Modeling Schedule Deviations of Buses Using Automatic Vehicle-Location Data and Artificial Neural Networks

RAVI KALAPUTAPU AND MICHAEL J. DEMETSKY

The establishment of the Advanced Public Transportation Systems program has encouraged bus transit operators to experiment with implementing automatic vehicle-location systems for real-time monitoring and supervision of operations. While the focus has primarily been on the implementation of technologies, such as automatic vehicle-location systems, it is necessary to experiment and develop advanced performance analysis and evaluation procedures that can assist in schedule planning and real-time service-control tasks. One potentially useful and effective approach to these tasks is system behavior modeling. In this study this method is used to model schedule behavior of buses on a route using schedule-deviation information. The primary objective of this study is to investigate the application of artificial neural networks, which have been shown to hold promise when applied to nonlinear dynamic system-modeling problems, for developing schedule behavior models. Models are developed using the schedule-deviation information obtained from Tidewater Regional Transit's automatic vehicle-location system. The time-series analysis approach is adopted for the development of schedule behavior models at the route level. The results of a case study are encouraging and demonstrate the usefulness of artificial neural network techniques, especially the Jordan networks and the Elman networks, for modeling schedule deviations of buses on a route.

In recent years, bus transit operators have been testing and implementing automatic vehicle location (AVL) systems for real-time monitoring and supervision of operations. However, implementation of technologies such as AVL systems needs to be complemented by the development of advanced performance analysis and evaluation procedures for assisting in operational planning, management, and real-time service-control tasks. While real-time monitoring provides useful information on bus transit operations, advanced analysis and evaluation procedures such as system behavior models can be useful for schedule planning and design of real-time service-control strategies.

The central idea of this study is that a schedule behavior model can provide an understanding of the past system behavior of buses on a route. Such a model has several potential uses. It can be used for prediction of schedule deviations at a downstream stop based on current and past schedule deviations of buses at timepoints in the upstream section. The predictive model can assist in the design, development, and real-time implementation of service-control strategies. Also, the schedule behavior models can be used for updating and modifying schedule plans. The models can be used to speed up and automate performance analysis and evaluation of service-control strategies. Currently there are no automated procedures available to evaluate the effect of implementing service-control strategies. The models can be integrated into an automated decision-support system to assist dispatchers and supervisors in real-time decision making on schedule and headway adjustments to improve service reliability.

Department of Civil Engineering and Applied Mechanics, Thornton Hall, University of Virginia, Charlottesville, Va. 22903.

In real-time operating conditions, the time required to make decisions based on graphical display codes and other features of a number of buses is not sufficient to make reasonable decisions regarding schedule or headway changes. Dispatchers and supervisors have to make quick decisions based on the information presented graphically on the screen. However, they must monitor several buses simultaneously, leading to information overload. Hence there is a need to develop computer-based analysis and decision-support tools to model the system's performance and use it to predict the future schedule behavior of a bus on a route. The key purpose of this study is to introduce the concept of schedule behavior modeling as a performance analysis tool for bus transit operations. The primary objective is to investigate the development of schedule behavior models using historical AVL data and artificial neural network (ANN) techniques. In this paper, system behavior is referred to as schedule behavior of buses and is used to denote the key performance indicator, schedule deviation, which is the difference between the actual arrival times computed from the location information and the scheduled arrival times of a bus at a timepoint on a specific route.

ANN modeling techniques have been of great interest to many researchers. These techniques have certain advantages such as not requiring to assume a priori the nature of the relationship between the dependent and independent variables. The modeling approach using neural networks performs two important tasks. First, the model learns the system performance using past and current AVL data. Secondly, the ANN models can be used for predicting the behavior of the buses. Such a system behavioral modeling approach has been successfully used in other dynamic-system performance analysis and control problems (*1,2,3*). The literature reviewed indicated that ANNs have the potential to capture the dynamic and interactive effects of schedule deviations of buses on a route network. In addition, they are able to capture the trend in a time series, especially when the relationship is nonlinear.

The basic approach adopted for ANN modeling of the performance of a bus transit system was to develop separate models for the different routes instead of one complete model for the entire transit route network. By using this approach the ANN modeling process becomes simpler and the training process is perhaps faster because of its reduced complexity: there is a smaller domain space to learn for one route, compared to learning all the routes in the transit network. In addition, such a modeling approach is appropriate and justified by the different physical, traffic, and environmental characteristics of the various routes. Modeling at the route level can help reduce the complexity of the modeling process and simplify and ease the implementation of the service-control strategies. In addition, such a modeling approach can help reduce the time required for system identification, and subsequent selection and implementation of a service-restoration strategy.

A number of ANN architectures and learning algorithms have been proposed and investigated for various problems. Since the primary objective of this study is to illustrate the applicability of the ANN approach for the problem of bus schedule behavior modeling, we discuss only the advantages of ANNs and the applicability of alternative strategies to developing ANN models for this problem. The fundamental concepts of ANNs are discussed in detail in the vast collection of relevant literature (*4,5,6,7,8,9*).

ARTIFICIAL NEURAL NETWORKS

ANNs are a type of learning system that has gained some prominence in the last decade because they can be trained to identify, classify, and predict nonlinear patterns and can solve complex problems much faster than traditional techniques. ANNs are a paradigm for intelligent processing of information for some specific objective such as classification, pattern recognition, decision-making, system behavior identification, and prediction. ANNs have a highly distributed parallel structure and when combined with powerful digital hardware technology can make model simulations economically and with relative ease. ANNs mimic human learning processes and therefore hold great potential as adaptive learning systems. ANNs can handle complex and nonlinear relationships that are common to dynamic systems like bus transit operations. In the case of nonlinear systems, ANNs have the distinct advantage over a standard regression method of not having to know the form of the function a priori. Unlike other mathematical techniques, ANN models' learning can be continuous, so that they can automatically adapt to the changing characteristics of the operating environment of buses. What this implies is that a base ANN model can be developed using historical AVL data and this base model can be updated and modified using new online data. The potential advantage of an ANN learning method is that, unlike mathematical simulation models, ANNs can be trained using observed data only, without requiring any knowledge of the internal structure of the system or of modeling techniques (*10*). This ability to approximate unknown functions through the presentation of past states of a system makes ANNs a useful modeling tool in engineering applications, such as bus transit schedule behavior modeling.

Lapedes and Farber (*1*) reported that simple neural networks can outperform conventional methods. Sharda and Patil (*11*) concluded from their work on 75 different time series that the simple neural network model could forecast about as well as the Box-Jenkins forecasting technique. Tang et al. (*12*) in their comparative study of the performance of ANNs and conventional statistical techniques concluded that for short-term memory series, ANNs appear to be superior to the Box-Jenkins model. A review of relevant literature indicated that each of the methods performed better than the other about half of the time.

In this study the focus is on using three different ANN architectures, namely feedforward networks with input windows, Jordan nets, and Elman nets. Jordan and Elman nets are two types of partial recurrent neural networks. These three network architectures have been feasible for modeling a number of engineering problems, such as system behavior identification and prediction and time-series modeling, among others (*1,2,3,12,13*). Hence our initial efforts are aimed at investigating these three ANN architectures.

The main distinction between the feedforward and the partial recurrent nets is in the network topology. The two types of partial recurrent nets (Jordan and Elman nets) have memory layers in addition to the basic architecture of a feedforward network. These network architectures are discussed briefly in the next two sections.

Feedforward Networks

Feedforward networks are the most commonly used network architectures for neural network modeling. Depending on the representation scheme, feedforward networks can be different types. Figure 1 illustrates the schematic architecture of a feedforward network with an input window. The most basic approach for handling time series is using an input window that holds a restricted part of the time series. This type of feedforward network seems appropriate for our modeling problem. A feedforward network with an input window has been shown to be superior to a simple feedforward network (*2,13,14*). The input window provides the network with information on previous states in the form of units in the input layer. This allows it to incorporate knowledge about previous states or past values of a time series. Therefore such an architecture is suitable for modeling spatiotemporal sequencing problems such as bus schedule behavior.

Partial Recurrent Neural Networks

A second way that a neural network can model and predict a time series is to incorporate an internal state that enables it to learn the relationship of an indefinitely large set of past inputs to future states. This is achieved via recurrent connections, and such a network is known as a recurrent network. If the recurrent networks are updated like feedforward networks (with a single update per time step) they are known as partial recurrent networks (*5*).

Partial recurrent networks have been suggested and proven to be applicable by many researchers (*7,8*) for dynamic problems involving temporal sequencing. The problem of bus schedule behavior prediction can be considered a spatiotemporal problem. The sched-

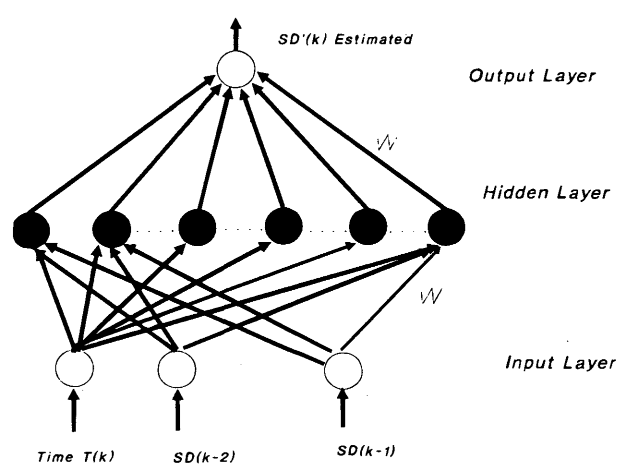

FIGURE 1 Architecture for a feedforward network with input windows.

ule deviation at a point in time is affected by the schedule deviation at previous timepoint(s). The spatiotemporal sequencing of the schedule-deviation information can be modeled and investigated for the purpose of predicting the schedule deviations at a timepoint downstream in the route network. This sequential information, regarded as short-term memory of the system's performance, can be an effective approach for developing an intelligent model of the bus transit schedule behavior. Partial recurrent networks, through their architecture, have the ability to store and use information about the previous state, and therefore are appropriate for the problem of bus schedule behavior modeling.

Jordan Networks

Jordan networks (*7*) are a type of partial recurrent neural network. Figure 2 illustrates the basic architecture for a Jordan network. The network has the following features:

- The input layer is fully connected to the hidden layer, and the hidden layer is fully connected to the output layer.
- Output units are connected to context units by recurrent one-to-one connections. Every context unit is connected to itself and also to every hidden layer unit.
- The number of context units is equal to the number of output units.

A partial recurrent network has an input consisting of two components. The first component is the pattern vector, which is also the only input to the partial recurrent network. The second component, the state vector, is given through the next-state function in every step. In this manner the behavior of a partial recurrent network can be simulated with a feedforward network that receives the state not implicitly through recurrent links, but as an explicit part of the input vector (*7*). These networks are regarded as having memory, as the recurrent connections allow the network's hidden units to see its own previous output. Therefore, behavior can be shaped by previous responses. This network memory concept can be used to model the schedule behavior of buses. The knowledge of schedule deviation of a bus at the previous timepoint or stop can be useful for developing a model of the system for eventual use as a prediction tool. The adoption of such a structure to the ANN model is appropriate for the bus schedule behavior problem because the schedule deviation at a timepoint has a strong relationship to the schedule deviations at the previous timepoints. The extent of previous timepoints that should be considered is yet to be researched. The approach of this study is to take advantage of these features of Jordan nets and investigate their applicability to schedule behavior modeling.

Elman Recursive Networks

An Elman recursive network is a type of partial recurrent network that is also commonly used for learning to recognize and generate sequences of inputs. The Elman net, in addition to the basic topology of a single-hidden-layer feedforward network, has a set of additional units at the input level that are referred to as context units. These context units are responsible for the dynamic behavior of the network. A typical architecture of an Elman recursive network is illustrated in Figure 3. The number of context units is equal to the number of hidden units. After each time step, the output values of the hidden units are copied to the context units. The context units thus provide the network with memory of the previous state through implicit representation in the internal state of the network (*8*). The important distinction between Jordan nets and Elman nets has to do with where the context units are present. The two networks are both essentially memory models, but they differ in whether they have the previous state's inputs or outputs in the memory.

SCHEDULE BEHAVIOR MODELS USING ANNS

Modeling Approach

In prediction modeling there are two basic approaches that have gained prominence and are often adopted. With the fundamental approach, it is believed that the forecasting process should at least approximately model the mechanisms that underlie the determination of the key variable being predicted (*13*). The key factors that affect schedule behavior and cause schedule deviations are:

$$SD\,(R_i, j, k, T) = \phi\,(\text{Traffic, Driver, Vehicle, Environment, Loading and Unloading}) \quad (1)$$

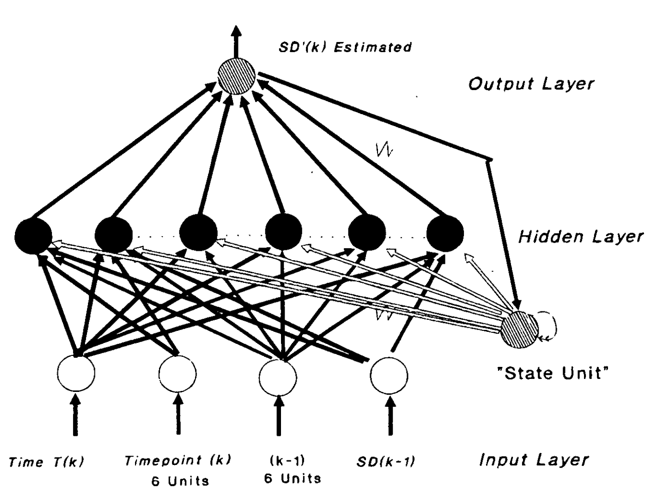

FIGURE 2 Architecture of a Jordan network.

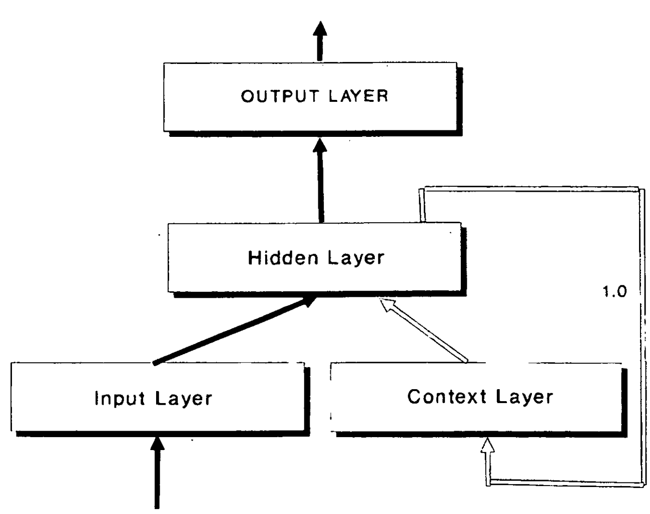

FIGURE 3 Architecture of an Elman network.

where

SD = Schedule Deviation,
R_i = Route i,
j = Direction,
k = Timepoint,
T = Scheduled Arrival Time, and
ϕ represents an unknown function that the network would try to ascertain during the training process.

System behavior modeling on this approach is currently not feasible due to lack of adequate information in the data set on many of the above factors that affect schedule behavior. For example, no information is collected on loading and unloading characteristics at each timepoint on a given route. The second modeling approach is to assume that all the available information (on key factors affecting schedule behavior) has already been represented by the values of the key variable being predicted (*13*). For example, with schedule deviation prediction, the values that indicate "early or late" have been influenced by the various factors that affect it, namely traffic conditions, driver characteristics, passenger loading and unloading characteristics, and vehicle condition. Therefore nothing else is considered while trying to predict the future of the system behavior except the past states of the key prediction variable, the schedule deviation. Hence a time-series approach is adopted that is mathematically represented as follows:

$$SD(k) = \phi\,[SD(k-1), SD(k-2), \ldots, SD(k-n)] \qquad (2)$$

where $SD(k)$ denotes the schedule deviation at timepoint k on a specific route and in a specific direction of travel. The term n represents the length of the input time series, or in other words, the short-term memory about the schedule deviations of a bus at timepoints in the upstream part of a route ($k-1, k-2, \ldots$, etc.).

The focus of this study is on developing ANN models for one particular scenario, that is, given a particular route and direction of travel. Two different ANN model sets, depending on the length of short-term memory about the time series ($n = 1$ and $n = 2$) provided, are investigated. For Model Set I, the schedule deviation at the previous timepoint $[SD(k-1)]$ is provided, while in the Model Set II two previous schedule deviation values $[SD(k-2), SD(k-1)]$ are provided. For each of these model sets, two different cases (Case A and Case B) are examined. The difference between the two cases is that in case B, the spatial information about the timepoints ($k, k-1, k-2$) are also provided to the network as inputs. The distinction between the two cases is illustrated in the input layer of Figures 1 and 2. This was done in order to investigate the effect of providing information about the spatial location of the buses on the route to the ANNs. The two sets of models are briefly described in the following section.

Model Set I: Using Short Input Series of Length $n = 1$

Case A

Input units: Schedule Arrival Time $T(k)$, Schedule Deviation $SD\,(k-1)$;
Output unit: Schedule Deviation $SD(k)$.
$SD(k) = \phi\,[SD(k-1)]$

Case B

Input units: Scheduled Arrival Time $T(k)$, Timepoint k, Timepoint $k-1$, Schedule Deviation $SD(k-1)$;
Output unit: Schedule Deviation $SD(k)$.
$SD(TP_j) = \phi\,[SD(k-1), k, k-1]$

Model Set II: Using Short Input Series of Length $n = 2$

Case A

Input units: Scheduled Arrival Time $T(k)$ Schedule Deviation $SD(k-2)$, Schedule Deviation $SD(k-1)$
Output unit: Schedule Deviation $SD(k)$.
$SD(k) = \phi\,[SD(k-1)]$

Case B

Input units: Scheduled Arrival Time $T(k)$, Timepoint k, Timepoint $k-1$, Timepoint $k-2$ Schedule Deviation $SD(k-2)$, Schedule Deviation $SD(k-1)$;
Output unit: Schedule Deviation $SD(k)$.
$SD(k) = \phi\,[SD(k-1), k, k-1]$

CASE STUDY: A SAMPLE ROUTE FROM TIDEWATER REGIONAL TRANSIT

Data Collection

In order to examine the concept of system behavior models and to investigate the application of neural networks to their development, real data from Tidewater Regional Transit's (TRT's) AVL system was obtained. A sample route (Rt. 23) was chosen for this study. The raw data stored in the form of binary files in the TRT's VAX system was converted into ASCII format. The history data files were preprocessed to extract only the desired information for developing ANN models. AVL information comprising 26 weekday (Monday through Friday) data was considered for modeling purposes. The focus is limited to weekday operations since insufficient weekend (Saturday and Sunday) data was collected.

Modeling Process

The ANN models were developed using the following procedure.

Step 1: Data preprocessing
Step 2: Network selection
Step 3: Learning algorithm and update function selection
Step 4: Weights initialization
Step 5: Network training
Step 6: Network testing and performance evaluation

These steps are discussed in detail in the following sections.

Data Preprocessing

Data preprocessing is the critical step in ANN modeling. In this case it covered about half of the modeling process. Data preprocessing involved two important steps: elimination of outliers or noise, and data scaling. Noise elimination involved removing "outliers" or absurd values of schedule deviation at that specific timepoint and replacing them with the value of schedule deviation from the timepoint immediately preceding it. The data was normalized using minimum and maximum values of the variables over the entire data set. The scaling of these two variables was accomplished using the following expression:

$$Xnorm = \left(2.0^* \frac{X}{MAX - MIN}\right) + \left[\left(-2.0^* \frac{MIN}{MAX - MIN}\right) - 1.0\right] \quad (3)$$

where X is the variable to be normalized, and MAX and MIN denote the maximum and minimum values of variable X in the data set.

In this study, for the scheduled arrival time (T) variable, MAX = 1440 min and MIN = 300 min. The scaling using the above expression, converts the data into the $[-1, 1]$ interval. It is important to set the scaling so that the units do not affect the net's output (that is, the inputs should be either unitless ratios or else chosen so that percentage changes are the same across monotonic transformations of input values). Having most or all inputs scaled identically to the output function can speed convergence. Normalization of the output data to the $[-1, 1]$ region prevents the propagation of large error signals during training, which could force the middle-layer nodes to saturate and become insensitive to training. The output variable, schedule deviation, was also normalized using the expression given in Equation 3 and the corresponding schedule-deviation values. The timepoint data was also transformed into a binary vector. There were six timepoints located on the route being studied. Therefore, a vector of length 6 was considered and the timepoints were transformed. For example, timepoint $k = 1$ was binarized as [1 0 0 0 0 0]. The data set consisting of 26 weekday AVL data was divided into three sets: one a training set consisting of 24 days of data, and two test sets consisting of one day's data each.

Network Architectures

As discussed earlier, three basic neural network architectures were examined in this study: feedforward networks with an input window Elman recurrent networks, and Jordan recurrent networks. All three types of networks had one hidden layer. The network features used in this study are given below.

Model Set I

Case A: The networks had two inputs. The input consisted of the scheduled arrival time $T(k)$ and an input window representing schedule deviation SD at the timepoint $k - 1$ immediately preceding the current timepoint k location. All the networks for this case had five hidden units and one output unit.

Case B: The networks, in addition to the inputs discussed in Case A, had 12 units representing the current timepoint location k and the previous timepoint location $k - 1$. The networks had a total of 14 input units. All the networks had 20 hidden units and 1 output unit.

Model Set II

Case A: Input Units: 3, Hidden Units: 6

Case B: Input Units: 21. Eighteen units correspond to timepoint location, 1 unit to scheduled arrival time $T(k)$, and 2 units to input windows for schedule deviations.

Hidden Units: 21.

ANN architectures are denoted as $I \times H \times O$, where I, H, and O represent number of input, hidden, and output units, respectively.

Learning Algorithm and Update Functions

Since both the input (time and location, etc.) and output (schedule deviation) variables were known quantities, the schedule behavior modeling using ANNs constituted a supervised learning problem; hence supervised learning algorithms such as Quickprop were useful. QuickProp, which was developed by Fahlman (*9*), is a faster and more efficient version of the standard backpropagation algorithm.

Weight Initialization

The weights were initialized depending on the type of network architecture selected. The weights for the connections were randomly chosen between -0.001 and $+0.001$ for a feedforward network.

Network Training

The networks were trained with the QuickProp learning algorithm until there was no substantial decrease in the mean square error (MSE) for every 1000 iterations. The TanH (hyperbolic tangent) activation function was used for the hidden units. Both the MSE and sum of square errors (SSE) were computed for each iteration of the training process. MSE was used as a stopping criterion during the training phase.

Network Testing and Performance Evaluation

The networks were tested on the two test data sets, and the MSE and SSE were computed. The network performance was evaluated using average percentage error (PE_{avg}) to check the accuracy of the trained ANN models on the test data sets. The percentage error PE_{avg} was calculated for each point in the test data set (having n patterns) using the following expression:

$$PE_{avg} = \sum_{i=1}^{n} \frac{(SDact_i - SDpred_i)}{SDact_i} \times 100 \quad (4)$$

where

$SDact$ is the actual schedule deviation;

$SDpred$ is the network predicted schedule deviation; and

PE_{avg} is used to justify the accuracy and validity of the ANN models.

DISCUSSION OF RESULTS

The performance results of various ANNs are summarized in Table 1. The results indicate that for Case A, the average percentage error PE_{avg} was 3.5 to 6.30 points lower for Model Set II than for Model

TABLE 1 Comparison of Predictive Performance of Various Neural Network Models

ANN NETWORKS	Mean Square Error, MSE	Average % Error (PE_{avg})
	TEST Data	TEST Data
MODEL SET I		
Case A : 2x5x1 NETS		
Feedforward Net	0.00279	28.19
Elman Net	0.00412	24.95
Jordan Net	0.00219	27.65
Case B : 14x20x1 NETS		
Feedforward Net	0.00421	26.55
Elman Net	0.00720	26.97
Jordan Net	0.00403	28.38
MODEL SET II		
Case A: 3x6x1 NETS		
Feedforward Net	0.00232	24.25
Elman Net	0.00416	21.27
Jordan Net	0.00203	21.38
Case B: 21x21x1 NETS		
Feedforward Net	0.00479	24.57
Elman Net	0.00663	25.60
Jordan Net	0.00476	25.00

Set I. This leads to the conclusion that increasing the input series from $n = 1$ to $n = 2$ results in lower error values and more accurate models. Thus, providing the networks with longer input time series (for the example route, $SD(k - 6)$, $SD(k - 5)$, $SD(k - 4), \ldots,$ $SD(k - 1)$) leads to improved results. More inputs will provide more information, and are thus likely to provide more accurate results. It is interesting to note that providing additional information on the spatial location (timepoints) did not improve the accuracy. In the case of Model Set I, the PE_{avg} for Case B was higher than for Case A, for both the Elman and Jordan nets. This can be attributed to the increase in number of inputs that resulted in an increase in the complexity of the network and causes a higher MSE for the same number of training iterations. The same error behavior was also observed for Model Set II. In addition, the training times for Case B models were significantly higher (nearly 1.5 to 2 times) than those of Case A models, especially for Model Set II. Therefore it was concluded that there is no distinct advantage in including the spatial location information for schedule behavior modeling. The overall accuracy of the models ranged from 71 to 78 percent. Since no previous work on schedule behavior modeling has been reported in the published literature, no comparative study of the results could be made. The lower accuracy in network performance can be attributed to the following reasons: inadequate training data set, nonoptimal training of networks or shorter input time series ($n = 1, n = 2$). The training data set consisted of only 24 days of AVL information. It is believed that a larger data set, consisting of at least 6 months of AVL data, will improve the accuracy of the various neural network models. This is because ANNs are data-driven models and using a larger data set would result in a much better generalization of the schedule deviations.

The actual versus network-predicted schedule deviations are illustrated in Figures 4a, 5a, 6a for Model Set I, Case A, and in Figures 4b, 5b, 6b for Model Set II, Case A. The figures show that the ANN models performed well in capturing the trend in the schedule deviations at different times of day. The three networks learned the decreasing (or larger values of schedule delays) trends very well for Model Set II. While the best results were obtained for the Elman and Jordan networks, there were no significant differences between the networks. Hence, no definitive conclusion can be made on the superiority of one architecture over the other. The partial recurrent net architectures incorporate knowledge about the past states internally, and therefore seem more suitable for our schedule behavior modeling problem.

The schedule behavior models can be used for predicting the schedule deviations at timepoints $(k + 1)$, $(k + 2)$, and so forth, if the schedule deviation at the current timepoint k is known from the

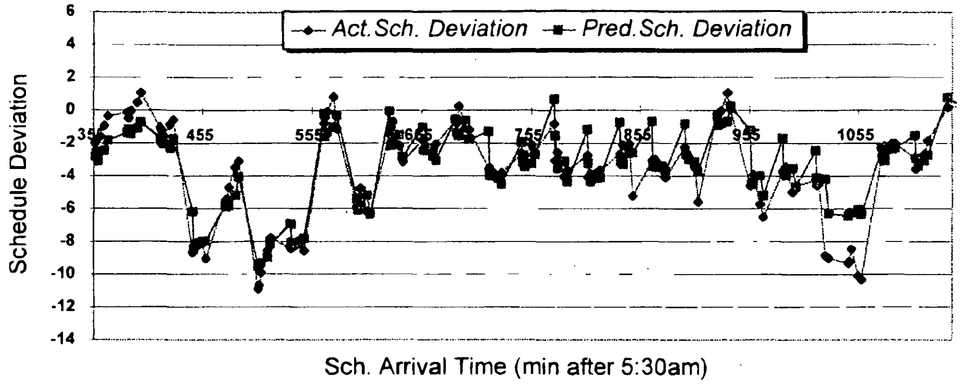

FIGURE 4a 2 × 5 × 1 feedforward net performance Model I: Case A.

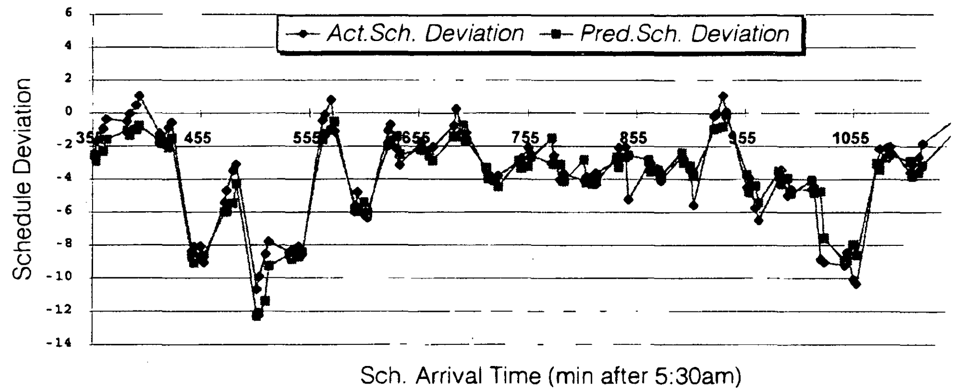

FIGURE 4b 3 × 6 × 1 feedforward net performance Model II: Case A.

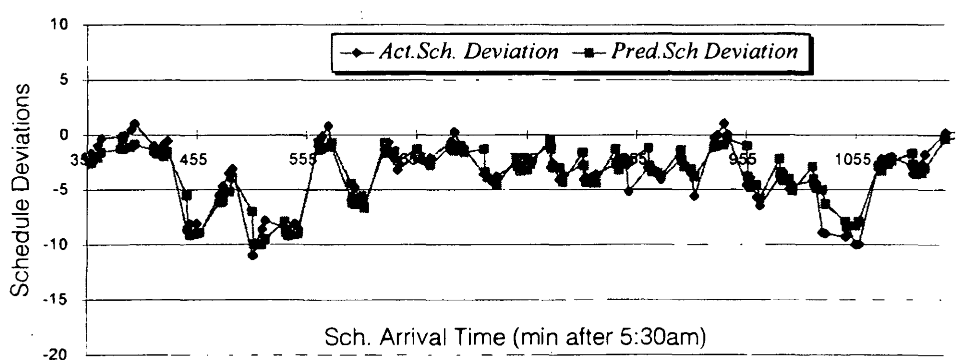

FIGURE 5a 2 × 5 × 1 Elman net performance Model I: Case A.

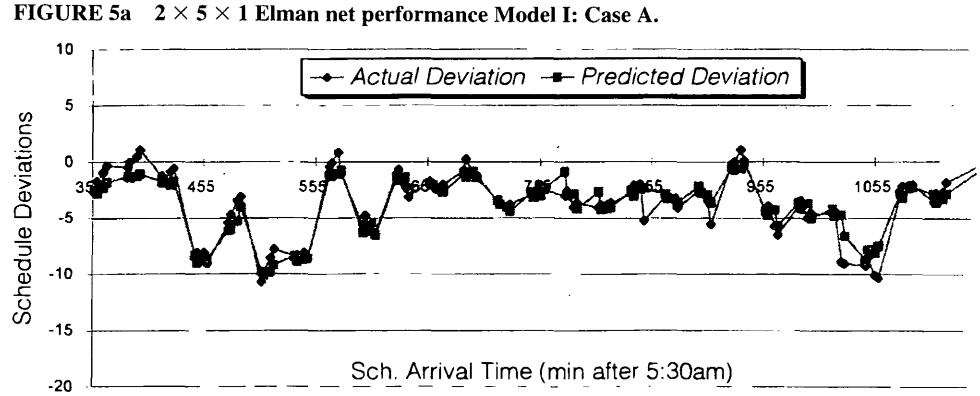

FIGURE 5b 3 × 6 × 1 Elman net performance Model II: Case A.

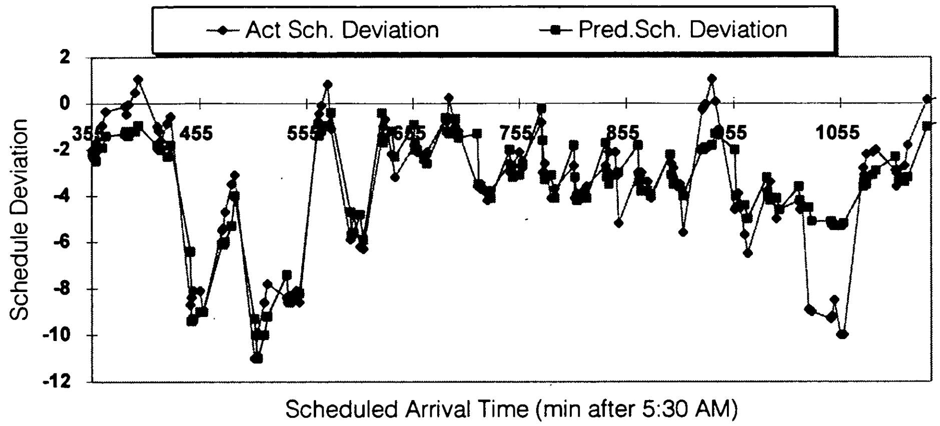

FIGURE 6a **2 × 5 × 1 Jordan net performance Model I: Case A.**

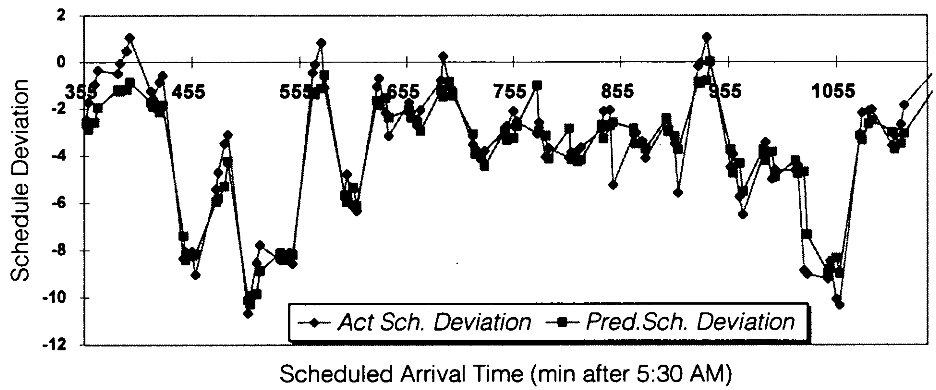

FIGURE 6b **3 × 6 × 1 Jordan net performance Model II: Case A.**

AVL system. This provides a time window for dispatchers and supervisors to evaluate the system performance online, implement any service-control strategy such as headway adjustments, and schedule adjustments appropriately. In addition, the models are useful for evaluating the effectiveness of any service-control strategy implemented. If the supervisors take appropriate control actions to offset any increase in schedule deviations estimated by the model at a downstream timepoint [$(k + 1)$, $(k + 2)$, . . . etc.], then the control strategies can be evaluated for effectiveness by comparing the actual schedule deviations at timepoints $(k + 1)$, $(k + 2)$, and so forth, with the model-predicted values to see whether there was a decrease in the schedule deviations. Currently, there are no procedures available for evaluating the effectiveness of service-control strategies in real time. Thus, the schedule behavior modeling approach proposed in this study can provide bus transit operators with an automated, online performance analysis and evaluation tool.

In summary, the ANN approach provides two distinct advantages over conventional statistical techniques for developing and implementing schedule-behavior models in real-world operations. First, the modeling process can incorporate the concept of spatiotemporal sequencing and short-term memory. Second, the models can first be developed off-line using historical data, and then used with current and new data for on-line updating of the models. This enables transit operators to deal with large amounts of data and a dynamic database in real time and thus can be useful in developing automated decision-support systems to assist dispatchers and supervisors with real-time service-control problems. Initial efforts are focused on investigating the development of schedule behavior models using ANN techniques.

CONCLUSIONS

The results from this case study indicate the suitability of the schedule behavior modeling methodology using ANNs. ANNs have the ability to incorporate short-term memory data about schedule deviations at consecutive timepoints on a route. While the results are encouraging, no definitive conclusions can be made regarding their performance unless a comparison is made between these results and other applicable techniques such as statistical methods. The methodology discussed herein for schedule behavior modeling can be used when applying other modeling techniques including statistical methods, among others. Ongoing research is aimed at investigating the modeling and prediction of schedule behavior of buses using conventional statistical techniques such as the Box-Jenkins model. Also under development are ANN models using longer input time series: $SD(k - 6)$, $SD(k - 5)$, . . . , $SD(k - 1)$. The development of schedule behavior models using the schedule deviation of buses on different routes arriving at a timed-transfer location is also being studied. Modeling the schedule behavior of buses on different routes arriving at a timed-transfer location will be useful for more efficient control of the arrival times of buses on various routes at the transfer location, and thus will minimize the number of missed transfers. In addition, the models can be useful for designing an optimal time window at timed-transfer locations.

ACKNOWLEDGMENTS

The authors wish to thank the Tidewater Regional Transit, Mid-Atlantic Transportation Centers Program, and the Virginia Department of Transportation for their support. The authors also wish to thank the Institute for Parallel and Distributed High Performance Systems, University of Stuttgart, Germany, for making available the Stuttgart Neural Network Simulator for our modeling experiments.

REFERENCES

1. Lapedes, A., and R. Farber. *Nonlinear Signal Processing Using Neural Networks: Prediction and System Modeling.* LA-UR-87-2662. Los Alamos National Lab Technical Report, Los Alamos, N.M., 1987.
2. Vemuri, V. R. and R. D. Rogers, eds. *Artificial Neural Networks: Forecasting Time Series.* IEEE Computer Society Press, Los Alamitos, Calif., 1994.
3. Su, H.-T., J. T. McAvoy, and P. Werbos. Long-Term Predictions of Chemical Processes Using Recurrent Neural Networks: A Parallel Training Approach. *Industrial Engineer and Chemical Research,* Vol. 31, 1992, pp. 1338–1352.
4. Mehra, P., and B. W. Woh. *Artificial Neural Networks: Concepts and Theory.* IEEE Computer Society Press, Los Alamitos, Calif., 1994.
5. Hertz, J., A. Krogh, and R. G. Palmer. *Introduction to the Theory of Neural Computation.* Addison-Wesley Publishing Company, Redding, Mass., 1991.
6. Weigend, A. S., B. A. Huberman, and D. E. Rumelhart. Predicting the Future: A Connectionist Approach. *International Journal of Neural Systems,* Vol. 1, No. 3, 1990, pp. 193–209.
7. Jordan, M. I. Attractor Dynamics and Parallelism in a Connectionist Sequential Machine. *Proceedings of the Eighth Annual Conference of the Cognitive Science Society,* Hillsdale, N.J., 1986, pp. 531–546.
8. Elman, J. L. Finding Structure in Time. *Cognitive Science,* Vol. 14, 1990, pp. 179–211.
9. Fahlman, S. E. Faster-Learning Variations on Backpropagation: An Empirical Study. In *1988 Connectionist Models Summer School* (T. J. Sejnowski, G. E. Hinton and D. S. Touretzky, eds.), Morgan Kaufmann, San Mateo, Calif., 1988.
10. Burke, L. Assessing a Neural Network: Validation Procedures, *PC AI,* March/April 1993, pp. 20–24.
11. Sharda, R., and R. B. Patil. Neural Networks as Forecasting Experts: An Empirical Test. *Proceedings of the IJCNN Meeting,* Washington, D.C., 1990, pp. 491–494.
12. Tang, Z., C. de Almeida, and P. A. Fishwick. Time Series Forecasting Using Neural Networks vs. Box-Jenkins Methodology. *Simulation,* Vol. 57, No. 5, Nov. 1991, pp. 303–310.
13. Weigend, S. A., and A. N. Gershenfeld, eds. Time Series Prediction: Forecasting the Future and Understanding the Past. *Proceedings of the NATO Advanced Research Workshop on Comparative Time Series Analysis,* Santa Fe, N.M., May 14–17, 1992.
14. Ulbricht, C. *Multi-Recurrent Networks for Traffic Forecasting.* Austrian Research Institute for Artificial Intelligence, Vienna, Austria, 1994.

Publication of this paper sponsored by Committee on Artificial Intelligence.