adapt themselves to new data. Therefore, the performance of the TDNN models in real-world implementation may be better than those reported in Table 4.

Comparison with Previous Research

Shimizu *et al.* [41] applied three state estimation algorithms – Kalman filter, fixed-interval smoother and m-interval polynomial approximation (MIPA) Kalman filter – to forecast short-term traffic based on hourly volume data obtained from Fukuyama city, Japan. With a methodology similar to this study, they only estimated hourly traffic volumes using the observed data from 7:00 a.m. to 7:00 p.m. The techniques were applied to two simulation sets. It was reported that the mean absolute errors (MAE) of the Kalman filter and fixed-interval smoother was 8.78% and 7.51%, respectively, for the first simulation set. Another comparison between the Kalman filter and MIPA Kalman filter was made based on 22 day simulations. The MAE for the Kalman filter and MIPA Kalman filter was 13.58% and 7.71%, respectively.

A study by Williams *et al*. [11] compared seasonal ARIMA models with many other statistical models. The comparisons included ARIMA models used by Smith [42]: Winters exponential smoothing models; nearest-neighbor models; neural network models; and historical average models. Similar to this paper, the data used in their study was hourly traffic flow. It was reported that seasonal ARIMA models outperformed other models. They tested their models on two freeway sites in Virginia. The average errors for their best seasonal ARIMA models at two sites were 7.41% and 7.49%, respectively.

It was reported by Lu [14] that predicting traffic flow by historical traffic data is more difficult with a large sampling interval than with a small sampling interval. This conclusion indicates that more accurate results would be obtained if the same models were applied to data with smaller sampling intervals. Park *et al.* [4] tested various models, including a radial basis function (RBF) neural network model, Taylor series, exponential smoothing method, double exponential smoothing method and backpropagation neural network models, on 5-minute traffic volume data collected by the TransGuide System in San Antonio. It was reported that the MAPE of various models ranged from 8.08% to 17%.

In general, it is clear from the above that various techniques can be employed for short-term traffic prediction, with average errors usually more than 7–8%. On the other hand, the average errors (equal to MAE or MAPE) for the best refined LWR models in the present study are less than 2%, and those for the best refined TDNN models less than 6–7% (Table 4). And it is anticipated that these TDNN models are expected to perform better in real-world implementation. Thus, it is reasonable to conclude that the models proposed here have a higher level of accuracy than the models studied in previous research.

Conclusion

Previous research into the application of ATIS has focused mainly on urban roads, whereas research is needed for all categories of highways, as highway agencies expect to implement ITS across their jurisdictions, both in urban and rural areas. This study used genetically designed

LWR and TDNN models to predict short-term traffic on two distinct rural roads in Alberta, Canada. The aim has been to illustrate a model refining method, which is useful for discovering the interaction between modeling techniques and underlying data structures, and obtaining models with high levels of accuracy.

The analysis indicates that highly accurate LWR models can be developed through the model refinement method discussed in this study. For example, when tested on the RLD road, average errors for the universal LWR models were 8%, and the 95th percentile errors were over 20%. For the refined seasonal single-hour models (e.g. July–August single-hour models), average errors were usually about 5%, and the 95th percentile errors were reduced to 10%. Seasonal day-hour models showed the highest accuracy. Average errors for LWR models were usually less than 1.5%, and the 95th percentile errors usually lower than 3%. TDNN models also have some improvements over the refinement process; however, the improvements are not as significant as those of LWR models. The average errors were reduced by only 2% from the universal models to seasonal day-hours models, whereas the improvements for LWR models were usually more than 7–8%.

These results indicate that there are interesting relationships between the temporal homogeneity of underlying observation sets and modeling techniques and resulting accuracy. Regression models would be a better choice if the underlying observations are homogenous, for example, the seasonal day-hour model. Neural network models would be a better choice when the underlying observations vary in a significant range (e.g. universal models). When applied to RLD, the average error for the universal LWR model was 7.81%, whereas it was 6.99% for the universal TDNN models. In contrast, the refined seasonal day-hour LWR models resulted in much lower errors than refined TDNN models.

It is advantageous to use both linear and non-linear modeling approaches for short-term traffic prediction. Developing both neural network and regression models for the same data sets can provide an additional check on a model's accuracy. Refined LWR models usually have a higher accuracy but they are prone to outliers in data sets, whereas neural network models are known for their tolerance to interventions and outliers in the data. Hence, refined TDNN models using data sets with outliers can expect to provide a good benchmark of modeling accuracy. If the outliers in the data sets can be removed, refined LWR models based on cleaned data sets can provide more accurate predictions.

The genetically designed models developed in this paper show higher accuracy than the models proposed in previous research [4,14,41,42]. The average errors for the best LWR models developed in the present study are less than 2–3% and those for the best TDNN models are less than 5–6%. Testing results based on two roads with distinct travel patterns and traffic volume levels clearly show their robustness. Therefore, it is reasonable to conclude that these models are desirable candidates for ATIS applications in rural areas.

Acknowledgements

The authors are grateful to NSERC, Canada for their financial support and Alberta Transportation for the data used in this study.

References

[1] Stephanedes, Y. J., Michalopoulos, P. G. & Plum, R. A. (1981) Improved estimation of traffic flow for real-time control, *Transportation Research Record*, 795, 28–39.

AQ1 [2] Jeffrey, D. J., Russam, K. & Robertson, D. I. (1987) Electronic route guidance by AUTOGUIDE: the research background, *Traffic Engineering and Control*, 525–529.

[3] Kaysi, I., Ben-Akiva, M. & Koutsopoulos, H. (1993) An integrated approach to vehicle routing and congestion prediction for real-time diver guidance, *Transportation Research Record*, 1408, 66–74.

[4] Park, B., Messer, C. J. & Urbanik, T. (1989) Short-term freeway traffic volume forecasting using radial basis function neural network, *Transportation Research Record*, 1651, 39–47.

[5] Hua, J. & Faghri, A. (1994) Applications of artificial neural networks to intelligent vehicle-highway systems, *Transportation Research Record*, 1453, 83–90.

[6] Smith, B. L. & Demetsky, M. J. (1994) Short-term traffic flow prediction: neural network approach, *Transportation Research Record*, 1453, 98–104.

[7] Zhang, H., Ritchie, S. G. & Lo, Z. P. (1997) Macroscopic modeling of freeway traffic using an artificial neural network, *Transportation Research Record*, 1588, 110–119.

[8] Smith, B. L. & Demetsky, M. J. (1996) Multiple-interval freeway traffic flow forecasting, *Transportation Research Record*, 1554, 136–141.

[9] Lan, C. J. & Miaou, S. P. (1999) Real-time prediction of traffic flows using dynamic generalized linear models, *Transportation Research Record*, 1678, 168–178.

[10] Ahmed, S. A. & Cook, A. R. (1979) Analysis of freeway traffic time–series data by using Box–Jenkins techniques, *Transportation Research Record*, 722, 1–9.

[11] Williams, B. M., Durvasula, P. K. & Donald, E. B. (1998) Urban freeway traffic flow prediction: application of seasonal autoregressive integrated moving average and exponential smoothing models, *Transportation Research Record*, 1644, 132–141.

[12] Lee, S. S. & Fambro, D. B. (1999) Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting, *Transportation Research Record*, 1678, 179–189.

[13] Davis, G. A., Nihan, N. L., Hamed, M. M. & Jacobson., L. N. (1990) Adaptive forecasting of freeway traffic congestion, *Transportation Research Record*, 1287, 29–33.

[14] Lu, J. (1990) Prediction of traffic flow by an adaptive prediction system, *Transportation Research Record*, 1287, 54–61.

[15] Dougherty, M. S. & Cobbett, M. R. (1997) Short-term inter-urban traffic forecasts using neural networks, *International Journal of Forecasting*, 13, 21–31.

[16] Yun, S. Y., Namkoong, S., Rho, J. H., Shin, S. W. & Choi, J. U. (1998) A performance evaluation of neural network models in traffic volume forecasting, *Mathematical and Computer Modelling*, 27, 293–310.

[17] Lingras, P. (2000) Traffic volume time-series analysis according to the type of road use, *Computer-Aided Civil and Infrastructure Engineering*, 15, 365–373.

[18] Lingras, P. J., Sharma, S. C. & Zhong, M. (2002) Prediction of recreational travel using genetically designed regression and time delay neural network models, *Transportation Research Record*, 1805, 16–24.

[19] Minnesota Department of Transportation (1999) *Minnesota Guidestar: Shaping Tomorrow's Transportation System Today*. Available at http://www.dot.state.mn.us/trim/ats/guidestar/

[20] Washington State Transportation Center (1996) *Advanced Traveler Information Systems*. Available at http://www.ota.fhwa.dot.gov/tech/ma/ap71.html

[21] Federal Highway Administration (FHWA) (1997) *FHWA Study Tour for European Traffic Monitoring Programs and Technologies*. FHWA's Scanning Program (Washington, DC: Federal Highway Administration, US Department of Transportation).

[22] Kalyar, I. (1998) Prediction of Sunday afternoon traffic using neural network and regression analysis, MASc Thesis, Faculty of Engineering, University of Regina, Regina, Saskatchewan, Canada.

[23] Friedman, J. H. (1995) Intelligent local learning for prediction in high dimensions, *Proceedings of ICTDNN '95/NEURONIMES '95: International Conference on Artificial Neural Networks*, Paris, France.

[24] Moore, A. W., Schneider, J. & Deng, K. (1997) Efficient locally weighted polynomial regression predictions, *Proceedings of 14th International Conference of Machine Learning*, pp. 236–244.

[25] Autonlab (2001) *Locally Weighted Polynomial Regression*. Available at http://www.autonlab.org/lwpr_src/lwpr.html#1.

[26] Zografski, Z. & Durrani, T. (1995) Comparing predictions from neural networks and memory-based learning, *Proceedings of ICTDNN '95/NEURONIMES '95: International Conference on Artificial Neural Networks*, Paris, France, pp. 261–226.

[27] Gorinevsky, D. & Connolly, T. H. (1994) Comparison of some neural network and scatter data approximations: the inverse manipulator kinematics example, *Neural Computation*, 6, 521–542.

[28] Schaal, S. & Atkeson, C. (1994) Robot juggling: an implementation of memory-based learning, *Control Systems*, 14, 57–71.

[29] Hecht-Nielsen, R. (1990) *Neurocomputing* (Don Mills, Ont.: Addison—Wesley).

[30] Holland, J. H. (1975) *Adaptation in Natural and Artificial Systems* (Ann Arbor, MI: University of Michigan Press).

[31] Buckles, B. P. & Petry, F. E. (1994) *Genetic Algorithms* (Los Alamitos, CA: IEEE Computer Press).

[32] Grefenstette, J. (1986) Optimization of control parameters for genetic algorithms, *IEEE Transactions on Systems, Man, and Cybernetics*, 16(1), 122–128.

[33] Harp, S., Samad, T. & Guha, A. (1989) Towards the genetic synthesis of neural networks, in: D. Shaffer (Ed.) *Proceedings of the 3rd International Conference on Genetic Algorithms*, pp. 360–369 (San Mateo, CA: Morgan Kaufman).

[34] Miller, G., Todd, P. & Hedge, S. (1989) Designing neural networks using genetic algorithms, in: D. Shaffer (Ed.) *Proceedings of the 3rd International Conference on Genetic Algorithms*, pp. 379–384 (San Mateo, CA: Morgan Kaufman).

[35] Montana, D. & Davis, L. (1989) Training feedforward networks using genetic algorithms, in: N. Sridhara (Ed.) *Proceedings of 11th International Joint Conference on Artificial Intelligence*, pp. 762–767.

AQ2 [36] Chalmers, D. (1991) The evolution of learning: an experiment in genetic connectionism, in: D. Touretzky *et al*. (Eds.) *Connectionist Models: Proceedings of the 1990 Summer School* (San Mateo, CA: Morgan Kaufman).

[37] Hansen, J. V., McDonald, J. B. & Nelson, R. D. (1999) Time series prediction with genetic algorithm designed neural networks: an experimental comparison with modern statistical models, *Computational Intelligence*, 15(3), 171–184.

[38] Box, G. & Jenkins, J. (1970) *Time Series Analysis: Forecasting and Control* (San Francisco: Holden—Day).

[39] Lingras, P. J. & Mountford, P. (2001) Time delay neural networks designed using genetic algorithms for short term inter-city traffic forecasting, *Proceedings of the 14th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems,* Budapest, Hungary, pp. 292–299.

[40] Sharma, S. C. & Werner, A. (1981) Improved method of grouping provincewide permanent traffic counters, *Transportation Research Record*, 815, 12–18.

[41] Shimizu, H., Yamagami, K. & Watanabe, E. (1995) Applications of state estimation algorithms to hourly traffic volume system, *Proceedings of the 2nd World Congress on Intelligent Transport Systems*, Yokohama, Japan.

[42] Smith, B. L. (1995) Forecasting freeway traffic flow for intelligent transportation systems applications, PhD Thesis, Department of Civil Engineering, University of Virginia, Charlottesville, USA.