ITS Journal, 7:3–41, 2002

1024-8072/02 $12.00 + .00
DOI: 10.1080/10248070190048664

REFEREED PAPERS

SHORT-TERM TRAFFIC FLOW PREDICTION USING NEURO-GENETIC ALGORITHMS

Baher Abdulhai
Intelligent Transportation Systems Centre, Department of Civil Engineering, University of Toronto, Toronto, Ontario, Canada

Himanshu Porwal
Will Recker
Institute of Transportation Studies, Department of Civil and Environmental Engineering, University of California, Irvine, California

This paper presents a new short-term traffic flow prediction system based on an advanced Time Delay Neural Network (TDNN) model, the structure of which is synthesized using a Genetic Algorithm (GA). The model predicts flow and occupancy values at a given freeway section based on contributions from their recent temporal profile (over a few minutes) as well the spatial profile (including inputs from neighboring upstream and downstream sections). An in-depth investigation of the variables pertinent to traffic flow prediction was conducted examining the extent of the "look-back" in time interval, the extent of prediction in the future, the extent of spatial contribution, the resolution of the input data, and their effects on prediction accuracy. The model's performance is validated using both simulated and

Received 9 November 2000; accepted 13 September 2001.

This research was funded by the California Partners for Advanced Transit and Highways (PATH) and the California Department of Transportation (Caltrans). It was facilitated for by the California ATMS Testbed, headquartered at the University of California Irvine.

Address correspondence to Baher Abdulhai, Intelligent Transportation Systems Centre, Department of Civil Engineering, University of Toronto, Toronto, Ontario, Canada M5S 1A4. E-mail: baher@ecf.utoronto.ca

real traffic flow data obtained from the ATMS Testbed in Orange County, California. Both temporal and spatial effects were found to be essential for proper prediction. Results obtained indicate that the prediction errors vary inversely with the extent of the spatial contribution, and that the inclusion of three loop stations in both directions of the subject station is sufficient for practical purposes. Also, the longer the extent of prediction, the more the predicted values tend toward the mean of the actual, for which case the optimal look-back interval also shortens. The results also indicate that the level of data aggregation/resolution should be comparable to the prediction horizon for best accuracy. The model performed acceptably using both simulated and real data. The model also showed potential to be superior to such other well-known neural network models as the Multi layer Feed-forward (MLF) when applied to the same problem.

Keywords: neural networks; traffic flow; genetic algorithms; prediction; freeway

INTRODUCTION

Advanced Traffic Management and Information System components typically rely directly on traffic monitoring data as inputs to their underlying decision logic. These systems utilize either historical, current, or projected traffic data. In this context, the problem of reactive versus anticipatory, or proactive, traffic control has received considerable attention in the past few years. The prime question is whether to formulate control decisions to react to the latest observed traffic conditions or, rather, to attempt to forecast or anticipate short-term future conditions as the basis for control decisions. Reactive control, as the name implies, reacts to already-observed conditions of the traffic stream. Not only do such control systems await problems to arise before reacting, but also the conditions of the traffic system to which the response is addressed may have changed by the time the control decisions are formulated and implemented. Alternatively, proactive control targets near-term anticipated conditions, and the traffic network would operate (in theory, at least) under control strategies that are more relevant to the prevailing conditions. Because of these features, anticipatory control is conceptually preferred over reactive control. Nevertheless, the appeal of anticipatory control

is usually discounted because of the inaccuracy of the essential "traffic forecasting" component. Forecasting traffic conditions, even a few minutes into the future, has proven to be a challenging task that needs more research and attention.

It is important to distinguish between short-term and long-term traffic flow prediction horizons. Traffic flow prediction in general comes in different forms and for different purposes. Longer-term prediction can be for horizons as long as years, which is geared more towards planning applications as opposed to traffic operations. For traffic operations, short-term dynamics-focused prediction is more suitable. The most common tools for this are either statistically based approaches, such as time series, or the emerging, more promising but less mature, network-wide dynamic traffic assignment approaches. In our view, for some ITS applications, such as ramp metering, for instance, both time series and DTA are not necessarily the best approaches. Time series approaches are inadequate simply because they rely solely on temporal evolution of traffic patterns (if exist!) at a given point with disregard to the dynamics of flow and congestion propagation from neighboring freeway sections. DTA approaches are complex and iterative (in most cases) which might not be fully justifiable for fairly confined environments such as traffic on a pipe-like freeway. For such an environment, a better version of higher-order, dynamic, traffic flow models would suffice, which is the focus of this research.

In this paper, the problem of short-term forecasting of traffic variables is studied and a new Artificial Intelligence (AI) based model using a combination of Genetic Algorithms (GA) and Neural Networks (NN) is presented.

BACKGROUND

Evolution of traffic patterns at a particular location x is essentially a spatio-temporal process. If both space and time are discretized, traffic patterns at a location x at time t depend on traffic patterns at locations x, $(x - i)$ and $(x + i)$, $I = 1, 2, \ldots n$ (where n defines the extent of spatial contribution) at times $(t - j)$, $j = 1, 2, \ldots, m$ (where m defines the extent of temporal contribution or look-back interval). For such a relatively confined environment as a stretch of mainline freeway, upstream sections send traffic to the location under consideration and downstream sections may send backward propagating shockwaves as well, as shown in Figure 1. Although the process is intuitively simple, its modeling is not. Several factors interact in a complex manner, including the levels and spatio-temporal characteristics of traffic in both the affecting and the affected sections, as well as the less well-understood effects of

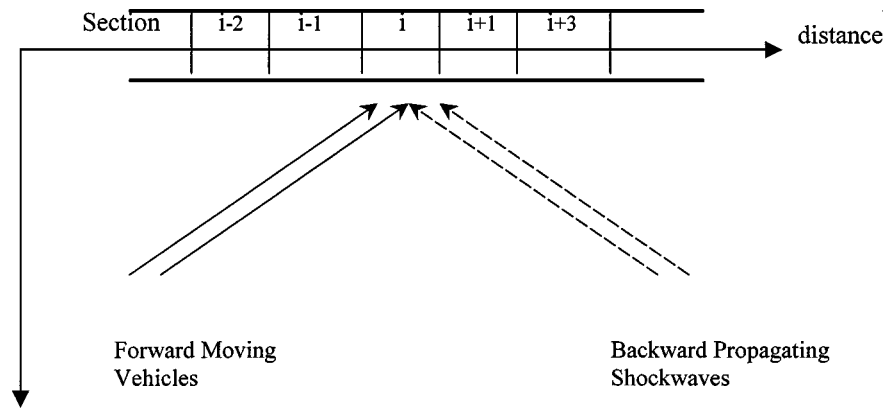


FIGURE 1. Schematic illustration of the traffic forecasting problem.

driver behavior. These factors combine with the general uncertainties of forecasting to make the dynamic prediction of traffic conditions a formidable modeling problem.

Although research in the area of traffic flow parameter prediction has been active in the past few years, the development of a model that yields sufficiently accurate and stable results has been elusive. A common approach to the problem of forecasting traffic parameters is based on time series models. However, forecasts using time series have been found to be *over-predictive and lagging* (Smith & Demetsky, 1995), which makes the prediction itself reactive in some sense, as it follows the current measurement with some time lag. Davis and Nihan (1991) attempted to replace the time series approach by a non-parametric regression approach, but they concluded that the performance of their k-nearest neighbor approach "*performed comparably to, but not definitely better than, the time series approach*." Several exploratory attempts have been made using Neural Networks (NN) as a replacement for the more traditional regression and time-series approaches (Smith & Demetsky, 1995; Doughetry & Lechevallier, 1995; Dougherty, Kirby, & Boyle, 1993). Common to all is a conclusion of potential superiority of NN and a recommendation for further in-depth investigation under different scenarios and using larger, real databases.

Existing conventional macroscopic traffic flow models also are numerous and vary from the very simple to the complex. A well-known set of such models is Payne's model and its offspring (with also well-known limitation; Daganzo, 1995). Payne (1971) introduced a traffic model that includes a momentum equation, derived from car-following theory concepts, in addition to the continuity equation that characterizes the well-known continuum model. It is the conceptual basis of Payne's model, rather its structural form, that has particular relevance to the research reported herein. The model relates the dynamic changes in traffic properties (speed) at a given section to both the temporal and spatial properties (speed and density) in the vicinity of the site, the latter by inclusion of a term that represents relaxation to equilibrium (that is, the effect of drivers adjusting their speeds towards the equilibrium speed-density relationship), the former with a term that represents anticipation, or the effect of drivers reacting to downstream traffic conditions (for example, the tendency to decrease speed if downstream density is higher due to congestion and vice versa).

Although it has been reported that the application of the Payne model has presented several problems, including instability (see, for example, Rathi, Lieberman, & Yedlin, 1987), the conceptual formalization of the model is appealing. However, its pre-specified model structure and

level of nonlinearity involved is, by its nature, limiting and without theoretical justification. In this research, we propose a modeling approach, based on Artificial Neural Networks (ANNs), that captures the overall spatio-temporal interaction of traffic parameters, without any limitations imposed by its model structure. The inputs to the proposed model are the most recently measured traffic parameters at the section under consideration, as well as from both the upstream and the downstream chain of sections; the output of the model is the anticipated traffic condition at that location in the near future. In addition, the model incorporates an adaptive feature that tailors it to the dynamic traffic environment.

Although the proposed model draws heavily on concepts embedded in traffic flow theory and resultant models, as well as on those forming the basis of forecasting models, the model structure itself is new, adaptive, and dynamic—the model's structure and underlying non-linearity evolve dynamically in response to the traffic environment through the use of Genetic Algorithms (GA) that "evolve" several advanced Time-Delay-based Neural Networks (TDNN). Moreover, the key variables in the prediction problem, including:

1. the extent of the look-back interval in time (how many minutes in the past, from current, affect the prediction),
2. the extent of prediction in the future or prediction "horizon,"
3. the extent of spatial contribution from neighboring freeway sections (how many loop stations upstream and downstream the subject station affect the prediction), and
4. the resolution of the data used for prediction (30 sec., 1 min., 5 min., etc.),

are themselves determined by an optimization process, rather than by arbitrarily fixing them through some trial-and-error process as common in the literature. The model development and results presented here incorporate: real as well as simulated freeway data, different freeway sites with different geometrics and entry and exit ramps, peak and near-peak traffic conditions to capture different levels of congestion, and different NN architectures.

GENETICALLY SYNTHESIZED/OPTIMIZED NEURAL NETWORKS

Genetic algorithms, from artificial intelligence, are defined by a problem-solving methodology that uses genetics as its model for problem solving, applying the rules of reproduction, gene crossover, and mutation to a

population of candidate solutions or pseudo-organisms. Those organisms can pass beneficial and survival-enhancing traits to new generations (Chambers, 1996). GA are known to be a powerful new technology for searching through large and complex solution spaces featuring large numbers of local minima. Although GA do not necessarily guarantee global optimality, they reach "practically" optimal solutions. Herein, the word "optimal" is loosely used to indicate practical optimality as opposed to global optimality in the mathematical sense.

Alternatively, Artificial Neural Networks (ANN, or simply, NN), also from artificial intelligence are mathematical models inspired by the human brain structure. ANNs prove to be superior to conventional techniques in the particular area of traffic pattern recognition and classification and are capable of learning from exemplar patterns (see for instance Abdulhai and Ritchie, 1999a, b, c; Abdulhai, 1996). The choice of neural networks structure and parameters, however, is an empirical-artistic exercise that relies on "rules of thumb" derived from past development experiences. The space of possible architectures and parameter combinations is extremely large. As a consequence, some significant amount of trial-and-error experimental hand-crafting is necessary before an adequate solution is achieved. It is impractical to rely on such "guesstimation" and trial-and-error to design networks for serious real-world problems. The empirical approach does not always produce a near-optimal network. Additionally, a good solution might be data-dependent, requiring re-optimization after every significant change in the application environment. The search for the best attainable network structure and parameter-setting combination is therefore a logical application for genetic algorithms.

Genetic algorithms have been applied to the problem of NN design in several ways. For instance, Montana and Davis (1989) have explored the use of GA in training a NN of known structure. Belew, McInerney, and Schraudolph (1990) used GA to set the learning and momentum rates for feed forward NN. Chang and Lippmann (1991) used GA to preprocess data in order to reduce the inputs to a NN without degrading performance. Harp and Samad (1991) explored using GA to discover the size, structure, and parameters of a network to be trained by a separate NN learning algorithm. Koza and Rice (1992) looked at GA as a tool for developing architectures and weights together. In 1997 BioComp Systems released a Neuro-Genetic Optimizer for the architectural optimization of neural networks. More details on the subject of using GA for NN development can be found in Chambers (1996) and Winter et al. (1995).

THE TIME DELAY NEURAL NETWORK MODEL

The Time Delay Neural Network (TDNN) (Haykin, 1994), schematically shown in Figure 2, features multiple connections between the individual neurons, as opposed to single connections as in the more basic NN. The multiple connections look-back over time to capture the temporal evolution of patterns in the data; that is, each neuron is provided with a memory in order to remember previous layer outputs for N periods of time. This is different from just lagging inputs N periods of time, as the look-back period of the TDNN affects the hidden layer and output layer as well, causing it to remember previously developed patterns and not just inputs.

Non time-delay NN such as the MLF can only learn an input-output mapping that is *static*. This form of mapping is well suited for cases where both the input vector and the output vector represent *spatial* patterns that are independent of time. It can be used to perform nonlinear predictions on a stationary time series, that is, when its statistics do not change with time. However, the time dimension is important for traffic flow predictions as traffic flow patterns evolve and change with time. Therefore, TDNN is expected to outperform MLF-like models and can be considered a more general form of the MLF. Similar to the MLF, the TDNN employs back propagation techniques for setting weights between neurons.

To train the TDNN network (further details can be found in Haykin (1994)), the actual response of each neuron (predicted traffic variables) in the output layer is compared with a desired target response at each time instant. Assume that neuron j lies in the output layer with its actual

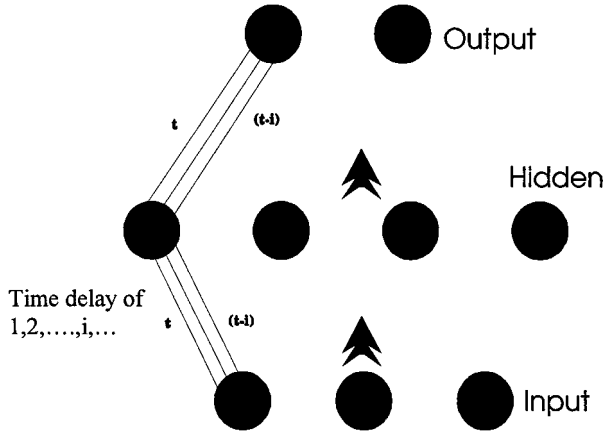


FIGURE 2. Architecture of the time delay neural network.

response denoted by $y_j(n)$ and that the desired response for this neuron is denoted by $d_j(n)$, both of which are measured at time n. The instantaneous value for the sum of squared errors produced by the network is as follows:

$$\xi(n) = \frac{1}{2}\sum_j e_j^2(n) \tag{1}$$

where the index j refers to the neurons in the output layer only, and $e_j(n)$is the error signal, defined by

$$e_j(n) = d_j(n) - y_j(n) \tag{2}$$

The goal is to minimize the cost function defined as the value of $\xi(n)$ computed over all time:

$$\xi_{\text{total}} = \sum_n \xi(n) \tag{3}$$

Differentiating the cost function with respect to the weight vector w_{ji}

$$\frac{\partial \xi_{\text{total}}}{\partial w_{ji}} = \sum_n \frac{\partial \xi(n)}{\partial w_{ji}}$$

using chain rule to express the derivative of cost function ξ_{total} with respect to the weight vector,

$$\frac{\partial \xi_{\text{total}}}{\partial w_{ji}(n)} = \sum_n \frac{\partial \xi_{\text{total}}}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)} \tag{4}$$

where the time index n runs over $v_j(n)$ and not $\xi(n)$. The partial derivative $\partial \xi_{\text{total}}/\partial v_j\,(n)$ is the change in cost function ξ_{total} produced by a change in the internal activation potential v_j of neuron j at time n. Moreover we recognize that

$$\frac{\partial \xi_{\text{total}}}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)} \neq \frac{\partial \xi(n)}{\partial w_{ji}(n)} \tag{5}$$

It is only when the expression is summed over all n that the equality holds. From (4) and using the idea of gradient descent in weight space,

the updating of the weights is done as follows:

$$w_{ji}(n+1) = w_{ji}(n) - \eta \frac{\partial \xi_{\text{total}}}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)} \tag{6}$$

where η is the *learning-rate parameter*. For any neuron j in the network, the partial derivative of the activation potential $v_j(n)$ with respect to the weight vector $w_{ji}(n)$ is given by $\frac{\partial v_j(n)}{\partial w_{ji}(n)} = x_i(n)$, where $x_i(n)$ is the input vector applied to neuron j. The local gradient for neuron j is

$$\delta_j(n) = -\frac{\partial \xi_{\text{total}}}{\partial v_j(n)} \tag{7}$$

Accordingly, we may rewrite (6) as

$$w_{ji}(n+1) = w_{ji}(n) - \eta \delta_j(n) x_i(n) \tag{8}$$

The explicit form of the local gradient $\delta_j(n)$ depends on whether neuron j lies in the output layer or a hidden layer of the network. The local gradients for the two cases are given below as

$$\delta_j(n) = \begin{cases} e_j(n)\varphi'(v_j(n)), & \text{neuron } j \text{ in the output layer} \\ \varphi'(v_j(n)) \sum_{m \in A} \Delta_m^T(n) w_{mj} & \text{neuron } j \text{ in the hidden layer} \end{cases}$$

where,

$$\varphi'(v_j(n)) = \frac{\partial y_j(n)}{\partial v_j(n)}$$

$\Delta_m^T(n) w_{mj}$: inner product of the vectors $\Delta_m(n)$ and w_{mj} both of which have dimensions $(m+1)$.

$\Delta_m(n) = [\delta_m(n), \delta_m(n+1), \ldots, \delta_m(n+M)]^T$, a vector.

A: set of all neurons whose inputs are fed by neuron j, located in a hidden layer, in a forward manner.

$v_j(n)$: internal activation potential of neuron j that belongs to set A.

In addition to the TDNN, this research also uses a modified version of the TDNN known as the *Continuous Adaptive Time neural network*, CATNN in its development of traffic flow prediction models. The CATNN have look-back intervals that adapt automatically as learning

progresses, seeking phase relationships that produce higher correlation over history. This adaptability in simply achieved by treating the look-back interval in the CATNN as a variable that gets optimized during the learning phase. It is important to emphasize that the only difference between the TDNN and CATNN is that the latter treats the look-back interval as an endogenous variable to be optimized among all the other variable and weights as opposed to being an exogenous input in the case of the TDNN. This usually adds significant flexibility in favor of the CATNN. However, in this particular application, a genetic algorithm is used to synthesize both networks, which in turn "examines" numerous TDNN with a variety of "fixed" look-back intervals, and therefore indirectly optimizes the look-back interval. To summarize, the look-back interval is optimized in the TDNN exogenously by the GA while being optimized endogenously by the CATNN, resulting in no significant difference between the performance of the two. For future other applications, it should be noted however that if a GA were not to be used, then the CATNN should be a better candidate compared to the TDNN. A commercial shell (NGO 1997), was used for synthesizing and training both the TDNN and the CATNN.

Lastly, it should be noted that having a time dimension in the form of a look-back feature in the TDNN and CATNN network types is what makes them particularly suitable for learning spatial patterns that change in time (i.e., *spatio-temporal*). Therefore, they tend to be superior to time series approaches that ignore the spatial component of the spatio-temporal patterns and also superior to static artificial neural networks that ignore the temporal component as our results will demonstrate shortly.

GENERAL PROCEDURE AND DATA DESCRIPTION

A section of Interstate 5 (I-5) freeway in Orange County, California, was selected for this research. The length of the section is approximately five miles with nine loop detector stations along the main line between the intersection of the I-5 and the I-405 freeways and the intersection of Jeffrey Road and I-5 in the city of Irvine as shown in Appendix A. This section also includes two off-ramps and five on-ramps. Therefore, flow and density data from a total of 16 loop detector stations were used. Real-time on-line loop data were available via the Advanced Traffic Management (ATMS) Testbed headquartered at the University of California Irvine (UCI).

In addition to these real-time loop data, a comprehensive set of simulated data was produced using Paramics, a state-of-the-art, ATMS-ready microscopic simulator (Paramics, 1998). A dynamic Origin-Destination

matrix available for the Testbed network that includes the section noted above was used to drive the simulator; O-D data for the evening peak of April 2, 1997, from 16:00 hr. to 18:00 hr. were arbitrarily selected. A comprehensive network (the whole Irvine network) that included the section of the I-5 freeway used as the test site was coded into Paramics with the exact geometry and loop detector station layout as in the real world. At each detector station, flow and density values were collected, summed across lanes, and used to develop the NN models. The finest resolution of the data used was 30 seconds.

The simulation data set was used for training, testing, and validating the TDNN/CATNN. The real data were reserved for real-world validation of the model and were not used in the development phases due to absence of ramp data. It is worthy at this point to justify the use of simulation data. Real data from on-ramps and off-ramps was not available in Southern California at the time of this research, a problem the fixing of which is beyond the scope, time, budget, and resources of this research. This lack of data is different from the problem of missing loop data and existence of holes in the data as frequently encountered in practice. Holes in the data can be fixed using numerous methods, the simplest of which could be simple interpolation. In our case however, ramp data were not available at all, forcing the research to resort to the most advanced simulation model available to us, keeping the rather limited main line real data for validation only. It should be noted that ramp data is necessary for vehicle conservation (total inflow = total outflow).

The neural networks were trained to predict the flow and density for a loop located near the center of the five-mile section, based on the flow and density in the recent past at both the test loop location as well as at the neighboring upstream and downstream loop locations. Of the nine mainline loop stations (numbered consecutively), station number 5 was used as the location at which prediction takes place, given the input from all stations (1 to 9 and the on/off ramp stations).

The "walk forward" method was used for developing the network. Under this method, training is conducted on the first preset number of records in the data set (defined as the training set), then testing is conducted on the next preset number of records (defined as the test set), followed by validation on the following preset number of records (defined as the validation set). This combination is called the first "fold." Once a fold is complete, the process walks forward in the data, and the training, testing, and validation episodes are repeated. The process continues until the end of the data is reached. The essence of the walk-forward method is to assure that training, testing, and validation pass iteratively over all the data. This way the network captures all information in the input space

rather than focusing on a subset of it, while maintaining independence between training on one hand and testing and validation on the other. The "walk forward" parameters used are 180 training records, 25 testing records, and 25 validation records.

A genetic algorithm (GA) is applied to an initial population of TDNN and CATNN networks of different structures, each of which representing one artificial chromosome. The objective here is to synthesize the neural networks themselves, that is, the variables treated by the GA are those pertinent to the structure of the neural networks, such as the number of hidden processing units, look-back interval, learning rate, input pruning, and so on, as opposed to the traffic variables themselves that the neural networks operate upon. The GA carries out simulated evolution on a population of network as follows:

1. Initialize a population of chromosomes (each is a neural network with a specific structure).
2. Evaluate fitness of each chromosome in the population in the form of the network's prediction performance.
3. Create new chromosomes by mating current chromosomes; select two parents at a time with selection probability made proportional to fitness value, and apply crossover (genetic material combination) and mutation (random alteration of some bits) to create two new children.
4. Delete members of the population to make room for the new chromosomes.
5. Evaluate the new chromosomes and insert them into the population.
6. Repeat till convergence is achieved or time is up.

Each network is transformed into one chromosome in the form of a binary string. The goodness of each of these solutions is evaluated and stored as a fitness record. This is the initial generation of chromosomes. The roulette-wheel-parent-selection (RWPS) mechanism is then applied to select two parent chromosomes for further mating and reproduction. The RWPS mechanism makes a list of available chromosomes, their fitness, and the running total of fitness, then generates a random number between zero and the accumulated total fitness, followed by selecting the first chromosome at which the running total of fitness is greater than or equal to the random number. This is analogous to the allocation of a pie-shaped slice of a roulette-wheel to each population member, with each slice being proportional to the member's fitness. Although, the effect of the RWPS mechanism is to return a randomly selected parent, each parent's chance of being selected is directly proportional to its fitness. On balance, over a number of generations, this will drive out the least fit members and contribute to the spread of the genetic material of the

fittest population members. Once two parents have been selected this way, one-point crossover is applied. Crossover recombines the genetic material in two parent chromosomes to make two children. One-point crossover occurs when parts of two parent chromosomes are swapped after a randomly selected point. Extra diversity in the children is added by applying bit mutation. When bit mutation is applied to a bit string, it sweeps down the list of bits, replacing each by a randomly selected bit if a probability test is passed. After a whole new generation of children is produced, they are evaluated and their fitness returned. Better solutions evolve as the process of reproduction and generational replacement proceeds. The process is terminated once it converges (does not improve further) or after a pre-specified number of generations. Further details on GA can be found in text books such as Chambers (1996).

For the genetic algorithm used to optimize the neural network structure, 30 generations and a population size of 300 were employed.

A number of different sets of TDNN and CATNN were developed and applied, in a phased sequence, to systematically:

- investigate the effect of the extent of prediction into the future on the prediction accuracy, using 30-second data resolution,
- optimize the spatial contribution from neighboring detector stations, and
- optimize/select data resolution that minimizes prediction errors.

PRELIMINARY INVESTIGATION AND CHOICE OF OBJECTIVE FUNCTION

Because the number of generations and the population size used by the genetic algorithm has a direct bearing on the optimality of the structure of the resulting neural network, a phase of preliminary investigation was necessary. In this phase, 30-second flow and densities from all stations were used to train the TDNN and CATNN to predict flow and density values at the test loop station (# 5).

The extents of prediction into the future were: 30 seconds, and 1, 2, 4, 5, 10, and 15 minutes. An optimization run using the GA to produce a "winner" TDNN or CATNN was made for each prediction extent. The objective (fitness) function used was the Average Absolute Error *(AAE)*, defined as the mean (across all records) absolute value of the difference between the actual value and the predicted neural output:

$$AAE = \frac{\sum |(y_{\text{actual}} - y_{\text{predicted}})|}{n} \tag{5}$$

where:

y_{actual} = actual value of the output in the data set;
$y_{\text{predicted}}$ = predicted output value, and
n = number of records in the data set.

For each of the prediction horizons, the population size and the number of generations used by the GA to optimize the NN were incremented and the effect of the results observed. The population size was incremented from 30 to 300 and the number of generations was incremented from 10 to 30. Based on the observations, combinations of population size and number of generations near the upper limits produced minimal improvement in the objective function. Therefore, the number of generations was set to 30 and the population size to 300 in the remainder of the research.

EFFECT OF EXTENT OF PREDICTION ON PREDICTION ACCURACY

To examine the effect of the extent of prediction on prediction accuracy, data resolution was fixed to 30 seconds and the spatial contribution fixed to full contribution from all loop stations. Then, both the TDNN and CATNN were optimized relative to structure and the look-back interval (temporal contribution). Although the TDNN has a fixed look-back interval (as opposed to the CATNN), the GA examines a number of candidate TDNNs, as chromosomes in the gene pool during optimization, with varying look-back interval settings; and, hence, the look-back interval is indirectly optimized as well. In the case of the CATNN, each network directly varies the look-back interval during training; this inherent feature, together with the variety of networks in the population processed by the GA, results in an optimized look-back interval.

An optimization scenario using the GA was implemented for each prediction horizon of 30 seconds, and 1, 2, 4, 5, 10, and 15 minutes. Interestingly, both network types showed similar performance (the two alternated minor superiority).[1] This could be due to the fact that although the TDNN does not optimize the look-back interval directly, the GA examines several TDNNs, each with a different "fixed" look-back interval. Therefore the look-back interval in the TDNN is also indirectly optimized by the GA as opposed to the network itself, as in the case of CATNN. This similarity in performance is specific though to the genetically optimized TDNN and CATNN. If no GA were employed, the CATNN would have yielded better look-back intervals. The resulting

[1]Therefore, the results reported generally pertain to both networks.

optimal network in each case was validated on the validation data set and the error reported. Figure 3a, for example, shows the predicted and actual flow and densities for the case of a 30-second prediction. Figure 3b shows the same for the other end of the spectrum, a 15-minute prediction.

From the plots of predicted value versus actual values (not all are shown for brevity), it was observed that the predicted flow and density values are quite close to the actual values for up to 2-minutes of prediction (see, e.g., Figure 3a). After two minutes, the predicted values increasingly tend to become closer to the mean of the actual values, which is very evident in the 15-minute extent of prediction shown in Figure 3b. This is in agreement with expectations; as the extent of prediction increases, it becomes increasingly difficult to predict far ahead using 30-second dynamics, and the model resorts to guessing the "average."

Figure 4 shows a summary of the extent of prediction versus the average absolute percentage error, defined as:

$$\text{Average Percentage error value} = \sum \frac{(|(y_{\text{actual}} - y_{\text{predicted}})|^{*}100)}{y_{\text{actual}}} \Big/ N \tag{6}$$

where N is the total number of records for which predictions are made. It can be seen from the figure that the average percentages of error are less than 10% for both flow and density up to about four minutes of prediction extent. After four minutes, the average percentage error values exceed 10% and increase gradually to 15% for 15-minute prediction.

The GA also reports the optimal "look-back" interval for each extent of prediction. The look-back interval is the number of time steps in the past that affected the prediction the most. An interesting pattern of "look-back" intervals was observed. The "look-back" interval was found to decrease as the extent of prediction increased, indicating that the temporal history has less bearing on far predictions—the best guess for which is around the mean values. Figure 5 shows the "look-back" interval versus the extent of prediction.

EFFECT OF SPATIAL CONTRIBUTION ON PREDICTION ACCURACY

In this section, the effect of the extent of upstream and downstream spatial contribution on the prediction accuracy is examined. Spatial

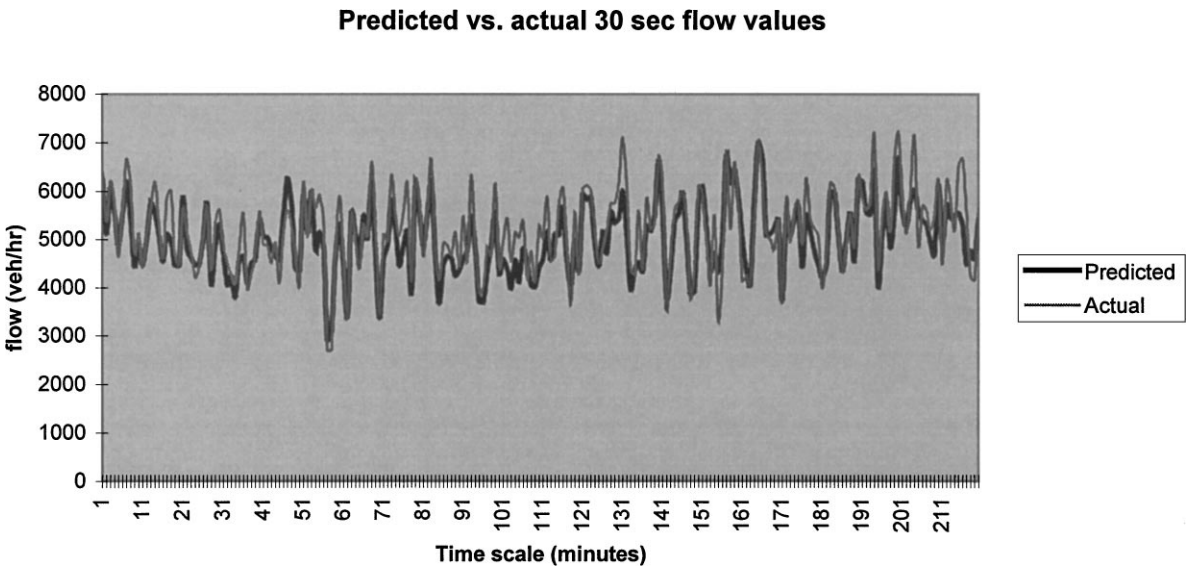


FIGURE 3a. 30 sec. predictions using 30 second data resolution. *(Continued)*

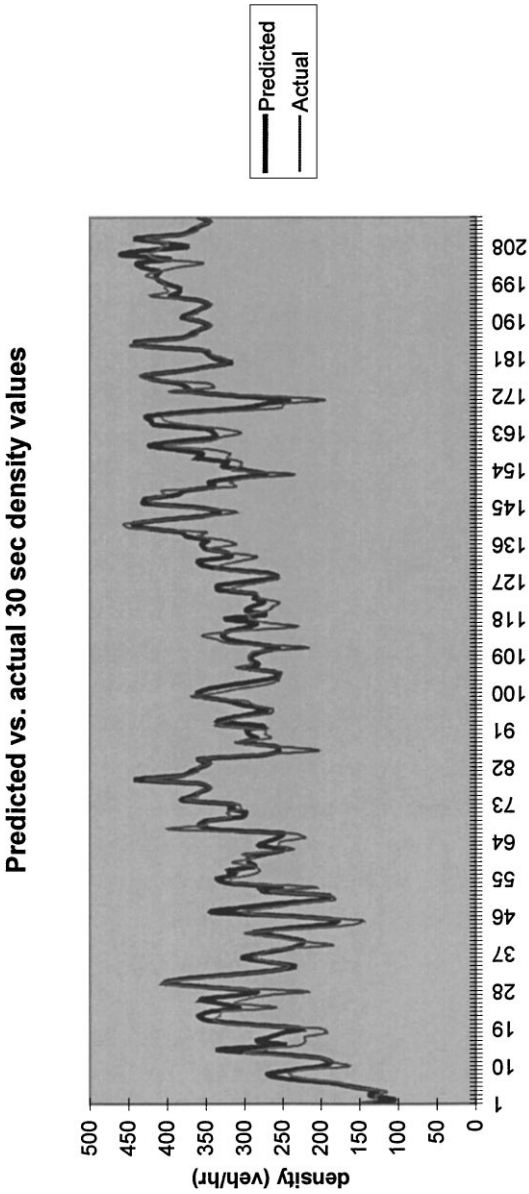


FIGURE 3a. (Continued)

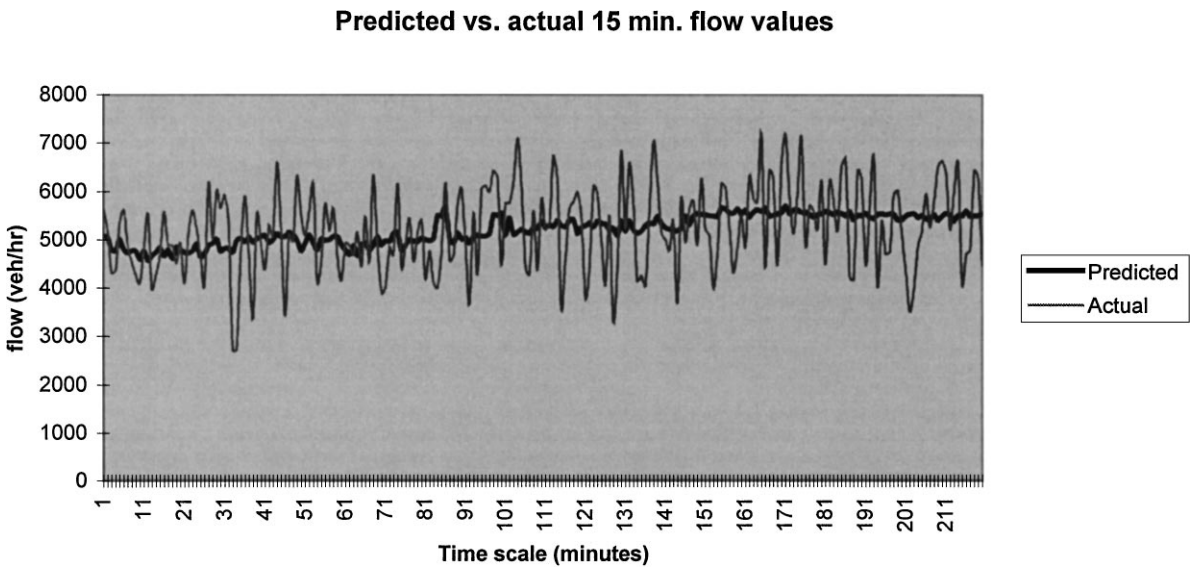


FIGURE 3b. 15 min. predictions using 30 second data resolution. *(Continued)*

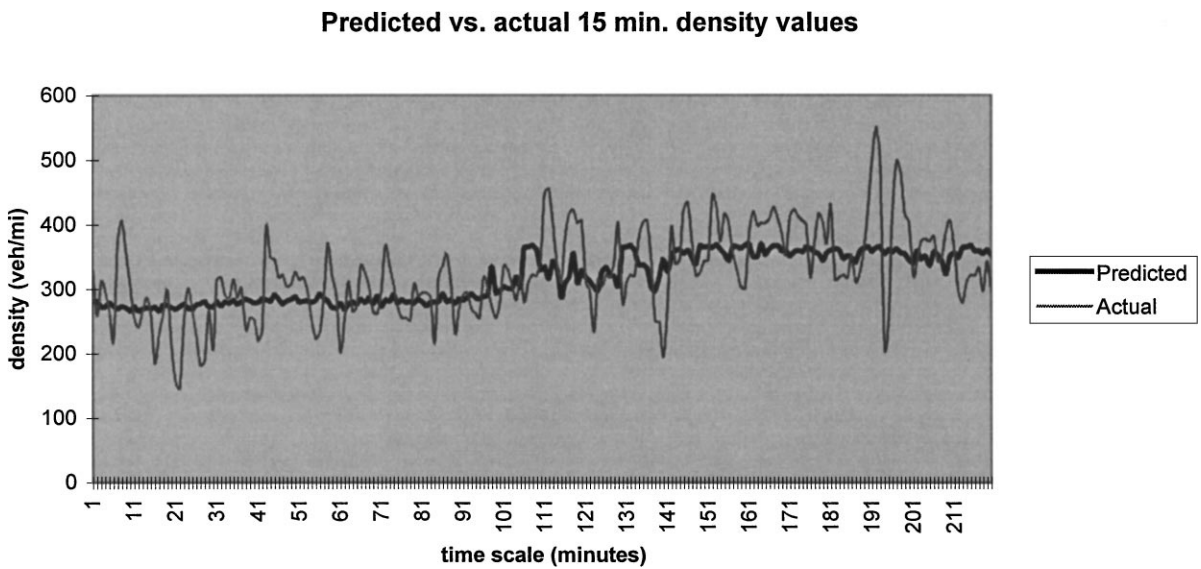


FIGURE 3b. (Continued)

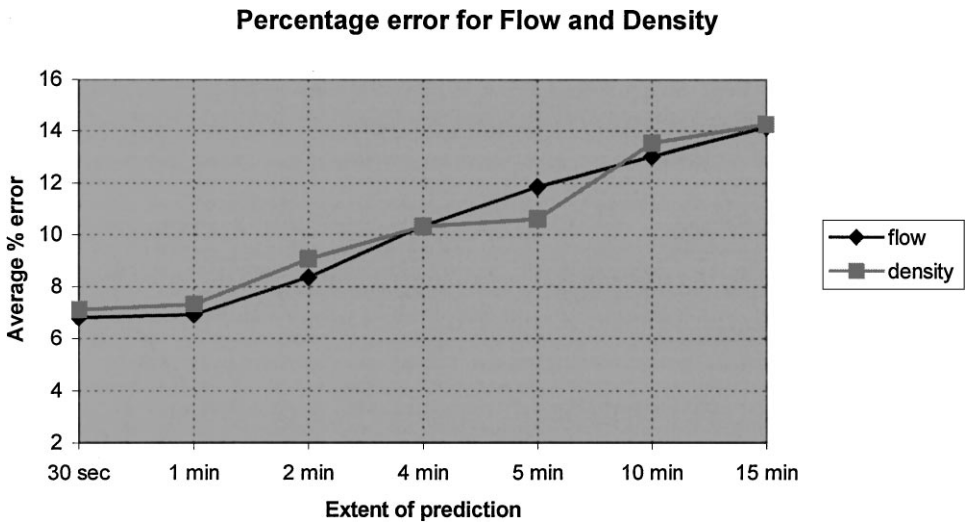


FIGURE 4. Average percentage errors for various extents of prediction using 30 sec. data.

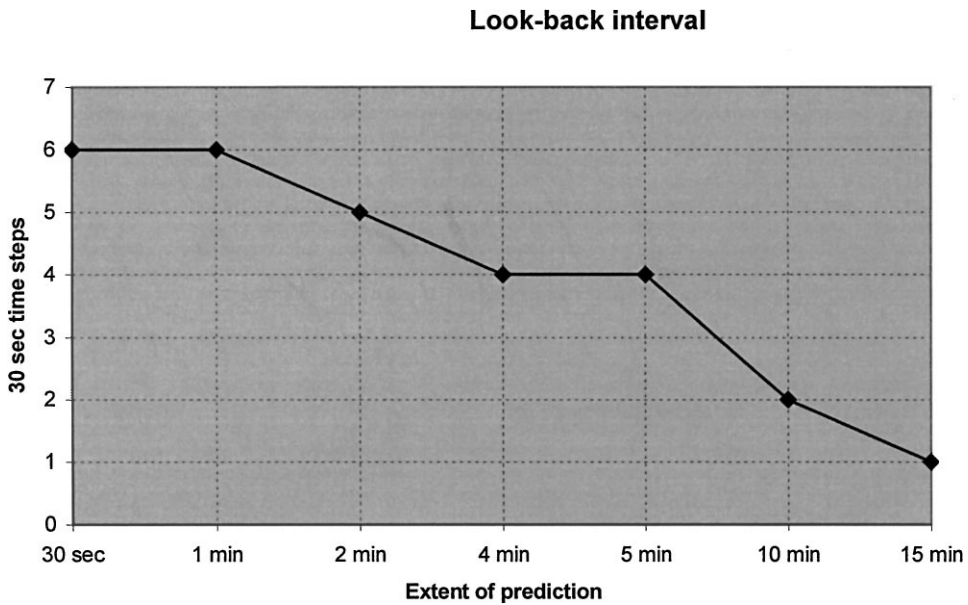


FIGURE 5. Look-back intervals for various extents of prediction using 30 sec. data.

contribution is defined by how many loop detector stations, upstream and downstream the subject station #5, are included in the training. For any given extent of prediction, four data files were prepared. The first data file had the spatial contribution from all the detector stations except the farthest two mainline stations. In the second data file the penultimate stations were dropped as well as any on/off ramp stations in between the farthest and the penultimate station pairs, and so on. The fourth data file included the subject station #5 only (i.e., no spatial contribution, and predictions are based on temporal history only). To keep the number of optimization runs reasonable, the extents of prediction used were restricted to 30 seconds, and 1 minute and 15 minutes only. Figure 6 shows the average absolute error versus the extent of spatial contribution for the three prediction extents, respectively. For contrast, Figure 7 shows the relationship between the extent of prediction and the error for the cases of full spatial contribution and no spatial contribution at all. The following observations can be made:

- The less the spatial contribution the higher the error, as shown in Figure 6.
- Three stations on both sides of the subject loop station are probably sufficient. More so in the case of finer resolution, say, 30-sec data.[2]
- The longer the extent of prediction based on 30-sec data, the less pronounced the effect of spatial contribution, as the spatial information becomes of less use to a model attempting 15-min prediction, for instance, using 30-sec data. Figure 7 shows that the benefit from full spatial contribution as opposed to no contribution at all is much more evident in the case of 30-sec predictions.

EFFECT OF DATA RESOLUTION ON PREDICTION ACCURACY

In this section, the effect of data resolution itself on the accuracy of prediction is examined. The resolutions considered were 30 seconds (the original data), and 1-minute, 2-minute, 5-minute, and 15-minute aggregations of the original data. The extents of prediction employed were multiples of the resolution of data used. For example, for the case of 2-minute resolution, the extents of predictions were 2, 4, 6, 10, and 14 minutes; for 5-minute resolution, the extents of prediction were 5, 10,

[2]It is noted that, in practice, it is often difficult to obtain good loop data from a large number of consecutive loop stations.

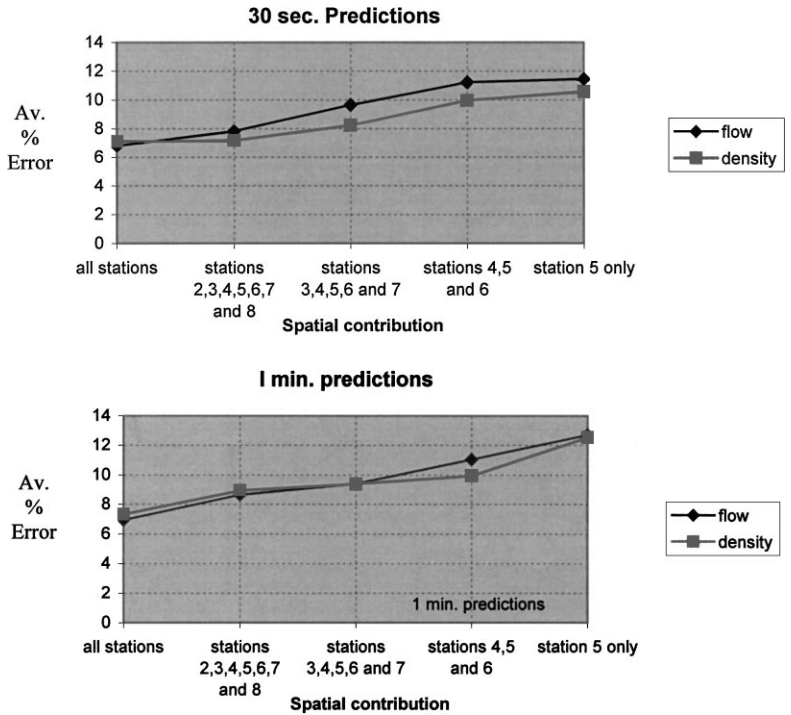


FIGURE 6. Average percentage errors vs. spatial contribution. *(Continued)*

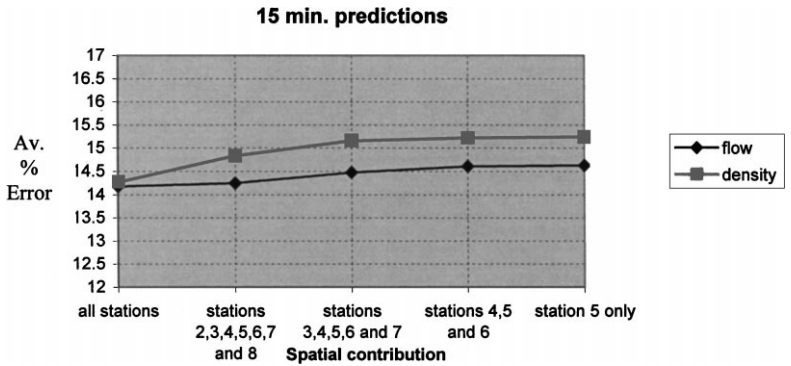


FIGURE 6. (Continued)

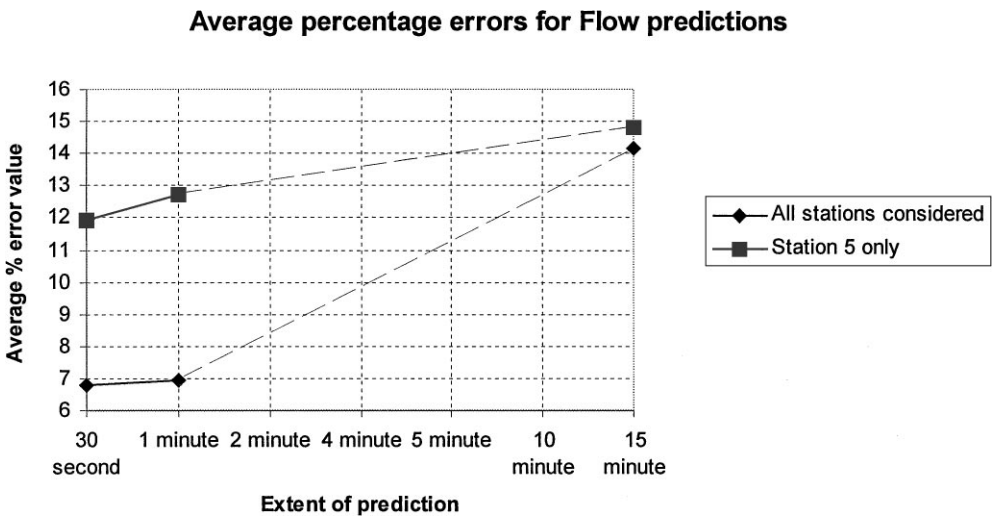


FIGURE 7. Benefit of spatial contribution. *(Continued)*

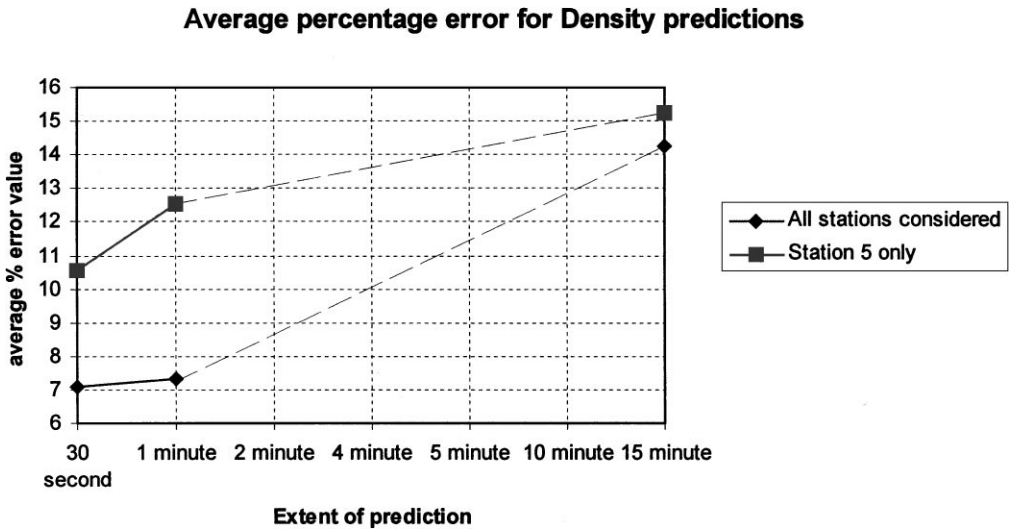


FIGURE 7. (Continued)

and 15 minutes; and so on. Full spatial contribution was assumed from all of the detector stations.

Figure 8 summarizes the errors versus the extent of prediction for all resolutions tested. Figure 9 summarizes the effect of data resolution on prediction error for the case of 15-minute prediction, taken as an example. It can be seen that the *higher* the level of aggregation (the lower the resolution), the *lower* the prediction errors in general for all prediction horizons. This is due to the disappearance of erratic dynamics in the values of flow and occupancy, common at 30-second readings and due to closer fit of the predicted values to the actual. This finding should be carefully interpreted, however. It does not mean that higher levels of aggregation and coarser data are always better, but rather that *higher levels of aggregation are better only for longer prediction horizons*. For instance, if 10-minute predictions are desired, 10-minute resolution is probably best, and so on. That is, *the level resolution generally should be the same as the prediction horizon*. This finding is significant because it has been thought that finer data should lead to better results.

VALIDATION USING FIELD DATA

To validate the models and verify that the above findings from simulated traffic scenarios are applicable to the real world, a validation phase using actual freeway data was conducted. Real 30-second flow and occupancy data for a section of freeway in the proximity of the study site used in the training were obtained; the selection of this site was designed to examine the transferability of the model, as well. The data collected for validation were for the two evening peak hours, that is, 16:00 hr. to 18:00 hr. on 15 November, 1997.[3] The selected site is located on the I-5 North in the cities of San Clemente and San Juan Capistrano in Orange County, California, covering the section of I-5 south of Vaquero up to San Juan Creek. Only the data for mainline stations were used, due to the absence of on/off ramp data from several consecutive stations. Holes in the data were filled using interpolation. Data files were prepared in accordance with the case of simulated scenarios. The extents of prediction used were the same, that is, 30 seconds, and 1, 2, 4, 5, 10, and 15 minutes. The genetic and neural parameters for training and testing were also kept the same.

Figure 10 shows the comparison of the average percentage errors for flow prediction for both real data and simulated data for the case of 30-sec resolution. It can be seen that the model behavior in the real

[3]This combination of site and date was selected after thorough search for good data from consecutive loop stations on different sections of the freeway.

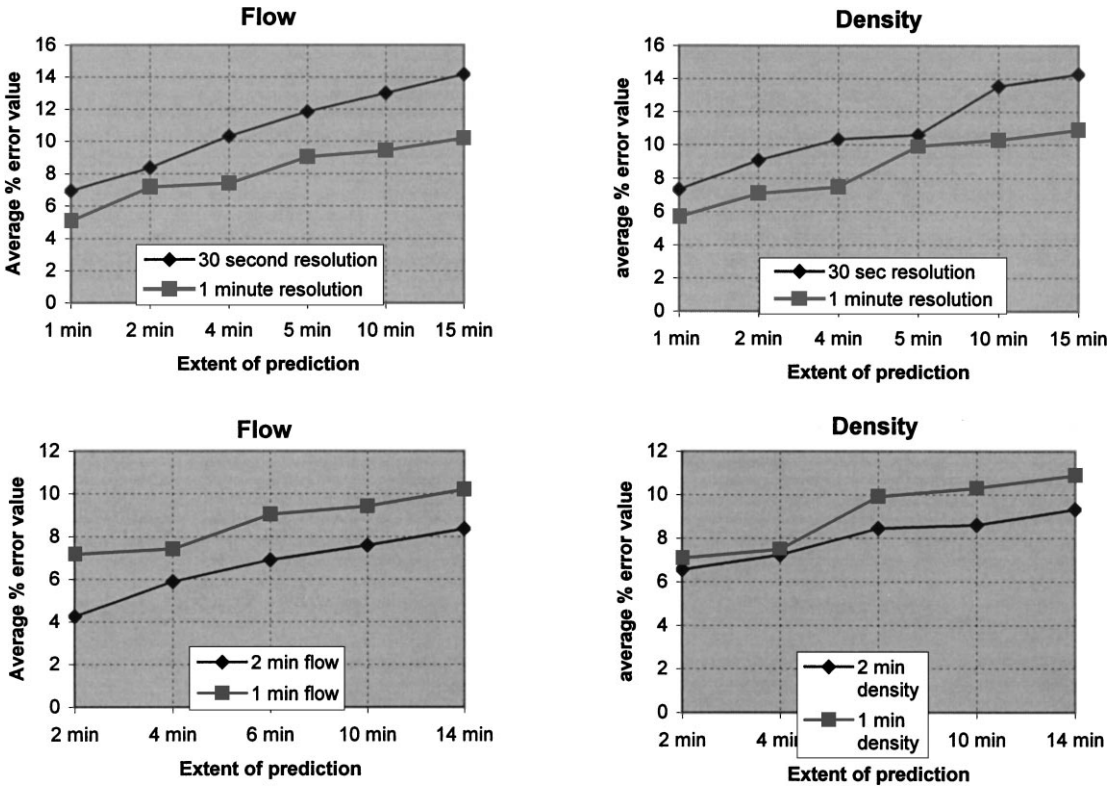


FIGURE 8. Effect of data resolution on prediction accuracy. *(Continued)*

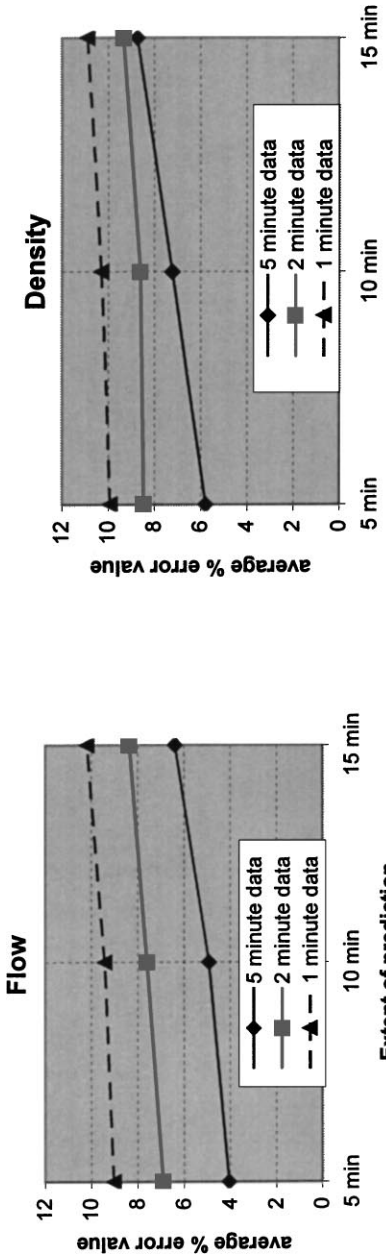


FIGURE 8. (Continued)

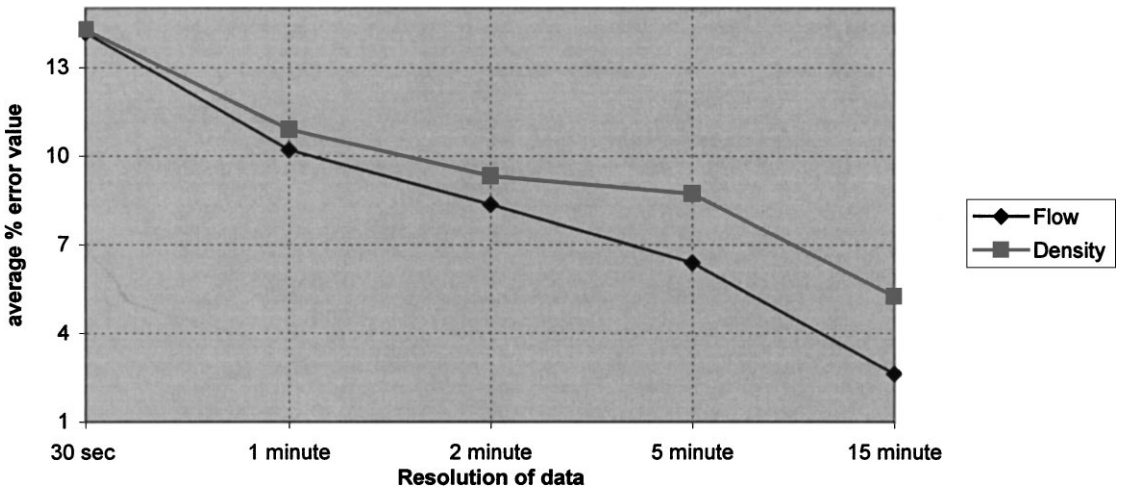


FIGURE 9. Reduction in error with higher data resolution for 15 min. prediction.

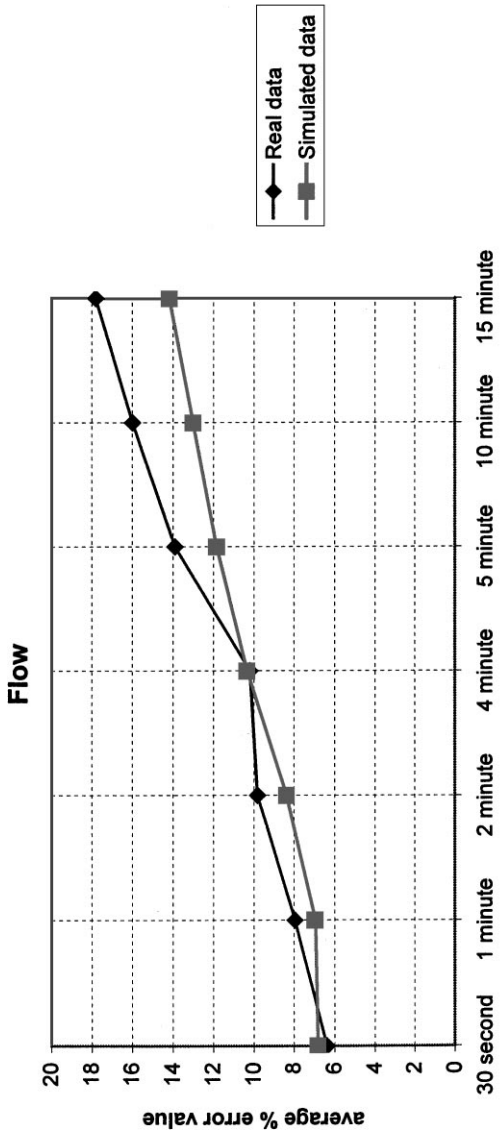


FIGURE 10. Performance using real data compared to simulated data (30 sec.).

world follows the same trends observed with simulated data, confirming the previous findings. The average percentage errors were found to be slightly higher in the case of real field data than for simulated data for a number of cases. Also, the difference between the average percentage error values for real field data and simulated data seem to increase as the extent of prediction increases. This could be attributable both to higher frequency dynamics in the real data as well as to less accuracy due to absence of ramp data and holes in the mainline data. Figure 11 shows similarity in prediction accuracy for flow and occupancy using the real data.

TDNN PERFORMANCE COMPARISON TO THE MLF MODEL

The performance characteristics of variety of TDNNs developed in this research were compared to a particular example of the widely used Multi Layer Feed Forward model reported by Zhang and Ritchie (1997). Using simulated speed, density, and ramp volumes (each with a resolution of 15 seconds) from a single neighboring station as input, they employed a MLF NN model to predict a 15-second event horizon. To facilitate cross comparison, the Zhang and Ritchie model was replicated using the same data used in this research, and employing a 30-second resolution instead of the 15 seconds used in their original study.

Figure 12 shows the relative performance of the two models. Only for the case in which the TDNN has no spatial contribution (i.e., has only a temporal component) is the average percentage error of the TDNN greater than the MLF (which has both a temporal and spatial component); and for this case the results are roughly comparable. For the case in which both models have the same spatial and temporal contributions, the TDNN outperforms the MLF; for all other cases, the average percentage error for the MLF is significantly higher than the TDNN. Ostensibly, the superiority of the TDNN is attributable to its ability to "look-back" over time and select the optimal temporal contribution. Also, significant improvement in the prediction accuracy of the TDNN with full spatial contribution is evident. This is clearly indicative of the significance of both the temporal and spatial contributions to the prediction of spatio-temporal traffic patterns.

SUMMARY AND CONCLUSIONS

This paper has presented a short-term traffic flow prediction approach and produced a system based on an advanced Time Delay Neural Network (TDNN) model synthesized using Genetic Algorithms (GA). The model predicts flow and occupancy values based on their recent temporal profile at a given freeway site during the past few minutes

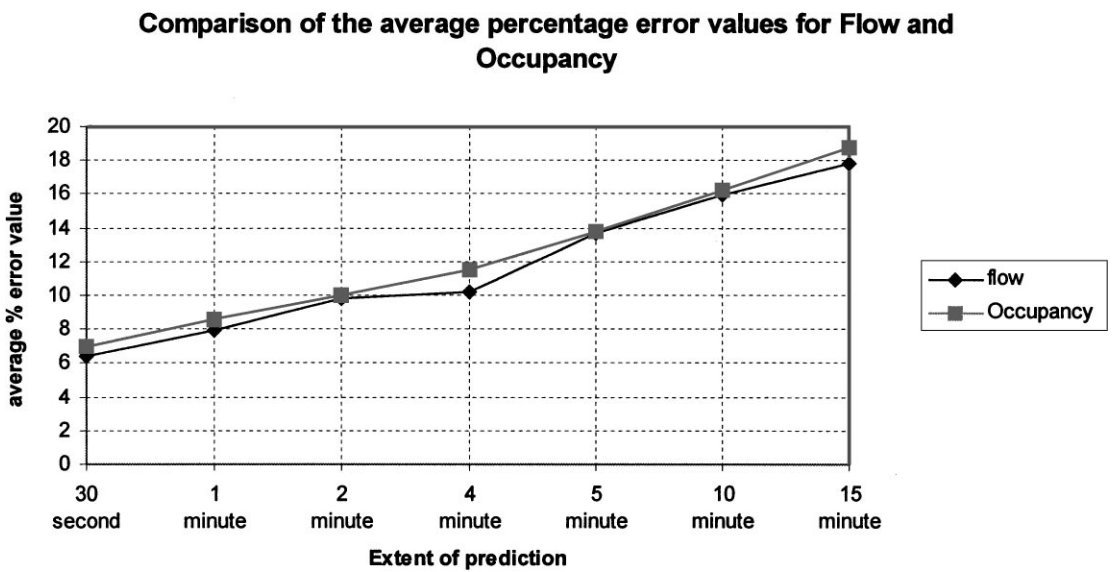


FIGURE 11. Comparison of flow and occupancy prediction accuracy using 30 sec. real data.

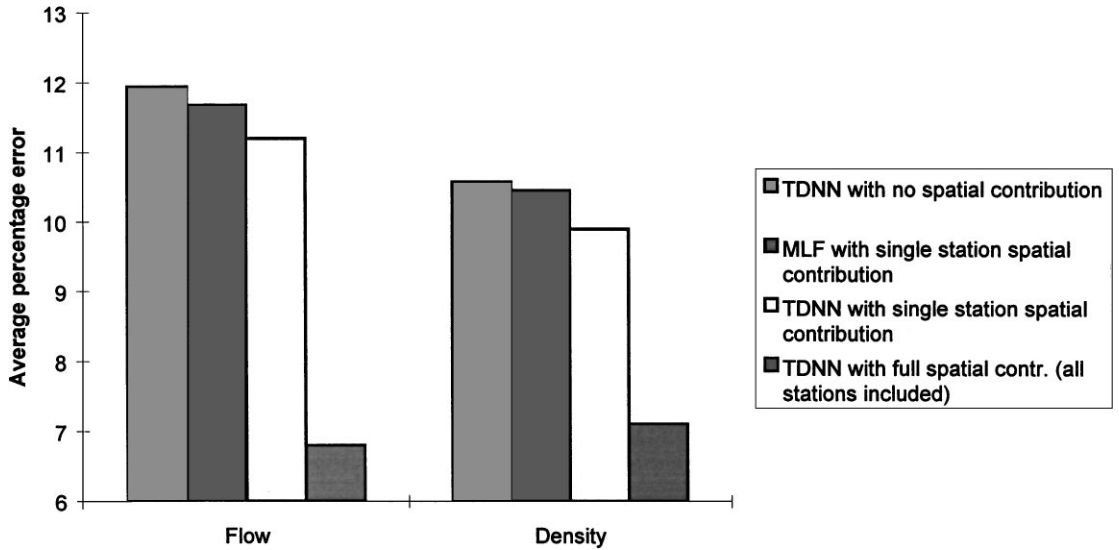


FIGURE 12. Comparison of the TDNN to the MLF.

as well as the spatial contribution from neighboring sites. The model performance was validated using both simulated and real traffic flow data obtained from the field. Both temporal and spatial effects were found essential for proper prediction. An in-depth investigation of the variables pertinent to traffic flow prediction was conducted; the extent of the look-back interval, the extent of prediction in the future, the extent of spatial contribution, the resolution of the input data, and their effect on the prediction accuracy. Obtained results indicate that the less the spatial contribution the higher the prediction errors. For the case of 30-second data for instance, prediction errors almost doubled (from 7% to 12% as in Figure 6) when spatial information was omitted and prediction is based solely on temporal trend. It was also found that the inclusion of three loop stations in both directions from the subject station is sufficient for practical purposes, keeping prediction error within 10%. Moreover, it was found that the longer the extent of prediction, the more the predicted values tend toward the mean of the actual for a given data resolution. The significance of selecting the optimal look-back interval also shortens due to become increasingly irrelevant with increasing extents of prediction. For the case of 30-sec data for instance, prediction for only another 30-sec in the future was 93% accurate and required six 30-sec look-back intervals (three minutes total), while prediction for 15 minutes in the future using the same 30-sec data was only 86% accurate and required only one 30-sec look-back interval. Interestingly, results revealed that *coarser* data resolution is better for *longer* extents of prediction. For instance, if 10-minute prediction is desired, it is better to use 10-minute average data than to use 30-sec data. The implication is that the level of data aggregation/resolution should be comparable to the prediction horizon for best accuracy. The model performed acceptably using both simulated and field data, bearing in mind that the real-data used is from the mainline only, that is, it does not include ramp data. The model also showed potential to be superior to such other well-known neural network models as the MLF has been shown in the literature to outperform other conventional statistical models.

REFERENCES

Abdulhai, B. and Ritchie, S. G., "Enhancing the Universality and Transferability of Freeway Incident Detection Using a Bayesian-Based Neural Network," *Journal of Transportation Research—Part C, Emerging Technologies*, 7, 261–280, 1999a.

Abdulhai, B. and Ritchie, S. G., "Towards Adaptive Incident Detection Algorithms," 6th World Congress on Intelligent Transportation Systems, Toronto, November 8–9, 1999b.

Abdulhai, B. and Ritchie, S. G., "A Preprocessor Feature Extractor and a Postprocessor Probabilistic Output Interpreter for Improved Freeway Incident Detection," Transportation Research Board, Washington DC, 1998. Also in *Journal of the Transportation Research Record*, TRR #1678, 1999c.

Abdulhai, B., "A Neuro-Genetic-Based Universally Transferable Freeway Incident Detection Framework," Ph.D. Diss., University of California Irvine, 1996.

Beale and Jackson, "Neural Computing: An Introduction," Bristol MA and Philadelphia, PA: Institute of Physics Publishing, 1992.

Belew, R., McInerney, J., and Schraudolph, N., "Evolving Networks: Using the Genetic Algorithms with Connectionist Learning," *CSE Technical Report* CS90-174, Computer Science, UCSD, 1990.

Biocomp Systems Inc., *Neuro Genetic Optimizer, User's Manual, Version 2.5*, 1997.

Chambers, L., "Practical Handbook of Genetic Algorithms," Vol. I & II, New York: CRC Press, 1996.

Chang, E. J. and Lippmann, R. P., "Using Genetic Algorithms to Improve Pattern Classification Performance," Lippmann, R. P., Moody, J. E., and Touretsky, D. S. (eds.), *Advances in Neural Information Processing* 3, 797–803, Morgan-Kaufmann, San Mateo, California, 1991.

Daganzo, Carlos F., "The Cell Transmission Model: A Dynamic Representation of Highway Traffic Consistent with the Hydrodynamic Theory." *Transportation Research*, 28B, 269–287, 1995.

Davis, G. and Nihan, N., "Nonparametric Regression and Short-Term Freeway Traffic Forecasting," *Journal of Transportation Engineering*, 117, 178–188, 1991.

Dougherty, M., Kirby, H., and Boyle, R., "The Use of Neural Networks to Recognize and Predict Traffic Congestion," *Traffic Engineering and Control*, pp. 311–314, 1993.

Dougherty, M. and Lechevallier, Y., "Short-Term Road Traffic Forecasting Using Neural Network," *Recherche Transports Securite, English Edition* 11, 73–82, 1995.

Harp, S. A. and Samad, T., "Genetic Synthesis of Neural Network Architecture," Davis, L. (ed.), *Handbook of Genetic Algorithms*, 202–221, Van Nostrand, New York, 1991.

Haykin, S., *Neural Networks, a Comprehensive Foundation*. Prentice Hall, New Jersey, 1994.

Hinton, G. and Waibel, A. et al. "Phoneme Recognition Using Time-Delay Neural Network." *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37, 328–339, 1989.

Koza, J. R. and Rice, J. P., "Genetic Generation of Both the Weights and Architecture for a Neural Network," International Joint Conference on Neural Networks, Baltimore, MD, June 7–11, 1992, 1992.

Kunhe, R. D., "Freeway Control and Incident Detection Using a Stochastic Continuum Theory of Traffic Flow," Proc. 1st Int. Conference on Applied Advanced Technology in Transportation Engineering, San Diego, pp 287–292, New York: ASCE, 1989.

Lighthill, M. J. and Whitham, G. B., "On Kinematic Waves: II. A Theory of Traffic Flow on Long Crowded Roads." *Royal Society, London Series A* (229), 1178, 317–345, 1955.

Michalopoulos, P. G., Yi, P., and Lyrintzis, A., "Development of an Improved Higher-Order Continuum Traffic Flow Model for Congested Freeways." Presented at the 70th Annual Conference of TRB, Washington, DC, Jan. 12–16, 1992.

Montana, D. J. and Davis, L., "Training Feedforward Neural Networks Using Genetic Algorithms," Proceedings of the 11th International Joint Conference on Artificial Intelligence, 762–767, Detroit, MI, August 20–25, 1989.

Paramics Traffic Simulator "User Manuals," Quadstone Limited, Edinburgh, U.K., 1998.

Payne, H. J., Models of Freeway Traffic and Control. Simulation Councils Proceedings Series, 51–60, 1971.

Phillips, W. F., "A New Continuum Model Obtained from Kinematic Theory," *IEEE Trans. Autom. Control* AC—23, 1032–1036, New Jersey, 1978.

Rathi, A. K., Lieberman, E. B., and Yedlin, M., "Enhanced FREEFLO Program: Simulation of Congested Environments," *TRR*, 1112, 61–71, 1987.

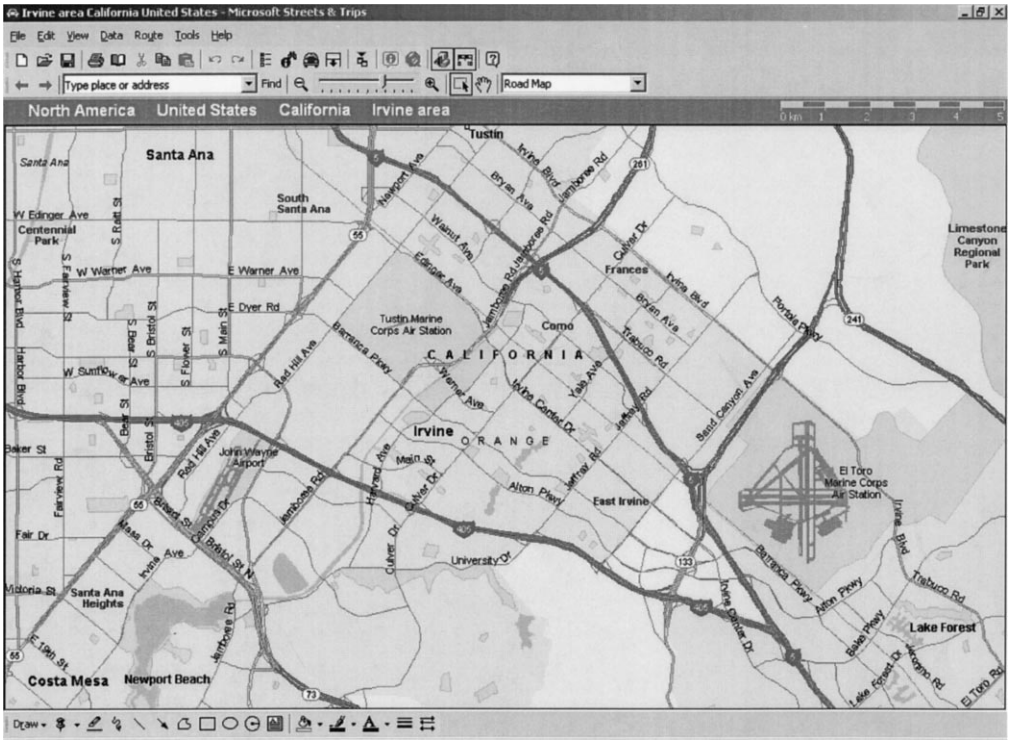
Ritchie, S. G., Zhang, H., and Lo, Z.-P., "Macroscopic Modeling of Freeway Traffic Using an Artificial Neural Network," *Transportation Research Record* 1588, 110–121, 1997.

Ross, P., "Traffic Dynamics," *Transportation Res. B*, Vol. 22B(6), 421–435, 1989.

Smith, B. and Demetsky, M., "Short-Term Traffic Flow Prediction: Neural Network Approach," *Transportation Research Record* 1453, 98–104, 1995.

Winter, G., Periaux, J., Galan, M., and Cuesta, P., *Genetic Algorithms in Engineering and Computer Science*, John Wiley & Sons Publishing, New York, 1995.

APPENDIX A.



Study site map.