Investigation of Bus Transit Schedule Behavior Modeling Using Advanced Techniques

Final Report
By

Ravi Kalaputapu
Graduate Student

and

Michael J. Demetsky
Professor of Civil Engineering

Report No. UVA/29242/CE98/102

School of Engineering and Applied Science
University of Virginia
Charlottesville, VA 22903

Prepared for
Mid-Atlantic Universities Transportation Center

May 1998

Disclaimer

Standard Title Page - Report on Federally Funded Project

1. Report No. UVA/29242/CE98/102	2. C PB99-134421	3. Recipient's Catalog No.
4. Title and Subtitle Investigation of Bus Transit Schedule Behavior Modeling Using Advanced Techniques		5. Report Date May 15, 1998 6. Performing Organization Code 111-9805
7. Author(s) Ravi Kalaputapu and Michael J. Demetsky		8. Performing Organization Report No. UVA/29242/CE98/102
9. Performing Organization and Address Department of Civil Engineering University of Virginia Mid-Atlantic Universities Transportation Center		10. Work Unit No. (TRAIS) 11. Contract or Grant No. USDOT-TPSU-UV-0003-1113
12. Sponsoring Agencies' Name and Address US Department of Transportation 400 Seventh Street, S.W. Washington, D.C. 20590		13. Type of Report and Period Covered 14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

The advent of the Advanced Public Transportation Systems program by the Federal Transit Administration has encouraged bus transit operators to implement automatic vehicle location (AVL) systems for real-time monitoring. While the primary focus has been on the implementation of technologies, such as AVL systems, it is necessary and, perhaps, important to develop advanced performance analysis and evaluation procedures that can assist in the schedule planning and real-time service control tasks taking into advantage the real-time monitoring data. One potentially useful and effective approach for assisting in service control tasks, is the *schedule behavior modeling* concept. In this research effort, this concept is introduced to model the schedule behavior of buses on a route using schedule deviation information. The schedule behavior modeling approach presented in this study represents an innovative concept for modeling the performance of bus transit operations.

This research focussed on investigating the application of artificial neural networks (ANN) and the Box-Jenkins technique for developing and testing schedule behavior models using data obtained for a test route from Tidewater Regional Transit's AVL system. The three ANN architectures investigated were: Feedforward Network, Elman Network and Jordan Network. In addition, five different model structures were investigated. The time-series methodology was adopted for developing the schedule behavior models. The modeling experiments provided no conclusive results. However, the Jordan network model provided encouraging results and performed well. Finally, the role of a schedule behavior model within the framework of an intelligent transit management system is defined and the potential utility of the schedule behavior model is discussed using an example application.

17 Key Words bus transit, schedule behavior modeling, transit performance analysis, artificial neural networks, Box-Jenkins technique		18. Distribution Statement No restrictions. This document is available to the public through NTIS, Springfield, VA 22161.	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages	22. Price

ACKNOWLEDGEMENTS

This research effort has been funded by the Mid-Altantic University Transportation Center, Division III and the Department of Civil Engineering, University of Virginia. The author's gratefully acknowledge the sponsor' suport. The authors are also grateful to Jeff Becker and other members of his staff at the Tidewater Regional Transit for providing the necessary data for conducting this research.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background

Transit operations have faced uncertainty both in terms of ridership and funding subsidies in recent years. creating a tremendous pressure on transit operators to improve the efficiency and cost-effectiveness of their systems. In a 1985 survey of 146 transit operators in the United States, conducted for the Transportation Research Board, 65 percent of all operators reported on-time performance as essential, critical, highly important, or very important for operational planning and management. Another 35 percent considered it as important or moderately important [NCTRP91]. Transit managers perceive effective and efficient bus service control as the most critical task in order to ensure reliable service to the passengers [NCTRP91]. Passengers want buses that run on time. Passengers are often faced with uncertain arrival times and the lack of real-time information on actual arrival times as compared to printed schedules. Such situations increase uncertainty, thus eroding their confidence on the system.

In the past, one of the most important limitations that bus transit operators have faced is the non-availability of comprehensive information on real-time operations. The evaluation of service performance was often time consuming and did not include data on the entire network. Hence this lack of continuous monitoring of daily operations affected operational effectiveness and success of management strategies to improve service reliability and efficiency. Data for the evaluation of service performance was often manually performed using supervisors stationed along the routes in the network. In addition, processing of the data collected took time and effort and was performed off-line. As a result transit performance evaluation was not useful to the management especially with regards to making quick and timely decisions regarding implementation of service restoration strategies in order to minimize the effects of schedule disruptions. Since the evaluation of transit operational performance is heavily dependent on the availability of timely, accurate, continuous and comprehensive information on operations, there is an immense potential to use advanced technologies to collect the required data. The advent of the Advanced Public Transportation Systems (APTS) program as part of Intelligent Transportation Systems (ITS) has encouraged transit operators to implement advanced technologies for accurate and real-time monitoring of their services. These systems are designed to provide the necessary information for performance evaluation and also timely implementation of appropriate response strategies in cases of service degradation or disruption. The APTS program was established by the Federal Transit Adminstration (FTA) to use advanced telecommunications, computer and other electronic technologies for the purposes of improving the cost-effectiveness and efficiency of public transportation services.

Advanced technology applications for bus transit operations involve two important elements, both of which are essential for achieving higher efficiency and reliability. These two elements, (i) technological hardware for collection of operational data and (ii) advanced management software support systems for intelligent processing of operational data, are essential for monitoring, operational management of bus transit operations. The technological hardware is the critical component for monitoring of everyday operations. The most useful and applicable advanced hardware technology is the Automatic Vehicle Location (AVL) System, which provides the ability to monitor operations in real time and with a high level of accuracy and reliability. The AVL system can provide transit management with information that can result in a reliable bus service in which buses run on time along a route and deviances in schedule adherence are minimized. However, this is possible only if efforts at hardware implementation of an AVL system are complemented by development of an intelligent decision support system (IDSS) for real time service control and operational management. In order to design and implement such automated decision support systems it is necessary to first develop advanced techniques for real-time performance analysis and evaluation. The real time operational performance modeling framework is essential in order to design and implement effective real time service control strategies. A serious limitation of current practices is that there have been few efforts at developing techniques for real time performance modeling and service control. The development of a transit management decision support system will help transit operators realize the stated benefits from implementation of a real-time monitoring system. Within the framework of the national ITS system architecture, the IDSS can be construed as an important component of the Public Transportation Operations user services group. The intelligent transit management system architecture discussed in Chapter 7 can serve as a basic foundation for the detailed development of the transit compoenent of the national ITS system architecture.

1.2 Problem Statement

Bus transit service control, that includes schedule control, headway control and dispatching control, is a critical concern of any bus transit management. AVL systems provide transit operators with the necessary technology to monitor the bus operations in real-time. This opens up a new window of opportunity for intelligent decision-making using advanced techniques for the purpose of achieving more efficient service control. However, there are two important observations that are relevant and worth mentioning. First, the large amounts of data from the AVL system has lead to an information overload on the dispatcher. Secondly, the large amount of information obtained from the monitoring system is yet to be effectively utilized for real time performance analysis and service control. These two observations make it necessary to address the problem of information explosion and underutilization of the wealth of information that is obtained from an AVL system. The large amounts of data provide an opportunity to investigate the development of real time performance analysis and evaluation techniques using advanced modeling techniques. The problem of information overload on dispatchers and supervisors can be addressed by undertaking the development of an automated system such as an intelligent decision support system (IDSS) for real-time operational management of bus transit operations. Under the umbrella of an IDSS, many subsystems have to be developed for various tasks relating to operational management. For example, a service control decision support subsystem can be developed to provide the dispatcher with the best service control strategy for any service degradation situation. Another example is a fleet management system that provides decision support for effective management of vehicles and operators. All these components are inter-linked and require synchronization, as any changes in real time operation requires time-critical decision support from both the service control system and the fleet management system. Any addition of buses to restore services on a given route requires advanced planning for vehicles and bus operators availability. The problem of developing a detailed system design for the IDSS component is beyond the scope of this research effort. Before the bus transit performance modeling problem is defined and the objectives of this research effort are enumerated it is important to define the term schedule behavior. The following section presents a simple definition of schedule behavior.

1.2.1 What is Schedule Behavior?

In this research the schedule deviation of a bus at various timepoints along the route is being used as a primary variable for performance modeling. The schedule deviation is defined as the difference between actual arrival times and scheduled arrival times at a timepoint. The term *schedule behavior* has been used to indicate bus transit operational performance which is measured in terms of the schedule deviation of buses. The schedule behavior modeling problem can be defined as a spatio-temporal problem since the schedule behavior of a bus on a particular route is dependent on where the bus is located along the route as well as on the time of day. The time of day is represented by the scheduled arrival time stamp while the location is represented by the timepoint information.

1.2.2 Why Model ?

The real-time monitoring of the bus transit system provides an opportunity to investigate the application of modeling techniques for the development of an advanced online system performance analysis and evaluation tool. Such techniques have been successfully used in similar dynamic systems, such as manufacturing, chemical process, nuclear power plants etc., that are continuously monitored and require a certain degree of control of operations [Foster92, Guo92, Chitra92]. In many such systems, *performance models* of the system behavior are developed in order to achieve more efficient and cost-effective control of the system operation. In observation-dependent systems such as bus transit system performance characteristics are inferred through measurement of the unfamiliar system from the analysis of a measured time record of its behavior in order to predict the future system performance characteristics. Hence the performance models are also referred to as prediction models when used to predict the future behavior of the system. Models of schedule behavior can be used in the development and implementation of bus transit service control strategies that attempt to minimize schedule degradation so that a certain level of service reliability is maintained. The performance models can be used to develop effective schedule adjustment and headway adjustment plans so that the bus transit resources, namely buses and drivers, are efficiently managed.

The primary focus of this research effort is on investigating the feasibility of developing schedule behavior models for buses on a transit route. The central idea of this research effort is that a schedule behavior model can provide a knowledgeable understanding of past system behavior of buses on a route. Such a model has several potential uses. It can be used to predict 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. The schedule behavior models can also be potentially utilized for updating/modifying schedule plans. The models can be used to speed up and automate the process of performance analysis and evaluation of implemented service control strategies. Currently there are no available automated procedures 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 for improving service reliability.

Traditionally, in a number of control problems, forecasting or prediction techniques are often used to model dynamic system behavior so to as assist in the development of effective and efficient control strategies. The bus service control problem holds potential for a similar application of a prediction model to assist in the design and development of an intelligent decision support system for selection of appropriate service control strategies. With the availability of a real time monitoring system, it is possible for bus transit management to explore the development of a schedule behavior prediction model and ascertain its potential utility in real-time service control. The time critical operation of the service control system requires the aid of a forecasting tool that essentially predicts schedule behavior of buses on any route at any given time of the day.

The research problem can be defined as predicting the schedule deviation of a bus at timepoint k, given the schedule deviations at its previous timepoints and also the time of day information. The time of day is represented by the scheduled arrival time of a bus at timepoint k. Mathematically, this is represented as :

Given : SD(k-1), SD(k-2) ....SD(k - $\boldsymbol{n}$), where SD denotes schedule deviation at timepoints k -1 , k-2, ...*etc.* $\boldsymbol{n}$ denotes the length of the input series or the number of upstream timepoints whose schedule deviation information will be included to model.

t(k) : Scheduled Arrival Time at timepoint k.

Predict: SD(k) : Schedule Deviation at timepoint k.

The rationale for the above definition of the schedule behavior modeling problem is presented in Chapter 4.

1.3 Objectives

The objectives of the proposed research are :

- To investigate the feasibility of modeling the schedule behavior of buses on a fixed route bus transit system.
- To formulate a structure for a schedule behavior model of a fixed route urban bus service.
- To demonstrate the application of AVL data and artificial neural networks for modeling the schedule behavior of buses on a transit route.

- To demonstrate the application of AVL data and time-series models based on the Box-Jenkins technique for modeling the schedule behavior of buses on a transit route.

1.4 Purpose and Scope

The purpose of this study is to investigate the feasibility of developing schedule behavior models which can be potentially useful as tools for the development and implementation of service control strategies. The utility of new and promising techniques such as artificial neural networks for modeling and prediction of the key operational measure of performance, namely, schedule deviation is investigated. The focus is on using historical schedule monitoring information for the purpose of developing schedule behavior models on a specific route in the network and to use the models to predict the schedule deviations of a bus at a downstream location based on the schedule deviation information at the current timepoint or location at different times of the day. Modeling at the route level provides an opportunity to analyze and evaluate service control strategies since service problems surface at this level of operations and adjustments are required at this micro level. The forecasting of the key schedule behavior indicator, schedule deviation, can provide transit management with operational performance characteristics that can be utilized for design, development and real-time implementation of quick-response service restoration strategies when problems such as delays due to recurring and non-recurring events occur on the transit network. The study performs a comparative analysis of the schedule behavior models developed using conventional time-series techniques and new and promising artificial neural network methods.

The scope of this research effort is limited to the development of relevant models using schedule monitoring data from a single route of Tidewater Regional Transit. The reason for choosing a single route was because the objective of this research was to investigate the feasibility of schedule behavior modeling and hence it is appropriate and reasonable to look at the problem at the simplest level which in the transit case is a route. In addition, while conducting basic research into feasability of a particular modeling a pproach it is pertinent and necessary to address the problem at the micro level. In addition, the study focusses on developing schedule behavior models that predict only one timepoint ahead.

1.5 Organization of the Report

Chapter 1 states the problem and motivation for this research, defines the objectives, purpose and scope of this research.

Chapter 2 presents an overview of current state of the art and practice for bus transit operational performance analysis and highlights the drawbacks of the current procedures in order to provide the neccessary justification for this research effort.

In Chapter 3 the basic concepts relating to the different modeling approaches, namely, artificial neural networks and statistical techniques are presented. This chapter presents a detailed background and overview of the three applicable artificial neural network techniques - Feedforward, Elman and Jordan networks. The chapter also presents an overview of ANNs applications to engineering problems highlighting the salient features that are relevent to this study. The limitations of artificial neural network approach are discussed. An overview of the Quickprop learning algorithm is also presented. In the second section of this chapter, the theoretical concepts for statistical modeling are discussed. A detailed review of the Box-Jenkins models (ARIMA) and the exponential smoothing techniques are discussed. The various features of the ARIMA model and its applicability to the bus transit performance analysis problem are highlighted.

Chapter 4 describes the general framework for schedule behavior modeling. It presents a detailed description of the modeling methodology which constitutes the most important contribution of this research effort . The chapter also dicusses the issue of explicit models versus implicit models in order to elucidate the usefulness of the artifical neural network approach for the bus transit performance analysis problem. This chapter also discusses the modeling process and presents general guidelines for modeling using artifical neural networks and the ARIMA modeling technique. The various model structures that are being investigated for modeling the schedule behavior of buses on an urban transit route are defined.

In Chapter 5 a detailed description of the case study is presented. The data collection procedure is discussed and the problems associated with data collection are highlighted. The contribution of this chapter is the detailed description of data preprocessing and manipulation technique that is required for developing the schedule behavior models. A sample illustration of the actual data coming from the AVL system is described. Also, the characteristics of the TRT's transit system is described.

Chapter 6 describes the results of the modeling experiments conducted in order to develop the schedule behavior models using the preprocessed data from our case study. A detailed analysis of the results is presented including a comparative analysis of the performance of the various models. The characteristics of the various models are highlighted.

In Chapter 7 a potential architecture for application of schedule behavior modeling concept within an ITS environment is proposed. The usefulness of the schedule behavior models within the context of an intelligent transit management system is highlighted. The various implementation related issues and potential problems and limitations are described. Finally the last chapter presents general conclusions, summary of results, limitations of this research effort and suggestions for further work.

CHAPTER 2

BUS TRANSIT PERFORMANCE ANALYSIS TECHNIQUES

2.1 Bus Transit Operational Management

Operational Management of bus transit involves the basic supervision of everyday system operations with focus on service control, fleet management, service supervision, dispatching control, and incident detection and response. These essential tasks are critical for transit management to develop suitable mitigating strategies to different problems associated with operations of buses. The problem identification and implementation of appropriate problem-solving strategies have to be performed under several constraints notable among which is time. Effective real-time control of bus transit operations is essential to provide the riders with an efficient and reliable service. Operational management and planning thus require effective and accurate monitoring of operations in real time. In addition, there is a need to develop effective performance evaluation procedures. One applicable technique is to develop models of system behavior using the real-time monitoring information that is available from the AVL system.

2.2 What is On-time Performance ?

On-time performance has been defined as *"a bus arriving, passing, or leaving a predetermined bus stop along its route within a time period that is no more than* ***x*** *minutes earlier and no more than* ***y*** *minutes later than a published schedule time"* [Guenthner]. On-time performance has been used as a measure of effectiveness of bus transit operations. A number of studies have delved into this research topic. The on-time performance is a critical performance indicator especially for timed-transfer bus transit systems. Typical values for x and y have ranged from -2 to +2 minutes [Guenthner].

2.3 State of the Art : Bus transit Performance Analysis Techniques

Previous research on on-time performance (OTP) analysis has focused on three approaches. The first approach concentrated on ascertaining the shape of the frequency distribution of schedule deviations [Talley87, Guenthner]. This approach, which essentially is a static modeling approach, provides a clear picture of on-time performance but does not systematically measure the effects of underlying causes (causal model, or in other words an *expost* approach). The second approach focused on obtaining key indicators of service reliability, such as running time and headway variation [Abkowitz83, Abkowitz84]. This approach did not directly address the on-time performance analysis problem as it only considered the various factors affecting service reliability in detail. The third approach is based on empirical analysis of on-time performance and essentially attempted to bridge the two previous approaches [Strathman]. Strathman et al. employed a multinomial logit model for modeling on-time performance. The logit model related the on-time performance to route, schedule, driver and operating characteristics. The model results showed that the probability of on-time failures increases during AM-PM peak periods and also as buses progress further along their routes. Strathman's observation on the influence of the location of the buses along a route on on-time performance is used as a rationale for formulation of the schedule behavior modeling approach discussed in Chapter 4.

One important observation that was noted from the literature reviewed is that previous efforts at developing on-time performance analysis techniques have mainly concentrated on analytical and simulation techniques. The lack of empirical research on on-time performance analysis problem can be attributed to logistical problems in collecting real-time data. Seneviratne [Senevi90] in his study developed a simulation model using Monte Carlo technique for analyzing the impact of different operating strategies (eg. additional time points, bus priority, and demand profile) on headway variations and degree of adherence to schedules (on-time performance).

Current approaches to modeling the schedule performance have some important drawbacks. First, they do not provide a usable framework for real time performance monitoring and control due to the fact that they employ *ex post* approach. Another drawback is that the level at which the models are applicable. Strathman's study provides results at the overall transit route system level. It is intuitive that the service problems relating to schedule adherence are often at the route level and which requires mitigating control strategies.

Another important drawback is that most models developed for schedule performance have not been assessed for their robustness.

Current methods for on-time performance analysis and evaluation have been based on static approaches relying heavily on schedule deviation data collected regularly but not in a continuous fashion. The models were essentially based on empirical analysis methods. However a bus transit system can be construed as a non-linear dynamic system. Thus, dynamic modeling techniques such as ANNs can be investigated for developing schedule behavior models. This research effort attempts to further the state of the art in bus transit performance analysis by proposing and developing alternative approaches, using advanced modeling techniques based on time-series methodology.

2.4 Significance

A detailed discussion of the state of the practice in supervision strategies of bus routes is presented in the NCTRP Synthesis of Practice Report 15 [NCTRP91]. This report, in its recommendations for new research directions, pointed the need to determine new techniques for modifying service control and supervisory practices. However research on these topics has been limited or none has been reported so far. Therefore the problem being attempted by this research effort is of great significance. The proposed research will bring new relevance to operational management practices and will try to address one of the key future research directions noted in the NCTRP report.

The literature reviewed on schedule planning and timetable management of bus operations revealed that transit operators have to compromise between several conflicting objectives that reflected demand, operating costs, quality of service, social needs, robustness of the schedule to disruptions etc. The ideal situation is to see that headway variations are smooth and also that there should be short transient phases before and after peak hours [Etschmair]. In addition the bus transit guidelines desire that the waiting time for riders at transfer points be minimum. Also the guidelines define that the schedules should satisfy a number of constraints : headway allowances, maximum schedule degradation, stabling and unstabling procedures etc. Etschmair in his study provided some detailed guidelines for control of bus transit operations. The control variables suggested by many researchers in this report are : headways, transfer waiting times and time and direction of stabling and unstabling events. The overall desire is to perform schedule and headway adjustment in order to achieve effective service control. This can be possible only if the changes in schedules are communicated to the riders through an on-line real-time passenger information system. The general desire is to go in for automation of the man-machine interface of schedule adjustment in order to obtain greater efficiency and also reducing the information load on the dispatcher [Watanabe]. In order to achieve greater automation of information processing and service control related decision support, an important step will be to design and develop a performance model of the real system that would aid management in the design and development of service control strategies based on schedule and headway adjustment.

2.5 Summary

Bus transit operational performance analysis and control is a critical problem that requires more efficient and effective approaches in order to realize the benefits from implementation of advanced real time monitoring systems. Current techniques for performance analysis are static in nature and may not be useful for assisting in efficient real-time control of operations. They are also not suitable for automating the decision support process for real-time control of operations. If effective real time bus control strategies have to be designed so that they can be implemented automatically in real-time, then the applicability and usefulness of alternative system modeling techniques have to be investigated. This study aims to investigate the applicability and usefulness of conventional statistical techniques and artifical neural network based modeling approaches for the bus transit performance modeling problem.

CHAPTER 3

MODELING TECHNIQUES

3.1 Artificial Neural Networks

Artificial neural networks are learning systems that have 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 some conventional methods. They have been shown to have a wide range of applicability in areas such as continuous speech recognition and synthesis, pattern recognition, classification of noisy data, nonlinear feature detection, market forecasting, nonlinear and adaptive control and process modeling. Also neural network learning systems contribute significantly to analysis, prediction, and optimization of chemical manufacturing units and power plants performance [Guo92, Rehbein92, Reins91, Chitra92]. Most of the applied research on this subject has concentrated on the use of ANNs to solve important system performance related problems or process control problems. An overview of various applications of ANNs is presented in a later section. One major problem in the study of ANNs is that the literature in this area is scattered over a vast number of publications in different disciplines. The broad spectrum of ANN research and its diverse sources, make it difficult for researchers to keep pace with current developments. Mehra and Wah [Mehra92] through their efforts have produced a comprehensive description of concepts and theory of ANNs that would alleviate the aforementioned difficulty.

One of the fundamental drawbacks of applying new and promising techniques such as artificial neural networks is that most of the research papers on theory recommend the application of a particular theoretical concept to tackle certain kinds of learning problems. The majority of articles presenting new theoretical concepts such as a new algorithm do not answer the following key questions [Prechelt93]:

> "*For what kind of problems does the new algorithm or architecture work well or not well* ?" and
>
> "*Under what conditions should we prefer the new algorithm over previously known ones* ?".

The answers to these fundamental questions is essential for successful development of applications to real world problems. However for this research effort we are handicapped by the absence of information on the above questions in the theoretical literature on artificial neural networks.

3.1.1 Advantages of Artificial Neural Networks

Many researchers have discussed the distinct advantages of artificial neural networks [Rumel90, Foster92, Weiss90, Guo92, Mehra92, Lippmann]. Artificial neural network have a highly distributed parallel structure and when combined with powerful hardware digital technology can make model simulations economical and with relative ease. ANNs mimic human learning processes and as such hold great potential as adaptive learning systems.

ANNs can handle complex and nonlinear models that are common to dynamic systems such as bus transit operations. ANNs offer the promise of being able to extract information from automatic vehicle location data in an efficient manner. 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 changing characteristics of the operating environment of buses. The potential advantage of ANN learning method is that, compared to 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 [Weigend92]. This ability to approximate unknown functions through the presentation of past states of a system makes ANNs a useful tool for modeling in engineering applications, such as for bus transit schedule behavior modeling. The modeling approach using neural networks essentially helps to perform two important tasks. First, it learns the system performance using past and current AVL data. Secondly, the ANN models can be used to predict the behavior of the buses. 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.

3.1.2 Drawbacks of Artificial Neural Networks

The literature has pointed out the following drawbacks of Artificial neural networks : (I) lack of standard guidelines for selection of ANN structures and training methods; (ii) with so many processing units (parameters) available there is a tendency to overfit the model. There is no clear and efficient way to avoid overfitting; (iii) no guidelines for handling multiple solutions and problem of local minimum; (iv) often take long time to train because of the necessity of using a large number of weights and hence may be slower than other mathematical techniques such as regression analysis etc.; (v) it is difficult to incorporate deep knowledge compared to machine learning techniques; (vi) the processing units and weights that in a way represent parameters and their values cannot be interpreted physically as in the case of statistical techniques [Burke93, Mehra92]. In general, an ANN does not give confidence intervals for its outputs. Without prior experience with the problem in hand, and as is the case with the bus schedule behavior modeling problem, the network topology is determined by trial and error. A too small network will make learning difficult and a too large network will generalize poorly.

3.1.3 Applications: An Overview

3.1.3.1 General Applications

Artificial neural networks provide an effective approach for a broad spectrum of applications. Essentially ANNs represent a paradigm for intelligent processing of information for some specific objective such as classification, pattern recognition or decision-making. A number of useful applications have been investigated but so far only a few or negligible number of successful applications have been developed and demonstrated for real world problems.

Artificial neural networks have been proposed recently for nonlinear prediction and system modeling [Lapedes87, Hernandez, Bhat90]. ANNs have been shown to have promise as models for a wide variety of largely deterministic problems in relatively few variables by Bhat and Avoy [Bhat90],and Lapedes and Farber [Lapedes87]. A number of well researched articles in application of neural networks in chemical engineering have been published in [CChEng92]. ANNs have also shown to be successful non-linear signal predictors in speech recognition by Iso and Watanabe [Iso90], Tebelskis and Waibel [Tebel90] and Levin [Levin90]. Neural networks have proven to be a promising alternative to traditional techniques for nonlinear temporal prediction tasks [Weigand 92, Lapedes87]. Such successful applications makes it imperative that we investigate the feasibility and the potential usefulness of applying artificial neural networks to our schedule behavior modeling problem.

Lapedes and Farber [Lapedes87] reported that simple neural networks can outperform conventional methods. Sharda and Patil [Sharda] 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. [Tang91] 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 Box-Jenkins model. A review of relevant literature indicated that each of the methods performed better than the other about half of the time.

Kumar et al. [Kumar91] used Elman's recurrent network to two types of speech recognition problems. They found in their study that Elman network is capable of learning the mapping by developing the appropriate dynamic behavior using context units [Kumar91]. Lambert et al [Lambert] in their investigation of chemical plant predictive modeling using recurrent networks concluded that recurrent networks are suitable for dynamic modeling problems. One of the shortcomings of previous research efforts at investigating the application of recurrent networks is that they usually take up small toy examples to demonstrate the applicability.

Some researchers have reported that ANNs can be potentially used for economic forecasting problems [Trippi93, Kamijo90]. A number of studies [Kimoto90, Sharda, Hoptroff, Refnes93,Varfis] on economic forecasting problems provided a comparative analysis between ANNs and conventional time series techniques such as ARIMA. ANNs have also been applied to electrical load forecasting problems [Park91, Bacha92, Connor93]. These studies made some general conclusions on the applicability of the various techniques to the specific problem but fell short of suggesting that one technique is superior to the other. Also, it has been stated by researchers that it is perhaps not necessary to make a comparative study in order to demonstrate that one method is superior to the other but it is important to investigate the conditions and design specifications under which any particular technique is suitable or not suitable to a specific problem under study. However, since there have been no prior attempts to develop schedule behavior models, the research approach adopted would also focus on conducting a comparative analysis between the various techniques selected for this study.

One major problem faced by researchers trying to investigate the application of ANNs for a specific real world problem is that past studies, if there are any, they do not adequately provide any insight and possible suggestions for network design and other modeling design considerations. Prechelt et al. through their investigation provided the primary reason for the caution and inability to use any past studies. The reason cited by Prechelt et al. is that most researchers conduct ANN modeling experiments with toy problems and hence any conclusions made on the applicability and usefulness of various learning algorithms and architectures cannot be used as a selection basis for conducting the experiments for another problem.

The second type of problem where there has been considerable interest and focus is control theory applications. The literature reviewed indicated that ANNs can be applied in two ways in the design of control systems. ANNs can be used to obtain a mathematical model of the real system to be controlled. They can also be used to design a controller once a model of the real system is available. Both of these tasks have been studied using ANNs [Beale92, LeCun88, Simpson, Kawato,Narendra].

In the case of a bus transit system, it is possible and necessary to develop models of the schedule behavior using the AVL information. However the bus transit system is a non-physical system where the system's behavior is affected by human factors such as driver characteristics, loading/unloading characteristics of passengers. and unpredictable factors such as traffic characteristics etc. Also, the control of bus transit operations involves human element (Drivers, Passengers etc) and hence ANN modeling for achieving control is not a suitable application.

3.1.3.2 Applications in Transportation

There have been some attempts to apply ANNs to new areas such as transportation in the past few years. Most applications relate to the use of ANNs for identification and classification for the purpose of aiding in decision making [Faghri92,Cheu91]. The specific areas where application of ANNs have been attempted are in traffic management, traffic signal control and pavement management.

Fagri and Hua [Faghri92] in their study list the potential applications of ANNs to various problems in the field of transportation, such as traffic management etc. However the study failed to list potential application of ANNs in the area of public transportation. Under the ITS program, new technologies have been proposed in different areas such as traffic management. However the application of advanced learning systems such as ANNs to public transportation management has not been proposed and attempted so far. Cheu et al. [Cheu91] investigated the use of a neural network model for freeway incident detection. Most of the applications in traffic engineering concentrated on using the standard backpropagation learning algorithm with feed-forward networks [Faghri92, Dough93,Smith95]. Kirby et al. [Kirby93] in their study on short-term traffic forecasting concluded that neural networks can be successfully trained to provide short-term traffic forecasts. When they compared the results from neural networks to that from the conventional AutoRegressive Integrated Moving Average (ARIMA) technique, they found that neural networks perform as well or better than the ARIMA models for short-term traffic forecasts involving 30 minute or less time bands. They also concluded that it is difficult and inappropriate to compare the two techniques with a single comparative measure such as a correlation coefficient. The same conclusion was made by a number of other researchers. However, the literature reviewed revealed the lack of a suitable scheme for performing a reliable and accurate comparative analysis between ANNs and ARIMA type of models.

Smith [Smith95] in his research effort on forecasting freeway traffic flow for ITS applications compared the performance of a feedforward network with other conventional techniques such as ARIMA and nearest neighbor and nonparametric regression. Smith's research was one of the first attempts to perform a comparative analysis of different modeling techniques and also investigate the feasibility of deploying the models in a field environment. Smith's research effort involved using real data sets. He concluded that the nonparametric regression model outperformed all others for multiple interval forecasting and also stated that the model was portable and easy to deploy in a field environment [Smith95].

Faghri et al. In their research study on modeling trip production using ANNs also involved the use of real data sets [Faghri96]. In their research they compared the performance of the ANNs with the performance of standard regression models. They concluded that the ANNs predictions were far more accurate than those based on regression analysis.

Gilmore et al. developed a Hopfield neural network model for adaptive traffic signal control using simulated traffic data [Gilmore95]. In their study they also proposed a simple feedforward network model for predicting urban traffic congestion. The drawback of this study was the use of simulated data for developing the models. In addition the study did not perform any comparative analysis of performance with other potentially applicable techniques.

Yun et al. investigated the application of a recurrent neural network to traffic volume forecasting [Yun96]. In their comparative analysis of a recurrent neural network with the traditional feedforward network and the Finite Impulse Response (FIR) model they observed that the recurrent neural network model outperformed the other models in forecasting very randomly moving data. However, the FIR model showed better forecasting accuracy than other models for the relatively regular periodic data. The authors concluded that the feedback mechanism of the previous error through the time learning technique in the time-delayed recurrent network naturally absorbs the dynamic change of the underlying non-linear movement.

Zhang et al. developed a macroscopic model of freeway traffic using an artificial neural network with two hidden layers. However, the study used simulated data to develop the models [Zhang97].

The application of artificial neural networks in transportation has evolved over time from simple toy applications using simulated data to more sophisticated applications such as those attempted by Smith, Faghri et. al and Yun et al. [Smith95, Faghri96, Yun96]. A review of the literature revealed that there has been no detailed implementation of ANNs for real world transportation related problems. However, there is a growing enthusiasm for the development of neural network applications to solve complex problems using real data. The advent of the ITS program has provided the necessary impetus to attempt new and exciting techniques to solve some critical problems relating to traffic management and transit operational management. As the state of the art with respect to ANNs is bound to improve over the coming years ANNs can be potentially good techniques for application in certain class of problems relating to traffic signal control, traffic flow forecasting etc.

3.1.4 Basic Structure of an Artificial Neural Network

Figure 3.1 illustrates the typical architecture of an artificial neural network. In general, the basic structure of any ANN model [Hecht90] consists of the following features:

- a network of highly distributed and interconnected "processing units" arranged in different layers,
- an interconnection scheme,i.e.,fully interconnected, partially/sparsely interconnected; uni-directional vs bi-directional, essentially feedforward vs recurrent etc.,
- activation functions (for relating output values of a processing unit to its inputs),
- a cost function that evaluates the network's output (e.g. squared error etc.),
- a unique learning law,
- data represented as training set and test set,
- training algorithm that changes the interconnection parameters (called weights) in order to minimize the cost function.

How a neural network performs is dependent on the critical task of dividing the data set into training and test sets. In addition there are some key issues that are critical for the successful implementation of a neural network model.

In the development of neural network models, the following fundamental questions

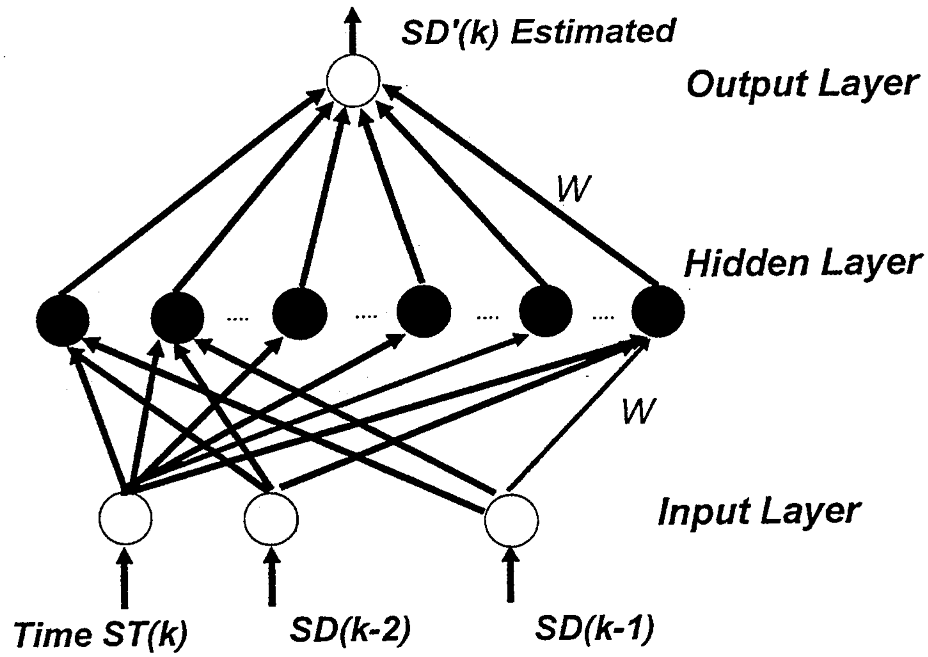


Fig 3.1: Typical Artificial Neural Network Architecture

need to be answered :

♦ *What network architecture should be used ?*

An artificial neural network typically consists of an input layer, one or more hidden layers, and an output layer. While defining the network topology the specific question of interest and importance is: What is the internal structure of the short-term memory networks? The answer to this question involves the specification of the number of hidden layers and number of units in each layer, the pattern of connectivity among units, and the activation functions to be used for the various types of units [Mozer92].

♦ *What is the learning paradigm to be used for the selected network architecture?*

♦ *How is the learning rule executed?*

In other words, what optimization technique is to be applied to help in training and its successful completion? The optimization technique helps define the convergence or successful termination of the training process. The learning rule essentially defines the relationship that governs the training process, i.e the relationship that defines how to adjust the connection weights. The selection of the connection weights for a neural network can be viewed as an optimization problem. The objective is to choose a set of connection weights that will minimize the network prediction error (the selected error function) over all the training examples. The most often used error functions are the sum of squared errors (SSE) or mean squared error (MSE) of the training data set.

With the neural network approach, the issues of architecture, network dynamics, training procedure, and representation are intricately related. They are viewed as different perspectives on the same underlying problems. Mozer [Mozer92] stated that a given choice of representation may demand a certain neural net architecture or a particular type of learning algorithm to compute the representation. He also stated that a given architecture or learning algorithm may constrain the class of representations that can be adapted for application with that particular architecture or learning algorithm. This is especially important for spatio-temporal sequencing problems such as the bus transit schedule behavior modeling and prediction. The following section describes three important architectures that will be applied for modeling the schedule behavior of buses.

3.1.5 Network Architectures

Artificial Neural Networks can be classified as Feedforward Networks (FFN) and Recurrent Networks (RNN). Recurrent networks can be further classified as fully recurrent networks and partial recurrent networks. The main distinction between the feedforward and the partial recurrent nets is in the network topology. The two types of partial recurrent nets, namely, 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 following sections.

3.1.5.1 Feedforward Networks (FFN)

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 3.2 illustrates the schematic architecture of a feedforward network with an input window. The distinct feature of this type of architecture is that the information propagates only in one direction (as indicated by the arrows). Each processing unit (represented as a circle) is called a neuron, and the interconnections between neurons are called synapses. The neuron first calculates the weighted sum of all synaptic signals from the previous layer plus a bias term, and then generates an output through its activation function. A detailed description of feedforward networks can be found in a number of research publications including [Su92].

The most basic approach for handling time series is using an input window that holds a restricted part of the time series. A feedforward network with an input window has been shown to be a more superior architecture than a simple feedforward network [Vemuri93,Ulbricht93,Weigend92]. The input window provides the network with

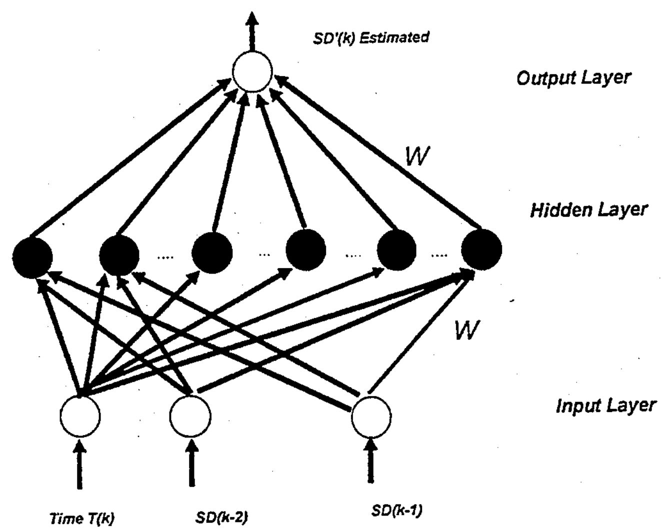


Figure 3.2: Architecture of a Feedforward Network With Input Windows

information on previous states in the form of units in the input layer. This allows for incorporating knowledge on previous states or past values of a time-series. Therefore such an architecture is suitable for modeling spatio-temporal sequencing problems such as modeling the bus transit schedule behavior. Time-delayed FFNs (Feedforward with input windows) have been shown by a few researchers to perform as well as recurrent networks but without the problems of lengthy training and the susceptibility of recurrent nets to be easily trapped into a local minimum [Vemuri93,Ulbricht93, Weigend92].

3.1.5.2 Partial Recurrent Neural Networks

Another approach to modeling using a neural network is to incorporate an internal state to enable it to learn the relationship of an indefinitely large set of past inputs to future states. This is achieved via recurrent connections and hence 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***" [Hertz91]. Partial recurrent networks are networks with characteristics of embedding memory in their architecture.

Partial recurrent networks have been suggested and proven to be applicable by many researchers [Jordan86,Elman90] for dynamic problems involving temporal sequencing. The bus transit schedule behavior prediction problem can be considered as a spatio-temporal problem. The schedule deviation at a timepoint is affected by the schedule deviation at the previous timepoint(s). The spatio-temporal 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 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 utilize information about the previous state and hence are appropriate for the bus transit schedule behavior modeling problem.

A partial recurrent network has an input consisting of two components. The first component is the pattern vector, which was also the only input to the partial recurrent network. The second component is a state vector. This 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 [Jordan86]. These networks are regarded to have memory as the recurrent connections allow the network's hidden units to see their own previous output. Therefore, subsequent behavior can be shaped by previous responses. This network memory concept can be utilized 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 performance. 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, in terms of how far back or how many previous timepoints one should consider, is a yet to be researched.

3.1.5.2.1 Elman Networks

An Elman network is a type of partial recurrent network that is also commonly used for learning to recognize and generate sequences of inputs. The Elman network consists of a single hidden layer feedforward network and in addition a set of additional units at the input level. These additional units are called *context units* and are responsible for the dynamic behavior of the network. Figure 3.3 illustrates a typical architecture of an Elman Network. The number of *context units* equals the number of hidden units. The output values of the hidden units at time step **t** are copied to the context units just prior to the forward

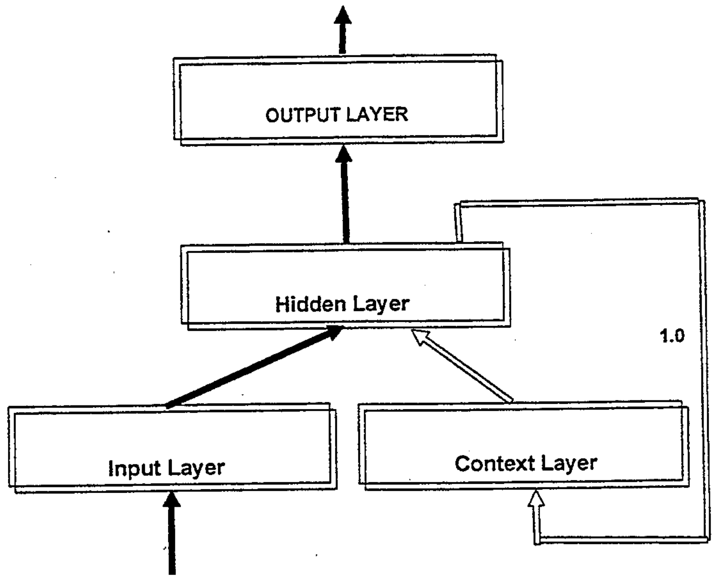


Figure 3.3: Architecture of an Elman Network

computation at time step **t+1**. The context units thus have the characteristics of providing the network with memory, and the internal representations that the network develops are sensitive to temporal context. Thus the effect of time are implicitly represented in the internal state of the network [Elman88]. Mathematically the network representations can be described as follows:

Let

$$x(t) = \begin{pmatrix} x_1(t) \\ \cdot\cdot \\ \cdot\cdot \\ x_p(t) \end{pmatrix}; \; y(t) = \begin{pmatrix} y_1(t) \\ \cdots \\ y_q(t) \end{pmatrix}; \; z(t) = \begin{pmatrix} z_1(t) \\ \cdot\cdot \\ \cdot\cdot \\ z_h(t) \end{pmatrix} \tag{3.1}$$

be the vectors representing the input, output of the network, and output of the hidden units to the network respectively, at time index **t**. Let the matrix **A** contain the feedback weights which are the weight connections from the context units to the hidden units. Matrix **B** contains the input weights (from the input units to the hidden units), **C** contain the output weights (from the hidden units to the output units), and c and d be constant column vectors representing bias values. Then the Elman network can be mathematically described by [Elman88, Kumar91]:

$$z(t) = f[Az(t-1) + Bx(t) + \mathbf{c}] \tag{3.2}$$

Where,

$$y(t) = g[Cz(t) + \mathbf{d}] \tag{3.3}$$

$$f(t) = \begin{pmatrix} f_1(t) \\ .. \\ .. \\ .. \\ f_h(t) \end{pmatrix} ; \quad g(t) = \begin{pmatrix} g_1(t) \\ .. \\ .. \\ .. \\ g_q(t) \end{pmatrix} \tag{3.4}$$

The Elman network has the following properties [Elman88]:

- The layer **I** is fully connected to the layer **I+1**.
- Each context layer is fully connected to its hidden layer. A hidden layer is connected to its context layer with recurrent **1-to-1** connections.
- Each context unit is connected to itself.
- If there is a context layer assigned to the output layer, the same connection rules as

for hidden layers are used.

3.1.5.2.2 Jordan Networks

A Jordan Network [Jordan86] is a type of partial recurrent neural network. The general architecture of a Jordan network is dynamic in the sense that the effects of temporal evolution are captured in the "state" of the network. Jordan networks can model sequential performance and hence are suitable for the bus schedule behavior modeling problem. A key feature of a Jordan net is that there is no explicit representation of temporal order and no explicit representation of action sequences [Jordan86].

A typical Jordan network as illustrated in Figure 3.4 has the following properties:

- The input layer is fully connected to the hidden layer. The hidden layer is fully connected to the output layer (connections are shown in dark arrow lines).
- Output units are connected to context units by recurrent 1-to-1 connections. Every context unit is connected to itself and also to every hidden layer unit (illustrated as grey arrow lines in the figure).
- The number of context units (denoted as "state unit" in Figure 3.4) are equal to the number of output units. The entities- plans, states, and outputs- are all assumed to be represented as distributed patterns of activation on separate sets of processing units [Jordan86]. The plan units and the state units together serve as the input units for a multi layer network. The network can perform arbitrary sequences by taking a plan as input and producing the corresponding sequence. There are two key functions that define the way the network is constructed. The first function ***f*** determines the output

action $\mathbf{x}_n$ at time to be chosen. The second function g determines the state $\mathbf{s}_{n+1}$. Both these functions depend on the constant plan vector as well as the current state vector.

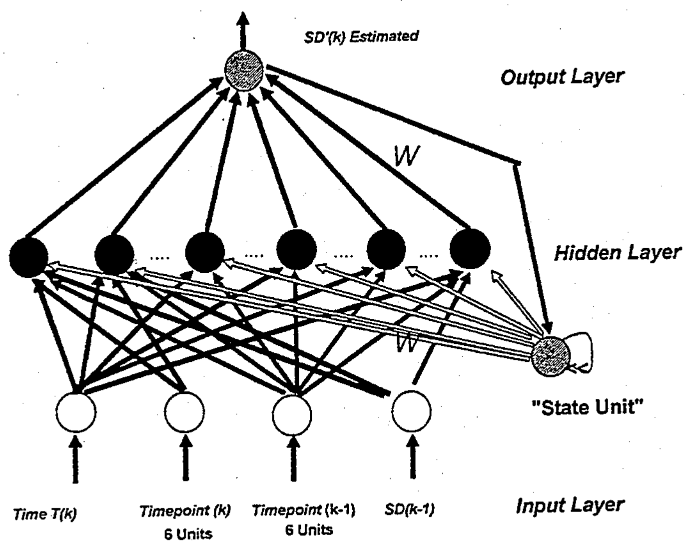


Figure 3.4: Architecture of a Jordan Network

f is referred to as the output function, and g is referred to as the next-state function [Jordan86]. Mathematically this can be described as :

$$x_n = f(s_n, p)\ ; \quad s_{n+1} = g(s_n, p) \tag{3.5}$$

The output function is generally nonlinear. The next-state function is implemented with recurrent connections from the state units to themselves, and from the output units to the state units. This allows the current state to depend on the previous state and on the previous output (which is itself a function of the previous state and the plan) [Jordan86].

3.1.5.2.3 Applicability of Partial Recurrent Networks to Schedule Behavior Modeling

Partial recurrent networks have the distinct characteristics of being able to incorporate past memory and hence can be deemed suitable for modeling the bus transit schedule behavior. There ability to process sequential data is the key reason for their suitability to model the schedule behavior of buses on a transit route. The journey of a bus through a series of timepoints or stops can be considered as a series or sequential information that needs to be incorporated in any modeling of the behavior in terms of schedule deviations. The reason being that bus schedule deviation at a timepoint or stop is influenced by the schedule behavior at the upstream timepoints or stops.

3.1.6 Learning Algorithms

Learning Algorithms (also known as learning rules or learning paradigms) are applied to adjust the connection weights of neural networks. Essentially, a learning paradigm or rule is a procedure for adjusting the weights W so as to make the actual outputs SD'(k) approximate the desired outputs SD(k). Most learning algorithms are based on Calculus, For example, Backpropagation learning uses the delta rule and its variations. Such paradigms use a gradient descent technique to adjust the connection weights in order to decrease the prediction error.

Backpropagation (BP) is a widely used tool in the field of artificial neural networks. It is an efficient and exact method for calculating all the derivatives of a single differentiable function that is referred to as a target quantity (such as an error function) with respect to a large set of input quantities (such as the parameters or weights). Backpropagation assumes that all processing units (also referred to as neurons) and connections are somewhat to blame for an erroneous output or response. The output error is propagated backwards through the connections to the previous layer in order to fix the responsibility for the output error. This process is repeated until the input layer is reached. The name "BackPropagation" derives from this method of handling errors.

The standard backpropagation algorithm has been show to have certain drawbacks. The standard BP is too slow to learn and also has a tendency to be stuck at local minima points. A number of researchers have concentrated their efforts at improving the standard BP. One such modified backpropagation was developed by Scott Fahlman [Fahlman88] and is known as QuickProp. The QuickProp algorithm is described briefly below.

3.1.6.1 QuickProp Learning Algorithm

QuickProp is a faster learning algorithm compared to Backpropagation. It is a second-order method, based loosely on Newton's method, but it is closer to being a heuristic technique rather than a formal technique. The key feature of QuickProp is that it uses information about the curvature of the error surface. This entails the computation of the second order derivatives of the error function. QuickProp assumes the error surface to be locally quadratic and attempts to jump in one step from the current position directly into the minimum of the parabola. QuickProp computes the derivatives in the direction of each weight. The Quickprop algorithm was developed using two major assumptions. First, it is assumed that the error vs. weight curve for each weight can be approximated by a parabola whose arms open upward. Second, that the change in the slope of the error curve, as seen by each weight, is not affected by all the other weights that are changing at the same time [Fahlman88].

The procedure involves first computing the gradient with regular backpropagation, and subsequently attempting a direct step to the error minimum using the following expression [Fahlman88]:

$$\Delta(t+1)\, w_{ij} = \frac{S(t+1)}{S(t)-S(t+1)}\,\Delta(t)\, w_{ij} \tag{3.6}$$

where:

w_{ij}	weight between units i and j
$\Delta(t+1)$	actual weight change
$S(t+1)$	partial derivative of the error function by w_{ij}
$S(t)$	the last partial derivative.

Using the update expression given in Equation (3.1), if the current slope is somewhat smaller than the previous one, but in the same direction, the weight will change again in the same direction. The step may be large or small, depending on how much the slope was reduced by the previous step. If the current slope is in the opposite direction from the previous one, that means that the minimum has been crossed and that the current position is on the opposite side of the valley. In this case, the next step will place us somewhere between the current and previous positions.

QuickProp Learning Parameters

There four key parameters of the Quickprop learning function. These are [Fahlman88, SNNS3.2] :

η: Learning parameter, specifies the step width of the gradient descent.

μ : Maximum growth parameter, specifies the maximum amount of weight change (relative to 1) which is added to the current change.

ν : Weight decay term to shrink the weights

d_{max} : the maximum difference $d_j = t_j - o_j$ between a teaching value t_j and an output o_j of an output unit which is tolerated, i.e. which is propagated back as $d_j = 0$.

3.1.6.2 QuickProp Learning Algorithm for Partial Recurrent Networks

The quickprop learning algorithm described, in the previous section, for a feedforward network is modified for the training of partial recurrent networks in the following way [SNNS3.2]:

1. Initialization of the context units. In the following steps, all recurrent links are

assumed to be absent, except in step 2(f).

2. The following steps are executed for each pattern in the training sequence:

   (a) the pattern and forward propagation are inputed through the network,

   (b) The error signals of the output units are calculated by comparing the computed output and the target (teaching) output,

   (c) the error signals are back propagated,

   (d) the weight changes are calculated,

   (e) *Only on-line training:* The weights are adapted,

   (f) the new state of the context units are calculated according to the incoming links,

3. *Only off-line training*: weights are adapted.

The parameters for this modified quickprop algorithm are the same as for the regular feedforward versions of this algorithm described in section 3.1.6.1.

3.1.7 Selection of a Suitable ANN Architecture

The selection of a suitable architecture for a given application depends on the problem characteristics. Joseph et. al [Joseph92] in their study stated that in order to select a suitable neural network architecture, four basic factors need to be considered. They are:

- <u>The nature of inputs</u> - the schedule behavior data of bus transit systems consists of continuous values and so only the models that take continuous values as inputs are considered.

- The knowledge or availability of the desired outputs - This distinction essentially defines whether supervised or unsupervised neural network architectures are appropriate. Since the AVL system provides information regarding actual arrival times, information about desired outputs is known. Hence supervised neural networks are more appropriate.
- The number of hidden layers - The number of hidden layers define the overall architecture of the artificial neural network. Typically, only one hidden layer is used. In this study, only one hidden layer is considered.
- Non-Linearity : the dynamic operation of buses on a transit route network results in the assumption that schedule behavior is often non-linear, so the network architecture selected must be able to handle non-linearity. On any given day, schedule behavior of buses along a particular direction and at specific timepoint 'TP_k' is often related to the schedule behavior of buses in the previous timepoint 'TP_{k-1}'. Hence the spatial context is also an important consideration.

3.2 Statistical Techniques

Statistical methods are conventional techniques that are often applied to develop mathematical models of dynamic systems. The theory behind various statistical techniques has evolved over several decades and is known to be well developed. Hence it is imperative that any investigation of modeling of a dynamic system using new and promising techniques should include a comparison with traditional statistical approaches. A number of statistical techniques exist and one has to carefully select the appropriate technique or set of techniques to as a model development strategy. The type of problem and the availability and type of data are the key issues influencing the selection of the appropriate approach.

There are two important statistical techniques that have been applied to forecasting problems in the past : Standard regression and Time-series. A time series has been defined in the literature as a set of observations obtained my measuring a single variable regularly over a period of time. The state of the art and practice has been well established with both regression methods and time-series methods. The field of research and theory is mature and very well developed. These methods have been used in areas such as process plant control, weather forecasting, stock market forecasting, predicting or estimating traffic flow [Makri83,Watson93,Weigend92,Weiss90]. The various time-series methods based on the Box-Jenkins methodology that are listed in the literature are: Auto Regressive Integrated Moving Average (ARIMA), Auto Regressive Moving Average (ARMA, X11ARIMA,)etc.These models have been shown to be flexible and successful in modeling a large variety of time series [Weigend92,Weiss90].

Conventional techniques for modeling and prediction are almost all based on linear or linearised models. The literature reviewed indicated that the practical success of these approaches is limited due to this linearity requirement apart from other drawbacks such as ravenous data requirements and extensive skills at interpreting the results. Traditional methods such as regression are also suitable for developing causal models. Makridakis et al. [Makri83] discuss in detail the traditional techniques for modeling and prediction.

3.2.1 ARIMA Models : Theoretical Background

ARIMA models combine three different types of processes:

- autoregression (AR)
- differencing to strip off the integration (I)
- moving averages (MA).

Three components are based on the concept of random disturbances or shocks. Between two observations in a series, a disturbance occurs that somehow affects the level of the series. Theses disturbances can be mathematically described by ARIMA models. The most general ARIMA model involves all three components. The general ARIMA model is conventionally written as ARIMA(p,d,q), where p is the order of autoregression, d is the degree of differencing, and q is the order of moving average involved.

Autoregression is a process in which each value in a series is a linear function of the preceding value or values. In a first-order autoregressive process only the single preceding value is used; in a second order process the preceding two values are used; and so on. An autoregressive process is conceptually shown to be one with a "memory" in the sense that each value is correlated with all preceding values.

Time series often reflect the cumulative effect of some process. The process is responsible for changes in the observed level of the series but is not responsible for the level itself. The levels themselves are the cumulative sum of the changes in each period. A series that measures the cumulative effect of something is called integrated. The way in which integrated series are studied is by differencing or in other words by looking at the changes from one observation to the next. The difference from one observation to the next is often small for series that wander. This stationarity of the differences is highly desirable from a statistical point of view. The integrated process can be viewed as one with a perfect memory of the previous value- but only the previous value.

In a Moving Average (MA) process, each value is determined by the average of the current disturbance and one or more previous disturbances. The order of the moving average process specifies how many previous disturbances are averaged into the new value. A moving average process is the most difficult of the processes to visualize.

The ARIMA models are developed based on the modeling procedure described by Box and Jenkins that allows one to construct the best possible model for a given series. The ARIMA models represent data as an explicit structure. The ARIMA models can be mathematically represented as [Box70]:

$$\varphi(B)\Phi(B^{S})\nabla^{d}\nabla_{S}^{D}(Z_t - c) = \theta(B)\Theta(B^{S})a_t \tag{3.7}$$

where

B: is the Back-shift operator (i.e., $Bx_t = x_{t-1}$);

$\nabla = 1 - B$; s = seasonality, a_t = white noise ;

$\varphi(B)$ and $\Phi(B^S)$ are nonseasonal and seasonal autoregressive polynomials respectively;

$\theta(B)$ and $\Theta(B^S)$ are nonseasonal and seasonal moving average polynomials respectively;

Z_t = series (transformed if necessary) to be modeled.

The modeling approach is described in the next section.

3.2.2 ARIMA Modeling Process

The Box-Jenkins approach essentially involves the following steps [Box70] :

- Model Identification
- Parameter Estimation
- Consideration of Alternative ARIMA models, if necessary
- Diagnostics

Identification of the processes underlying the series forms the first and most subjective step. The three integers p,d, and q in the ARIMA(p,d,q) process need to be determined. The identification process begins with first determining whether the series is stationary or not. This is done using a plot of the key variable. This is needed as the identification process for the AR and MA components requires stationary series. A stationary series has the same mean and variance throughout. When a series is non-stationary, then the series is transformed until a series that is stationary is obtained. The method of transformation is differencing. Each value in the original series is replaced with the difference between that value and the preceding value. The order of differencing is the number of times the differencing is done. The typical values are 0 and 1. The next step in the identification process is to obtain the values p and q. In the case of nonseasonal processes, the values of p and q are determined from the autocorrelation function (ACF) and partial autocorrelation function (PACF) of the series. The autocorrelation function calculates the autocorrelations at lags 1,2, and so on. The partial autocorrelation function gives the corresponding partial autocorrelations at intervening lags.

Once the series is identified the coefficients of the model are estimated. The estimation process involves iterative calculations. The maximum-likelihood coefficients are estimated and a new series is added that represents the fit or predicted value, the error, and the confidence limits on the fit. The new series (fit and residuals) are used for diagnosis of the model for selecting the best model.

The diagnosis step involves checking the following [Box70, SPSS]:

- The ACF and PACF of the error series should not be significantly different from 0. If the first or second order correlation is large then the model is mis-specified and hence an alternative model needs to be estimated.
- The residuals should be without pattern. A common test for this is the Box-Ljung Q statistic. Q at a lag of one quarter of the sample size and not more than 50 should be evaluated and must not be significant.

The ARIMA type of Box-Jenkin models have the following advantages:

- Explicit structural relationships that can be clearly understood;
- Consistent performance compared to ANNs;

3.2.3 Drawbacks of Statistical Techniques

Parametric statistical approaches have been shown to have difficulty with detecting important changes in the data. In monitoring high-dimensional time-series, explicit mathematical models that contain parameters of unspecified value are "fit" to the data. The fitting process than determines statistically significant values for the parameters and these are used to characterize the time-series and the underlying phenomenon or system it represents. However more often model mis-specification limits the ability of the model to capture essential features of the data-generating mechanism accurately and with a high degree of reliability [Burke93].

The literature reviewed indicated that regression methods are not very good in their predictive capabilities and usually are more useful for prediction of linear systems. Statistical methods have poor predictive capabilities for non-linear dynamic systems. These methods necessitate that a mathematical model be defined and the various parameters be known. Hence they are often difficult to formulate and do not give a good predictive model. Mathematical models such as these are never a perfect representation of a dynamic system. They often lead to underfitting or overfitting because of the need to assume the underlying distribution function and also the need to estimate the various parameters in the mathematical models. The statistical methods also have to overcome the problem of high correlation that exists in many problem situations. The standard statistical methods such as test of significance (t-statistic, R^2 etc) and correlation tests do not always imply causality or predictive power. Regression techniques also require that a particular form for the distribution function be assumed. Depending on the curve, one can end up in poor predictive values through underfitting or overfitting of data to a selected function. In addition regression techniques are computationally intensive. For **n** inputs, there will be $2\mathbf{n} + \mathbf{n}(\mathbf{n}-1)/2$ terms in the regression model. If there are **k** terms, 2^k-1 regression models are possible [Burke93].

3.2.4 Statistical Modeling Issues

The following modeling issues are raised and addressed while identifying and selecting the type of statistical technique to be used for developing bus transit schedule behavior models :

- Whether to model the schedule deviations at a given timepoint on a given route and direction of travel as a function of schedule deviations at the previous timepoints on the same route and direction of travel, i.e. in the form :

$$\mathbf{SD(k) = \varphi\,[\,SD(k-1), SD(k-2)...SD(k-n)\,].}$$

- Extensions of the basic Box-Jenkins AR(p) model described in the above equation to include moving average MA(q) terms, differencing, seasonal components and other data transforms.
- Whether to include spatial information about location of timepoints as predictor variables within the time series model.
- How to tackle missing or erroneous data?
- What kind of relationship does the bus schedule behavior data exhibit? Whether it is linear or nonlinear; stationary or non-stationary; random or seasonal.

Some of the issues such as the missing or erroneous data are common to both ANN models and ARIMA model. These issues are addressed in Chapters 4.

3.3 Summary

This chapter provides background information on the different modeling techniques suitable for the schedule behavior modeling problem. A historical overview of ANNs,advantages and disadvantages of ANN models was presented. Basic guidelines for selection of a suitable ANN architecture(s) were discussed. The theoretical background on two important ANN architectures (i) Feedforward networks, and (iii) Partial recurrent networks was also presented. This chapter also described the learning alogirthm: QuickProp. The standard backpropagation algorithm has been shown to be slow and inefficient and hence a faster version developed by Scott Fahlman called Quickprop was deemed appropriate and suitable for developing the ANN models.

Partial recurrent networks have the distinct advantage of being able to represent sequences. This key attribute formed the basis for adopting this technique for investigating the schedule behavior modeling problem. The partial recurrent network architecture incorporates dynamic behavior through its representation structure and hence is suitable for non-linear modeling. Good internal modeling is an important step towards the development of efficient bus transit operational control strategies. Partial recurrent networks by virtue of their structure have the potential for good internal modeling of schedule behavior of buses on a route. Good internal modeling can be relevant and important for real-time bus transit service control. Given sufficient information on the system (location of the bus, schedule deviation at the current location), which constitutes the inputs to the system, real-time service control requires some prediction of the schedule deviation of a bus at subsequent timepoints or stops. Furthermore, the prediction must be able to change as the bus travels along with other traffic on any given route and whose influence is unknown and unpredictable. Therefore, any function approximation that models the schedule behavior of a bus must be adaptable. In this study the primary focus is on investigating the application of three different ANN architectures. They are : (i) Feedforward Networks (ii) Jordan Networks, and (iii) Elman Networks. The main purpose for including this chapter was to provide a basic understanding of ANNs. This chapter's main contribution is that it provides a rationale for choosing the ANN approach for modeling schedule behavior of buses by describing the advantages and also by providing an overview of past applications.

This chapter's contribution is also towards providing the theoretical background on the ARIMA modeling technique that will be used for modeling the schedule behavior of buses. This chapter also briefly described the relevant statistical modeling issues that are important for developing schedule behavior models. The reason for selecting the ARIMA modeling technique is because of its strong theoretical underpinnings making it more reliable and robust and therefore suitable for modeling problems. The second reason for selecting the ARIMA modeling technique was because of the stated objective of this research effort to perform a comparative analysis with ANN models.

CHAPTER 4

SCHEDULE BEHAVIOR MODEL: CONCEPTUAL FRAMEWORK AND DEVELOPMENT PROCESS

4.1 Introduction

Modeling is a very general mathematical concept that has been used to solve a variety of engineering problems. Statistical techniques have been often used to develop models to understand the relationship between an observed phenomena and the factors that affect or influence its characteristics. The problem of schedule behavior modeling can be construed as a time series prediction problem. Hence time series modeling and prediction techniques are applicable. Chapter 3 provided the neccessary justification for selecting the various neural network architectures and also the statistical techniques to model schedule behavior data. Chapter 3 also provided the theoretical background on a selected set of techniques for modeling the bus schedule behavior. The main purpose of this chapter is to describe in detail the conceptual framework for modeling bus transit schedule behavior using the aforementioned techniques. The modeling procedure seeks to investigate the issue of how many timepoints in the upstream section of the route should one include in the modeling and hence prediction at a given timepoint? Since the objective of this dissertation is to discern the relative performances and hence merits of different approaches, different models are developed using different lengths of input sequences, i.e, considering 1,2,3,4 and 5 previous timepoint schedule behavior information. An important point to note is that schedule behavior modeling is a very complex and difficult problem requiring spatio-temporal considerations. A review of the relevant literature indicated that such a problem has not yet been researched in depth and hence very little or no guidelines are available for developing the modeling framework and also the appropriate modeling technique for the schedule behavior problem .

4.2 Modeling Approaches

In prediction modeling there are two basic approaches that have gained prominence and 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 key variable being predicted [Weigend92]. The key factors that affect schedule behavior and cause schedule deviations are:

$$SD\,(R_i, j, k, T) = \varphi\,(Traffic, Driver, Vehicle, Environment, (Un)Loading) \tag{4.1}$$

where,

SD = Schedule Deviation,

R_i = Route i,

j = Direction,

k = Timepoint Location,

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/unloading characteristics at each timepoint on a given route. The second modeling approach is to assume that all the information (on key factors affecting schedule behavior) that is available, has already been represented by the values of the key variable being predicted [Weigend92]. 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/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 :

$$SD(k) = \varphi[\{SD(k-1), SD(k-2)\ldots SD(k-n)\},\{T(k),T(k-1)..\}] \tag{4.2}$$

where,

SD(k) denotes the schedule deviation at timepoint ***k*** on a specific route and 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 ..etc). In this study the focus is on developing schedule behavior models using Equation 4.2 for a given route and direction of travel. The various model structures investigated in this study are described in later sections of this chapter.

4.3 Explicit Models versus Implicit Models of Schedule Behavior

One of the key questions while developing models of a particular system is whether to consider an explicit structure on an implicit one. The claim that is often made is that an explicit structure to model a system of interest is inherently better than an implicit structure that an artificial neural network has and that is often referred to as the *black box*. The various possibilities for the actual structure of a schedule behavior model are discussed below.

Consider a route which has six timepoints and where the scheduled and actual arrival times at a timepoint are known. The schedule deviation at timepoint ***k*** is therefore known for various trips during different times of the day. The conventional time series approach would propose a relationship of the form:

$$SD_{k,t} = \varphi_{k,t-1} SD_{k,t-1} + \varphi_{k,t-2} SD_{k,t-2} + \cdots + \epsilon_t \qquad (4.3)$$

where ***k*** indicates timepoint location and ***t*** indicates the different times of the day which in our context represents the trip number, and SD denotes schedule deviation. However, such an approach, which essentially has been used in the past to obtain distributions of schedule deviations for different times of the day, takes no account of the spatio-temporal or structural relationship that exists between a measurement at one point and a measurement at another; namely, that the bus passing through a timepoint moves to the next, in a time that varies because of a number of factors such as traffic, loading and unloading characteristics (reflecting demand), number of bus stops in between, driver characteristics etc. Moreover, these characteristics are inherently reflected in the arrival time of a bus at subsequent timepoints. In other words, the lags, p, (in the expressions of the kind $\varphi_{k,t} SD_{k,t-p}$) should be regarded as variables that are related to the travel time (or the speed of the bus between two timepoints) and not as fixed quantities. Consequently, the set of forecasting model equations become complicated considerably and such a model form cannot be ascertained using existing techniques. Moreover, such an approach does not capture the spatio-temporal sequencing relationships that the schedule deviation of a bus is believed to have. Hence we can infer that no matter how well a time series model, as represented by Equation 4.3, might appear to fit the schedule deviation data, their functional form is not necessarily one that is consistent with the schedule behavior characteristics of buses on a route. Now consider the expression given below:

$$SD_{k,t} = \varphi_{k-1,t} SD_{k-1,t} + \varphi_{k-2,t} SD_{k-2,t} + \cdots\cdots + \epsilon_t \qquad (4.4)$$

where ***k*** indicates timepoint location and ***t*** indicates the different times of the day which also refers to the trip number or the scheduled arrival time of the day, and SD denotes schedule deviation. This expression is used to model the schedule deviations of buses traveling in a particular direction, at consecutive timepoints on a route and at a specific scheduled time of tha day at the starting timepoint. This model form captures the inherent relationship that exists between the schedule behavior at consecutive timepoints and thus is a more realistic and useful form that needs to be ascertained. The second approach to overcome the problem of inconsistent functional form using the conventional time-series equation,as given by Equation 4.4, is to develop the schedule behavior models using artificial neural networks.

The basic question of whether explicit representation is better than implicit representation is quite difficult to answer [Kirby94]. While the traditional mathematical techniques such as time-series have a very well established theory, the ANN methods are new and look promising. As discussed in Chapter 3, there are relative merits in using both the explicit representation as done by the ARIMA model and the implicit model representation as done by an artificial neural network model. Since no prior work exists at modeling the schedule behavior of buses using the time-series approach, the best answer is to investigate both the implicit and explicit modeling techniques and then do a comparative analysis as to which provides the more accurate and reliable predictions for the schedule behavior modeling problem. Many researchers have stated that theoretically it may be proven that a particular model is representationally more powerful than another because it is capable of representing more complex curves and functions [Foster92, Pollard92,Weiss90]. However, in practice any learning method or system using a theoretically less powerful representation may yield better predictions than one with a potentially stronger and complex representation [Weiss90]. According to Weiss et al., [Weiss90] what is important is that for any given representation, a learning method must be capable of finding good fits within that model. Weiss et al. also state that a method must use data efficiently and find a compact and often simpler solution at an appropriate level of complexity.

The model structures proposed in this and the previous section, as defined by

Equations 4.2 and 4.4, clearly implied that there are a large number of possible definitions for SD(k). The model structures chosen for experimentation in this effort were a compromise between fully capturing the system status, and keeping the size of number of input parameters to a minimum. While the basic framework for schedule behavior model was as defined in this section, some modifications of the model structure, in terms of additional independent variables, was chosen for experimentation during the model development process. The basic form of the schedule behavior model defined by Equation 4.2 was selected for model development in this study.

The following sections describe the schedule behavior model development effort using the two basic techniques described in Chapter 3 - artificial neural networks and statistical methods. The description also illustrates the complexities with conducting experiments for developing ANN based models.

4.4 Artificial Neural Network Models

The development of effective ANN models requires experimentation with network topology, input variables, learning coefficients etc. It is an exercise in art and science. The scientific part involves trial and error experimentation using network design guidelines discussed in Chapter 3. The various steps in the ANN modeling process is described below.

4.4.1 ANN Modeling Process

The artificial neural network models were developed using the following procedure. The design of the modeling experiments required a detailed planning and was tedious due to the fact that there are no standard procedures for developing ANN models. The development of the ANN models could be done by a trial and error procedure especially with regards to the selection of the network architectures, learning parameters and network training.

The following is the step by step process for developing the ANN models:

STEP 1:	Data Preprocessing
STEP 2:	Network Selection
STEP 3:	Learning Algorithm/Update Function Selection
STEP 4:	Weights Initialization
STEP 5:	Network Training
STEP 6:	Network Testing.

In order to develop the ANN models the Stuttgart Neural Network Simulator (SNNS) was used. The neural simulator had the capabilities to develop all of the proposed ANN network architectures. It is an X-windows based application developed using the OSF/Motif class library. It also provided graphic features to trace the propagation of error over each iteration. The various steps in the modeling process are described in detail in the following sub-sections.

4.4.1.1 Data Preprocessing

Data preprocessing is the critical step in artificial neural network modeling. It covers about half the modeling process. Data preprocessing involved two important steps: (I) Elimination of outliers or noise, and (ii) Data scaling. A detailed description of data preprocessing step is provided in Chapter 5.

4.4.1.2 Network Architectures

As discussed earlier, three basic neural network architectures are examined in this study: (I). Feedforward Networks and (ii). Elman Recurrent Networks and (iii) Jordan Recurrent Networks. All of the above network architectures considered had three layers, with an input layer, hidden layer and an output layer consisting of units. Since it was required to produce continuous valued output, linear units were used in the output layer. Similarly, since the inputs were continuous valued, linear units were used in the input layer. For the units in the hidden layer, logistic function was chosen.The number of hidden units was varied in order to obtain a reasonable performance of the model on the test data set.

For each of the above-mentioned architectures, three different sets of models, as described below, were experimented depending on the spatial and temporal sequencing information being considered for input to the artificial neural network. These model structures essentially define the different network topologies based on the nature of input information. They also represent networks of different complexities for schedule behavior model development experiments. ANN architectures are denoted as IxHxO, where I, H, and O represent number of input, hidden, and output units respectively. The network structures used in this study are defined in the following section.

4.4.1.3 ANN Model Structures

Different network topologies are considered depending on the number and content of input variables. The model development process was designed such that different ANN model structures were experimented with based on the basic framework of the schedule behavior model defined by Equation 4.2. The different model structures experimented are referred to as Model Sets in this study. In all, two different model sets were designed for investigation. The important distinguishing characteristics of the three model sets in the amount of information i.e., inputs included for analysis. In Model Set I the focus is to experiment and study the effect of sequencing on learning and neural network modeling. The notion here is that the schedule deviations at a timepoint k-1 affect the schedule deviations at the next timepoint k and hence the sequential information may provide better learning of the schedule behavior of the system. This in turn helps in more accurate predictions of schedule deviations at the subsequent timepoints on a given route and direction of travel.For Model Set II, the schedule deviation information along with the timepoint information represented by the timepoint index on a route was also presented as input to the network. This was done in order to ascertain if providing the timepoint information would have any influence and therefore would affect the model's accuracy. The scheduled arrival time was also included as input to the ANN models, since it gives us an indication of the time of the run, i.e. whether it is during peak hour etc. The basic structure of all the models are described below.

CASE NUMBER	INPUT SERIES LENGTH	INPUT UNITS
IA	1	Scheduled Arrival Time (T(k)),Schedule Deviation SD(k-1)
IB	2	Scheduled Arrival Time (T(k)), Schedule Deviation SD(k-2), Schedule Deviation SD(k-1)
IC	3	Scheduled Arrival Time (T(k)), Schedule Deviation SD(k-3), Schedule Deviation SD(k-2), Schedule Deviation SD(k-1)
ID	4	Scheduled Arrival Time (T(k)), Schedule Deviation SD(k-4), ScheduleDeviation SD(k-3), Schedule Deviation SD(k-2), Schedule Deviation SD(k-1)
IE	5	Scheduled Arrival Time (T(k)), Schedule Deviation SD(k-5), Schedule Deviation SD(k-4), Schedule Deviation SD(k-3), Schedule Deviation SD(k-2), Schedule Deviation SD(k-1)

Table 4.1 : Artificial Neural Network Model Structures for Model Set I

Model Set I: Models using only Schedule Deviation Information at previous Timepoints.

The various models designed and investigated under this model set formed the main focus and the most important task of this study. The model structures designed and investigated under this model set included experimenting with various lengths of the input series. It was postulated that the behavior observed at an upstream timepoint might be of help in making such a prediction. The principal goal of such an experimentation was to investigate the question of how many upstream timepoints, which is described as the length of the input series, should be included for analysis in order to accurately predict the schedule deviation at a given site. Four different models with different input series length ranging from n=1 to n=5 were designed and developed using the various ANN model architectures. The length of the input series represents a way to define the memory of the schedule deviation series. Table 4.1 details the various scenarios for the model set I that were designed based on the length of the input series. For all the five cases IA-IE, only one output unit,namely, Schedule Deviation SD(k) was used. This means that the ANN models were designed for predicting only one timepoint ahead. In addition to investigating the impact of the length of the input series, the impact of providing the timepoint information in the form of position al information along a route was investigated. The next section briefly describes the model structure.

Model Set II : Models Developed using Schedule Deviation and Timepoint Information

This model set differs from model set I in that the timepoint information (i.e, positions k, k-1, k-2 along the route) was also provided to the network as inputs. This was done in order to investigate the effect of providing knowledge about the spatial location of the buses on the route to the ANNs. The following model set was included in the investigation in order to ascertain if the inclusion of the timepoint information (essentially the position of the various timepoints along the route) had any impact on model performance. It was decided to first check if the inclusion of timepoint information had any impact and hence only two two cases ($n = 1, n = 2$) were selected for the experiments. If the timepoint information had any impact on model performance then the cases involving longer input series $n = 3$ to $n = 5$ would also be considered for investigation.

CASE NUMBER	**INPUT SERIES LENGTH**	**INPUT UNITS**
IIA	1	Scheduled Arrival Time (T(k)), Schedule Deviation SD(k-1), Timepoint (k), Timepoint (k-1);
IIB	2	Scheduled Arrival Time (T(k)), Schedule Deviation SD(k-2), Schedule Deviation SD(k-1), Timepoint (k), Timepoint (k-1), Timepoint (k-2);

Table 4.2 : Artificial Neural Network Model Structures for Model Set II

4.4.1.4 Learning Algorithm/Update Functions

Since both the input (time and location,schedule deviation at previous timepoints etc.) and output (schedule deviation) variables are known quantities, the schedule behavior modeling using artificial neural networks constitutes a *supervised learning* problem and hence supervised learning algorithms such as QuickProp can be applied. QuickProp described in Chapter 3 is a faster and more efficient version of the standard Backpropagation algorithm.

4.4.1.5 Weight Initialization

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

4.4.1.6 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) was computed for each iteration of the training process. MSE was used as a stopping criterion during the training phase. The learning rate parameter was chosen in the range of 0.0005-0.00001.

4.4.1.7 Network Testing

The networks were tested on the test data set and the MSE and SSE were computed. The absoute errors were also computed on the test data set in order to be used for conducting a comparative analysis between various modeling approaches.

4.5 Statistical Models

The time series model structure selected for investigation was based on Equation 4.4 described in section 4.3. An important point to note is that the time series model does not use the full set of independent variables included in the definition of SD(k). The statistical software package SPSS/PC+ was utilized to develop and test the ARIMA models.

4.5.1 Exponential Smoothing

The exponential smoothing technique was first applied on the data set to see if a simple model can explain the underlying schedule behavior model structure. The exponential smoothing parameter α was close to zero indicating that the exponential smoothing method simply predicts the overall mean and does not use information from the most recent observations. Hence it is of little use and a more sophisticated technique such as the ARIMA method is perhaps a more appropriate modeling technique for the schedule behavior data set.

4.5.2 ARIMA Technique

As described in Chapter 3, the ARIMA model is one of the most advanced time series models, and therefore was chosen for application in this research effort. In order to be able to compare statistical modeling with the ANN models, two model structures used in ANN model development process were selected for the statistical modeling process. In addition to the average absolute error, the average percentage error, and the distribution of error for the ARIMA models were computed in order to assist in comparative analysis of the various models. The two modeling structures that were chosen for investigation and comparative analysis are shown are in Table 4.3. These model structures were selected in order to be able to conduct a comparative analysis between the ANN modeling approach and the ARIMA technique. The Model A case is identical to the model case IC shown in Table 4.1. The results of the best fit ARIMA model was compared with the results of the IC set of ANN models.The Model B case is the same as the case IE shown in Table 4.1. The results of the best fit ARIMA model with this structure was compared with the results of the IE set of ANN models. In order to select the best model among the various ARIMA models for a particular model structure the Akaike's Information Criteria (AIC) was considered.

The ARIMA model development process involved the resolution of a number of issues. The primary issue was of missing/erroneous values. Since most data sets invariably contain missing or erroneous values, an effective technique was required to address this issue. The simplest approach adopted for this study was to replace the missing values with historical averages [Terry]. The risk involved with such extrapolation is that such an approach can lead to the selection of a wrong form of a model. This complexity of missing/erroneous values is one of the reasons why time series models appear to be not well suited for wide-scale application to schedule behavior modeling problem.

MODEL	INPUT SERIES LENGTH	INPUT UNITS
A	3	Scheduled Arrival Time (T(k)), Schedule Deviation SD(k-3), Schedule Deviation SD(k-2), Schedule Deviation SD(k-1);
B	5	Scheduled Arrival Time (T(k)), Schedule Deviation SD(k-5), Schedule Deviation SD(k-4), Schedule Deviation SD(k-3), Schedule Deviation SD(k-2), Schedule Deviation SD(k-1);

Table 4.3 : Time Series Model Structures

4.6 Comparative Analysis

In order to conduct a comparative analysis, a set of measures of performance are defined. Multiple performance criteria were used in order to fuly evaluate the performance of the four modeling techniques. Ideally, multiple measures of performance provide a more realistic, meaningful, and non-relativistic evaluation approach for a comparative analysis of different modeling approaches [Smith95].

4.6.1 Measures of Performance

The performance of the modeling techniques was evaluated using the average absolute error, average percentage error (PE_{avg}), and distribution of absolute error. These are briefly described below.

The average absolute error is the primary measure of model accuracy. The measure is an indicator of how far the model's schedule deviation values differ from the actual schedule deviation values. The average absolute error for the test data set is used as measure to compare the performance of the various models.

The average percentage error PE_{avg} is calculated for each point in the test data set (having n patterns in the test data set) using the following expression:

$$PE_{avg} = \sum_{i=1}^{n} \frac{(SDact_i - SDpred_i)}{SDact_i} x\ 100 \qquad (4.5)$$

where,

SD_{act} is the actual schedule deviation;

SD_{pred} is the network predicted schedule deviation.

The distribution of absolute error serves as an indicator for evaluating the model's ability to underestimate or overestimate the schedule deviations at the next timepoint. In this study, the distribution of error is measured in terms of the threshold intervals of the absolute error: The intervals defined are (in minutes): greater than 0 and less than or equal to 2, greater than 2 and less than or equal to 3, greater than 3 and less than or equal to 4, and greater than 4. The distribution of absolute error is also a criteria that provides a mechanism to evaluate the potential for practical implementation of the various approaches.

4.6.2 Statistical Test for Comparison of Average Absolute Error

In order to conduct a comparative statistical analysis of the performance of various models the Wilcoxon test was chosen. Past studies that have compared ANNs with Statistical models had suggested the use of Wilcoxon test for evaluating the statistical significance of the models' performance[Smith95]. The Wilcoxon signed-rank test was used to evaluate the statistical significance of differences between the various ANN and statistical models. The Wilcoxon test is a nonparametric test that infers from paired sample cases. In this study, the pairs are the absolute error estimated by two models at a given scheduled arrival time. The null hypothesis of the Wilcoxon test states that the mean of the two populations, in this case defined as the absolute errors for the two models, are equivalent i.e, $\mu_1 - \mu_2 = 0$. The alternate hypothesis is that the mean for model 1 is greater than the mean for model 2 i.e, $\mu_1 - \mu_2 > 0$. The Z statistic is calculated to determine the level of significance. A detailed description of the Wilcoxon test can be found in any statistical reference book including in [Brieman73].

4.7 Summary

The main contributions of this chapter is the description of a basic modeling approach for the schedule behavior modeling problem and the formulation of the schedule behavior model development process using the modeling techniques described in Chapter 3. The most suitable approach for schedule behavior modeling was judged to be the time-series approach essentially due to lack of sufficient data on causal factors. The relative merits of explicit versus implicit modeling approaches were also highlighted in this chapter in order to justify the stated objective of investigating the schedule behavior modeling using both the modeling approaches.

Different modeling techniques and model structures for developing schedule behavior model were presented in this chapter. The various modeling techniques selected required different levels of effort, knowledge, and modifications to the basic model structures discussed in section 4.3. The model development effort especially in the case of the ANN paradigm requires a series of trial and error experiments in order to resolve the issues such as: number of inputs, number of hidden units, initial weights, criteria for error propagation, number of iterations, learning parameters values etc. Therefore, the design of the experiments required trial and error approach in the selection relevant factors. A number of ANN models were developed for each model structure and architecture using different values for the relevant parameters and the model which showed the best performance in terms of the MSE and SSE for the test data was selected as the appropriate model. The next chapter presents a detailed description about the case study used for investigating the feasibility of developing schedule behavior models.

CHAPTER 5

CASE STUDY : DESCRIPTION AND DATA SET DEVELOPMENT

5.1 Introduction

Tidewater Regional Transit operates a 170 bus system and has implemented an automatic vehicle location system for real-time monitoring and supervision. AVL Data was obtained from the Tidewater Regional Transit's AVL system. The headways on TRT's route network vary on this from 15 minutes to 1 hour during different times of the day and for different days of the week. There are three different scheduling plans for each route depending on the day of the week, i.e., Weekday covering Monday to Friday; Saturday; and Sunday. For this study, we are limiting our scope to weekday operations. Sufficient data could not be obtained for Saturday and Sunday operations.

The AVL data is stored on the microVAX system that is the central computer used for the real-time monitoring of the bus operations. TRT began its installation of the AVL system in 1989. The AVL system has faced several calibration problems. TRT has been using the AVL information for generating schedule adherence reports. The computer system has had capacity problems because the same computer system has been used for other inventory, payroll and other in-house personnel management tasks. This makes it difficult for TRT to store AVL data that can be used for the development of intelligent processing techniques. The data is stored as history files with a date stamp attached to the file name, eg. HISTF930319.DAT. These files contain all events taking place from the start of operations to the last operation in the night. TRT's AVL system records all events ranging from time point to timepoint update of actual arrival times of buses. The data is represented as record types and event types. TRT's system consists of nine records and about 75 event types. All the data pertaining to the different record types and eventypes are stored in one single file, the history file HISTFXXXXXX.DAT (XXXXXX stands for year, month and date). Therefore preprocessing was necessary in order to obtain data relating to schedule behavior. The following data is to be extracted from the history files : Route Number, Route-Block, Direction, Vehicle Number, TimePoint Location, Schedule Arrival Time, and Actual Arrival Time at a timepoint, Incident Event Type. In addition, the following information was collected from TRT's system: Location of Transfer Points, Incident description and route diversions. One important point to note is that location data updates are available only at timepoints along any route and not at every bus stop. Currently AVL history data is stored on magnetic tapes for about two weeks and then erased. The problem has been that the history data files are very large and TRT's VAX system doesn't have the desired storage capacity. Data for 26 weekday operations was downloaded from TRT's microVAX system. The raw data from the AVL system is stored in binary format.

The sample route considered for this study is Rt 23 of TRT's bus transit system.. The headways on this route are 30 minutes. The bus travel time is 30 minutes. Figure 5.1 shows an illustration of route 23. The route has six timepoints with one of them being a transfer point to routes 9 and 23. Tables 5.1-5.2 illustrate the characteristics of the some the

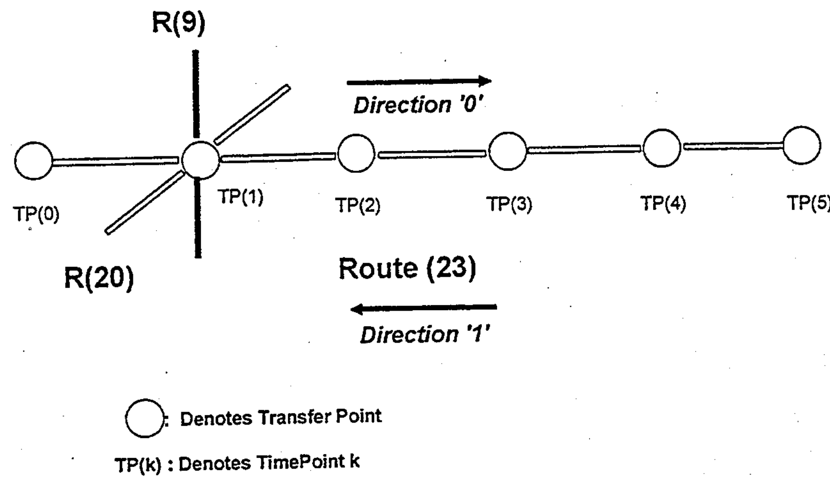


Figure 5.1: An Illustration of the Sample Test Route 23

important routes on TRT's network including Rt. 23. This route was selected because of the simplicity in the route characteristics such as 30 minute headways with 30 minute travel time between end points. This means that any given time only one bus exists along a particular direction of the route. This reduced the complexity in the data preprocessing step. In addition, the reason for selecting only one route was that it is easier and less complex to use a single route to investigate the feasibility of schedule behavior modeling using advanced techniques. The second reason is that no prior research work exists on applicable modeling approaches and hence it is appropriate to first investigate the feasibility of modeling and ascertain a suitable modeling approach before any large scale development of schedule behavior models for a transit route network. Table 5.3 shows a sample data sheet obtained from TRT's AVL system.

Route #	Start Point	End Point Location	Headway (Min) *	Headway (Max) *	Total BTT @ **
1	Pembroke East	Monticello/Charlotte	15	60	88/92
9	Va Beach Blvd/Scott	Monticello/Charlotte	30	30	30/39
20	Atlantic/ 68th St.	Monticello/Charlotte	15	60	92/97
23	Va Beach Blvd/Scott	Medical Tower	30	60	25/18
29	Pleasure Hs/Shore	Pembroke East	60	60	90/85
36	Pembroke East	Lynnhaven Mall	60	60	28/29

Table 5.1 : TRT's Sample Route Network Characteristics

<u>NOTE</u>: * : In Minutes

$BTT_@$: Bus Travel Time for One Direction

** : 88/92 indicates BTT in minutes for the two directions.

Transfer Point #	Routes Involved	Location Description
TP_1	1, 20, 29, 36	Pembroke East
TP_2	20, 29	First Colonial Rd, Laskin Rd
TP_3	1, 29	Pleasure House Rd, Shore Dr
TP_4	1, 20, 9	Monticello Ave.,Charlotte St.
TP_5	*9, 20, 23*	*Va Beach Blvd., Scott St.*

Table 5.2 : Sample Timed-Transfer Points and the Routes Involved

EV#	SCH. TIME		ACT. ARR. TIME			ROUTE	BLK #	NODE	DIR	VEH#	SD
56	4	40	4	47	14	1	241300	1580	2	2402	-7
56	4	44	4	55	39	1	243300	1580	2	1721	-11
58	4	45	4	47	31	1	241300	1580	2	2402	-2
56	4	58	4	58	34	1	244300	1580	2	22	0
58	4	54	5	1	53	1	243300	1590	2	1721	-7
55	5	3	5	5	59	1	243300	401	2	1721	-2
55	5	10	5	9	51	1	243300	1590	2	1721	0
56	5	14	5	12	44	1	238300	1580	2	1951	1
55	5	15	5	21	34	1	243300	1580	2	1721	-6
58	5	18	5	15	13	1	244300	1780	1	22	2
58	5	24	5	24	12	1	238300	1590	2	1951	0
56	5	29	5	16	43	1	26400	1580	2	2482	12
56	5	29	5	42	44	1	235300	1580	2	1821	-13
55	5	30	5	27	42	1	244300	870	1	22	2
56	5	30	5	30	21	1	233300	1580	2	1601	0
55	5	30	5	38	0	1	243300	870	2	1721	-8
55	5	33	5	33	50	1	238300	401	2	1951	0
58	5	39	5	46	17	1	235300	1590	2	1821	-7
55	5	40	5	37	15	1	238300	1590	2	1951	2
55	5	43	5	50	10	1	243300	1780	2	1721	-7
55	5	45	5	51	46	1	238300	1580	2	1951	-6
55	5	48	5	49	44	1	235300	401	2	1821	-1
55	5	49	5	44	49	1	244300	1580	1	22	4
55	5	53	5	52	7	1	241300	980	2	2402	0
58	5	53	5	55	24	1	26400	110	1	2482	-2
59	5	54	5	48	35	1	244300	1590	1	22	5
58	5	54	5	49	13	1	244300	1590	2	22	4
55	5	55	5	53	44	1	235300	1590	2	1821	1
58	5	50	6	5	39	1	233300	1780	1	1601	-15
56	5	59	6	1	4	1	239300	1580	2	2012	-2
55	6	0	6	5	25	1	235300	1580	2	1821	-5
55	6	0	6	6	16	1	238300	870	2	1951	-6
55	6	0	6	19	42	1	233300	870	1	1601	-19
55	6	3	6	2	5	1	244300	401	2	22	0
59	6	3	6	10	15	1	243300	110	2	1721	-7
58	6	5	6	10	24	1	243300	110	1	1721	-5
58	6	9	6	8	29	1	239300	1590	2	2012	0
58	6	9	6	8	52	1	239300	1590	2	2012	0
55	6	10	6	5	3	1	244300	1590	2	22	4
59	6	12	6	0	39	1	241300	1840	2	2402	11

Table 5.3 : Sample data sheet obtained from TRT's AVL System

5.2 Data Preprocessing

The data preprocessing task is the most critical one for ANN modeling. Nearly 60 % of the time is spent on processing the raw data to a format suitable to the modeling process. The data preprocessing task involved the following three important steps:

- Data Extraction/Conversion
- Data Normalization
- Data Representation/Encoding

These three steps are discussed in detail in the following sections.

5.2.1 Data Extraction/Conversion

The raw data from the history files were preprocessed in order to compute the desired schedule behavior indicator, namely schedule deviation. Data preprocessing was also necessary for converting the raw data that is in binary format to an ASCII file and also for extracting only the event type that stores the actual arrival times information. The raw data or the history files were obtained on TK150 DEC tapes. The data was transferred from the tapes on to the IBM RS6000 Unix based machines using the DEC's tape drives available with the VAX system in the Physics Department. This had to be done for two reasons: a) Academic Computing Center does not have any DEC tape drives, and b) the data was stored on the tapes using VAX's backup command. The data files had to be first retrieved on the VAX system and then transferred to the RS6000's using File Transfer Protocol (FTP). The AVL history files were preprocessed using C programs to extract only the desired data fields and for the selected sample route.

5.2.2 Data Normalization

Normalization of inputs has been suggested by many researchers as a way to improve the performance of the learning algorithm and thus achieve better results [Widrow79]. The notion being that large magnitude of inputs leads a problem wherein the hidden units get driven to near their limits and thus leading to the exclusion of nonlinear effects. Therefore the suggestion is to normalize the inputs to a scale of 0 to 1 or -1.0 to +1.0. The goal is to keep the magnitudes of the inputs to a hidden layer unit to be around 1 or 2. Normalization or scaling is an important consideration for gradient descent methods such as standard backpropagation since they are sensitive to scaling. In this study the main variable, schedule deviation was normalized to values in the range -1.0 to 1.0. Momentum, if properly chosen, alleviates bad scaling to some extent. The main emphasis in the ANN literature on initial values has been on the avoidance of saturation, hence the desire to use small random values. How small these random values should be depends on the scale of the inputs as well as the number of inputs and their correlations. Standardizing inputs removes the problem of scale dependence of the initial weights. It has also been stated in the literature that both TanH and sigmoid are useful in different applications [Mehra92]. If the ends of the data range are important to define the output then a TanH function is best applied as you will have the most learning at the ends of the range (-1 , 1). If the data increases or decreases unidirectionally then sigmoid (0,-1) would be a better choice. It's all how the data is encoded and presented to the net. When the scaled inputs are around zero, you have very little learning. Researchers [Mehra92, Prechelt94] have also been able to show that it is possible to substantially increase learning rates in networks by encoding inputs as binary patterns using -1,1. However such binary encoding is not suitable to our problem.

The data was normalized using minimum and maximum values of the variables over the entire data set. The scaling of the variables was accomplished using the expression:

$$Xnorm = (2.0 * \frac{X}{MAX-MIN}) + ((-2.0 * \frac{MIN}{MAX-MIN}) - 1.0) \tag{6.1}$$

where,

X denotes the variable to be normalized;

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 minutes and MIN = 300 minutes. The scaling using the above expression converts the data into the [-1,1] interval. The scaling is important so that the units in which they are given do not effect the net's output (i.e., the inputs should be either unit-less ratios or else chosen so that percentage changes are the same across monotonic transformations of input values). It can speed convergence to have most or all inputs scaled identically to the output function. 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 same expression given in Equation 5.1 but using the corresponding schedule deviation values. The timepoint data is also transformed into a binary vector. For our case study (Rt. 23) there are six timepoints located on the route. Hence a vector of length six was considered and the timepoints were transformed in the following manner. 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. A training set consisting of 24 days of data, and two test sets consisting of one day's data each.

5.2.3 Data Representation/Encoding Schemes

The training data selection problem is the first problem that needs to be solved when applying ANNs using supervised learning techniques. The training data selection problem essentially entails the selection of an appropriate data set for training the ANN. Such a training data set must contain the underlying relationship that the ANN should capture. This problem is a complex one as in most cases the underlying relationship is unknown. Once a training set has been selected, the next problem in the sequence that needs to be solved is classified as representation problem. Representation involves obtaining an answer to the question of "how to design the ANN structure such that there is at least one solution that learn the training set" [Nasci94].

The representation problem concerns the following network design issues [Nasci94]:

- How many hidden layers are to be used for modeling?

- How many units should be there in each hidden layer?
- Which functions should we use for the hidden units?

The problem defines the number of input and output units. In the case of Feedforward neural networks with input windows, the length of input window defines the number of units in the input layer.

Once the network input and output representation has been designed to solve the schedule behavior modeling problem, it is still necessary to look for the network internal representation. The representation problem is then to choose the ANN structure such that an internal representation exists, i.e. that there is at least one set of weights (parameters) that can reproduce the training data set with a small error. A review of the current literature revealed that there is very little theory to help in this task. Hence the scheme is to apply a trial and error procedure to arrive at a good representation.

Hornik et al. established that a feedforward ANN with as few as one hidden layer using arbitrary squashing activation functions (such as TanH, Sigmoid etc) and no squashing function at the output layer are capable of approximating virtually any function of interest from one finite multi-dimensional space to another to any degree of accuracy, provided sufficiently many hidden units are available. Stinchcombe and White extended this result and showed that even if the activation function used in the hidden layer is a rather general nonlinear function, the same type of feedforward ANN is still a universal approximator. Funahashi, Cybenko, Ito etc. also obtained similar results.

As can be seen that while there are proofs in the literature to prove that there is a feedforward ANN with just one hidden layer using squashing or non-squashing function in the hidden layer that provides a relationship for the input-output mapping, it is still not possible to deduce from these proofs the ANN topology (number of hidden layers and number of units in each hidden layer) or, once the network topology is chosen, how to determine the network weights (parameters). Another point to note is that the theoretical literature providing the above proofs, does not provide any clarifications to the question: which function is more suitable to be used as the activation function for the hidden units given a specific criterion such as minimum number of hidden units. There is yet to be any concrete theoretical background to this issue. Hence the only solution to the data representation problem for our specific ANN modeling of schedule behavior data is to adopt a trial and error procedure and experiment with various number of hidden units and various squashing functions.

5.3 Problems with Data Handling and Preprocessing

The AVL system at TRT uses a microVAX system as its central computer for two way data and voice communication. The systems file handling and storage required some guessing in order to extract the desired information from the binary files. The field formats given in the system manuals did not correspond to the actual raw data files stored. Hence a trial and error method was used to decipher the field format of the history files. The procedure was automated in the sense that a computer program written in C was used rather than a manual technique and this process saved considerable time. Once the field format of the data file was known a simple program in C was written to extract the necessary information for schedule behavior modeling. Figure 5.2 illustrates a flow chart of the data collection process.

One key problem encountered was the lack of sufficient storage disk space on the TRT's computer system. The large size of each days history file (approx 1.6 MB) creates a storage problem despite the data being in binary format. The large size of the history file demands a huge storage capacity which adds to the cost of system. TRT's solution was to store the history files as system backup files on tapes. Even then TRT's data storage capacity was only two weeks of data. Transit systems need to address this critical problem at the system implementation stage and should be aware of the data storage needs in order to develop utilize the data for application of advanced analysis and evaluation techniques to enhance system operational performance. In addition absurd/unrealistic values of actual arrival times had to be replaced by average values.

5.4 Guidelines for AVL Data Storage and Handling

The problems faced and the experience gained from the task of collecting AVL data has led to provide some useful suggestions for data storage and handling. Most transit systems implementing AVL systems are bound to experience problems of various degrees with data accuracy, data storage and handling. The large amount of data that is generated in real-time creates a difficult but solvable data storage and handling problems. It is necessary for transit systems to fully understand the importance of the issue right at the system planning stage. Adequate data storage and handling capabilities must be provided along with a fast and reliable computer that forms the heart of the AVL system.

5.5 Summary

The data preprocessing task was the most important step in the modeling process. A set of C programs were written in order to filter and extract the desired data. The normalization of the data was performed as suggested in the literature by many researchers involved in applying ANNs for modeling and prediction problems. The argument in favor of normalization of the data set was overwhelming so as not to ignore it. In the case of any real world application of ANN techniques, how the data is represented is at least as important as what ANN paradigm is chosen to model. The procedure adopted emphasizes informative data representation and encoding and uses generic preprocessing techniques to transform the raw data into an informative input format.

CHAPTER 6

MODELING RESULTS AND ANALYSIS

6.1 Introduction

The different types of schedule behavior models described in the previous chapter were developed using the data set from the case study and their performance analyzed. The models were developed using the training data set defined in Chapter 5 and the model performance evaluated using the test data sets. The measures of performance defined in Chapter 4 were used as the criteria for evaluating the performance of the various models.

The first step involved the point-to-point mapping between the schedule deviation and the time of the day for a specific route and a specific direction. Figure 6.1. shows a crossplot of the schedule deviation series data. The scattering of the data points indicates that there is a nonlinear relationship between the time of day and the schedule deviations. This suggests that nonlinear modeling techniques are more appropriate for modeling schedule behavior and hence justifying our modeling approach.

6.2 Results : Model Set I

The class of models in model set I were developed using the training data set and their performance evaluated using the test data set. The test data set defined in Chapter 5 allows for an objective analysis of each model's performance. The training and performance

```
         +-----------------------------------------+
      20 |                                         +
         |                   1                     |
         |                                         |
         |   1                         11        |
  S      |   3331221 2244214   1312 2221 11 32    |
  A     0|   34363454542644228754375533333432      +
  _      |   11 141221     1374 4345543231231      |
  M      | 1          12111  31 1142      11     |
  I      |                        1                |
  N      |                        1                |
     -20 |              1                          +
         |                     1      1          |
         |                                         |
         |                                         |
         |                                         |
     -40 |            1                            +
         +-----------------------------------------+
          6    8     10  12    14    16     18  20

                TIME_HR : Time of Day (Hrs)

SA_MIN : Schedule Deviation (minutes),  TIME_HR : Time of Day
```

Figure 6.1: ScatterPlot of Schedule Deviation vs Time of Day.

6:00 am to 8:00 pm., 03/15/93. Direction 1, Route 23

evaluation characteristics are defined in this section.

6.2.1 Training Characteristics

The Mean Squared Error (MSE) values for model set I after training was completed for the different ANN architectures are shown in Table 6.1. Although the MSE values are the error measure drops significantly during the first 100 iterations and then decreases gradually. In order to minimize the risk of overtraining or undertraining, the following not zero, theyare relatively small and hence it can be inferred that the training was done sufficiently in order to provide a reasonably good fit with the training data. Figure 6.2 illustrates the error profile measured error profile measured in terms of MSE after various iterations during training. The MSE was used as a measure for monitoring the training process. It can be noted that procedure was adopted:

Step 1: Select Training Parameters : Initialize Random Weights.

Step 2: Train the network for 100 iterations.

Step 3: Stop Training. Save the network characteristics.

Step 4: Test the model on the test data set and save the MSE and the network.

Step 5: a) Repeat Step 2 to Step 4.

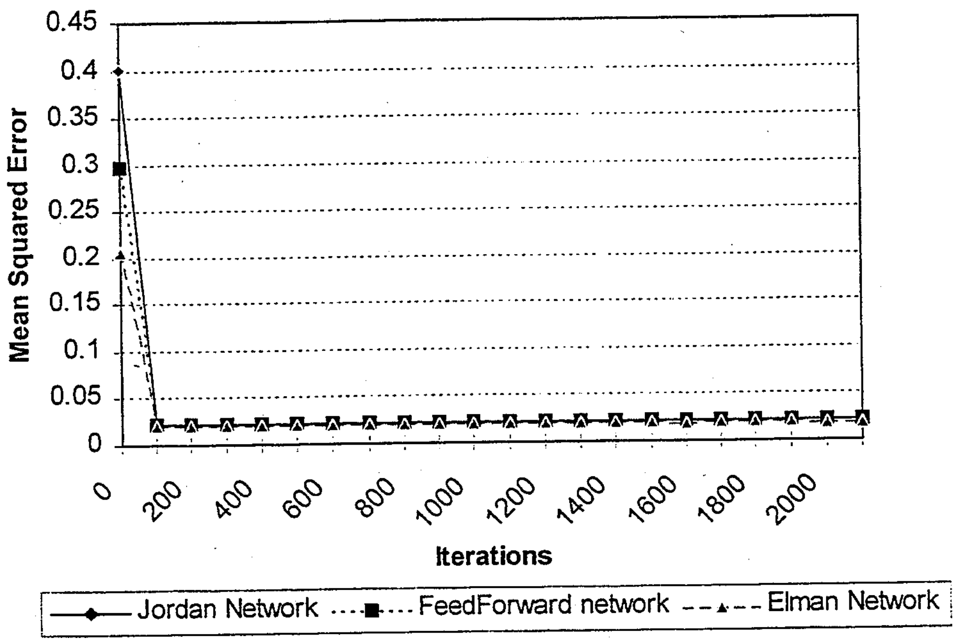
b) Stop Training if there is no change in the MSE values from the previous iteration cycle.

c) Evaluate all the MSE's. Select the network characteristics with the lowest MSE value as the model.

ANN Training Procedure

Figure 6.3 illustrates the distribution of connection weight values. It indicates that the connection weights are well distributed and approximate a normal [Rathburn93]. Table 6.2

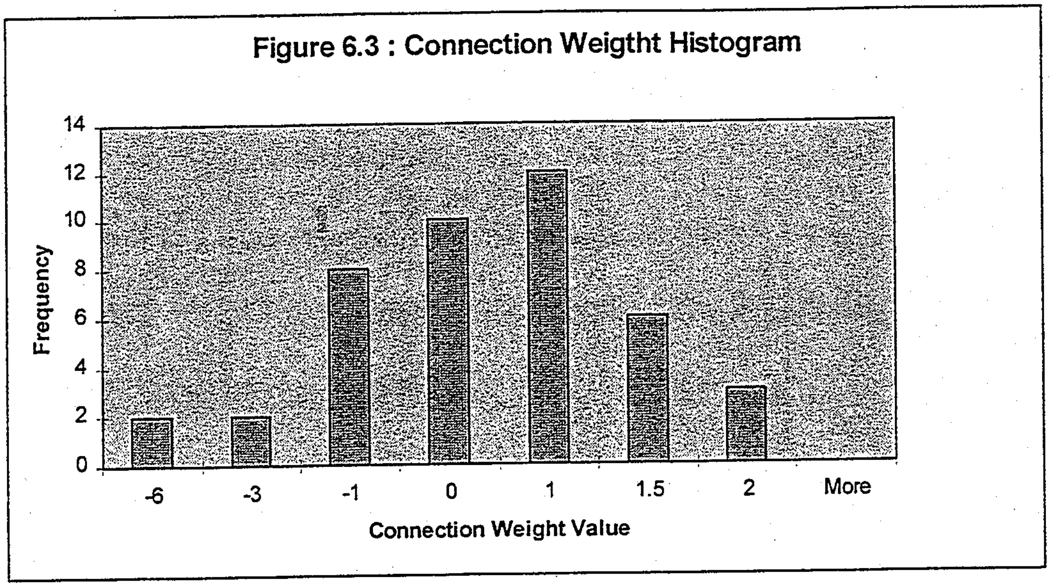
Figure 6.2 : ANN Network Learning Measured as Mean Squared Error During Training : Model Set IA



MODEL SET I						
ANN	MEASURE	IA	IB	IC	ID	IE
FeedForward Network	Mean Squared Error	0.0248	0.0195	0.0188	0.0202	0.0142
Elman Network	Mean Squared Error	0.0194	0.0185	0.0179	0.0222	0.0095
Jordan Network	Mean Squared Error	0.0211	0.0192	0.0185	0.0195	0.0126

Sample Size for Training : 2124

Table 6.1 : Mean Squared Error for Different ANN Architectures and Model Structures after Training : Model Set I



ANN Architectures (IxHxO)					
ANN Type	IA	IB	IC	ID	IE
FeedForward Network	2x5x1	3x6x1	4x7x1	5x8x1	6x10x1
Elman Network	2x5x1	3x6x1	4x7x1	5x8x1	6x10x1
Jordan Network	2x5x1	3x6x1	4x7x1	5x8x1	6x10x1

Table 6.2 : Best ANN Architectures (IxHxO) at the End of ANN Modeling Process : Model Set I

shows the best architectures obtained for the three different ANNs and distribution. Thus we can conclude that the network learning process worked appropriately for the five model structures. The architectures were obtained after a series of experiments with different number of hidden units for any given model structure and ANN type. The number of hidden units for a particular model structure was kept the same in order to allow for unbiased evaluation of the performance of the different ANN types. The I indicates the number of input units , the O indicates the number of output units and in this study was constant at 1 since the focus was only to predict one timepoint ahead. The number of hidden units was varied and the ANN models were evaluated both for their performance in their training in terms of mean squared error and also on the test data using the average absolute error. The number of hidden units that provided the best are shown in the Table. One important point to note is that it is nearly impossible to obtain a theoretically optimum number of hidden units. It would require a large number of experiments and hence would involve a lot of time. One of thumb rules that was adopted for this ANN modeling experiments was to keep the number of hidden units not very large (around twice the number of input units) in order to allow for faster training and also prevent the possibility of too many loose connections that have very little influence on the model's performance but which would increase the value of the error function.

6.2.2 Performance Evaluation Characteristics

The measures of performance defined in Chapter 4 were computed for the three ANN architectures and for the various model structures. The measures of average absolute error and average percentage error of the ANN models using the five model structures IA-IE are shown in Tables 6.3. The table indicates that the average percentage errors are rather high with the highest accuracy achieved being only 79%. One plausible reason can be that the training data set may not be adequate for providing a good generalization. While the average percentage error for the various models seems high it is important to ascertain the distribution of the absolute error in order to evaluate the model's practical usefulness within a reasonable error distribution. Table 6.4 shows the distribution of error for each of the models in model set I. The important point to note here is the distribution of absolute error in the range of less than 1 minute and greater than 3 minutes. In real world application, absolute error values less than 1 minute for the schedule deviation will not have much practical significance while errors greater than 3 minutes can be potentially important on operational management and hence it is necessary to see what percentage of the cases that the models give an estimate for the absolute error greater than 3 minutes.

Table 6.4 indicates that the performance of the various models improves as the input series length(n) is increased from n = 1, to n =3 but then drops out rapidly for n =4, and n= 5. The plausible reason is poor training due to the presence of too many free parameters (weights) as a result of greater number of input and hidden units for model structures ID and IE (representing n = 4 and n =5) compared to model structures IA-IC. The presence of

MODEL SET I						
ANN	MEASURE	IA	IB	IC	ID	IE
FeedForward Network	Average Absolute Error (min)	1.082	1.081	0.966	1.066	1.77
	Average % Error	32.48	32.11	21.9	22.44	51.79
Elman Network	Average Absolute Error (min)	1.071	1.006	1.006	1.002	2.021
	Average % Error	32.15	29.88	22.84	21.09	59.12
Jordan Network	Average Absolute Error (min)	1.132	1.044	0.905	1.307	2.298
	Average % Error	33.98	31.01	20.55	27.52	67.23

Sample Size = 135

Table 6.3 : Model Set I :- Error Measures for Different ANN Architectures and Model Structures on Test Data Set

greater number of connection weights can lead to white noise which would result in poorer generalization. One potential way to improve the performance under such conditions is to "prune" the network by selectively stopping further weight changes on certain network connections. It can be noted that the distribution of absolute error is not normal justifying the selection of Wilcoxon test for assessing the statistical significance of differences between the various models' average absolute error.

		DISTRIBUTION OF ABSOLUTE ERROR % Cases with Absolute Error of			
MODEL	ANN TYPE	0 - 1 (min)	1 - 2 (min)	2 - 3 (min)	> 3 (min)
IA	FeedForward	60.06	29.24	4.41	6.29
	Elman	64.15	25.78	4.41	5.66
	Jordan	61.95	26.41	5.03	6.61
IB	FeedForward	60.94	29.30	4.29	5.47
	Elman	67.19	23.83	3.51	5.47
	Jordan	62.89	29.69	1.95	5.47
IC	FeedForward	65.93	25.19	5.92	2.96
	Elman	65.18	26.67	4.45	3.7
	Jordan	69.63	23.71	3.70	2.96
ID	FeedForward	62.07	25.29	8.05	4.59
	Elman	67.82	20.69	4.59	6.90
	Jordan	51.72	34.48	6.90	6.90
IE	FeedForward	35.94	34.38	15.62	14.06
	Elman	46.88	31.25	7.81	14.06
	Jordan	28.12	31.25	14.06	26.57

Table 6.4 : Distribution of Absolute Error for various Models :
Model Set I : Test Data Set

6.2.4 Analysis

Table 6.5 shows the results of the Wilcoxon signed-rank tests for the model set IC. Examining the results shown in Tables 6.3-6.5, model structure IC provided the best results in terms of the measures of performance for all the three different ANN model architectures. Among the three architectures, the Jordan network performed better than the other two architectures for the case study. While the average absolute error is shown to be the lowest for the Jordan Network in Table 6.3, statistical test results shown in Table 6.5 indicate no significance at the 95 % confidence level. The results presented in Tables 6.3-6.5 for each model structure and for each ANN architecture are analyzed and discussed in detail below:

Null Hypothesis	**Alternative Hypothesis**	**Z - Statistic**	**Significance at 0.05 level?**	**Preferred Model**
$\mu_{FF} - \mu_{JO} = 0$	$\mu_{FF} - \mu_{JO} > 0$	-1.605	No	Jordan
$\mu_{EL} - \mu_{JO} = 0$	$\mu_{EL} - \mu_{JO} > 0$	-1.01	No	Jordan
$\mu_{EL} - \mu_{FF} = 0$	$\mu_{EL} - \mu_{FF} > 0$	-0.709	No	FeedForward

Table 6.5 : Wilcoxon Signed-Rank Tests: Model Set IC

where,

μ_{FF} : Average Absolute Error for Feedforward Network

μ_{JO} : Average Absolute Error for Jordan Network

μ_{EL} : Average Absolute Error for Elman Network

6.2.4.1 Model IA

For the model structure IA, the Elman network architecture provided the lowest level of absolute error. However, the performances of the three architectures was not significantly different. The distribution of the absolute error also showed similar behavior. The Jordan network showed a greater tendency (6.61 % of the cases) to err on its estimates by more than 3 minutes compared to the Elman network (5.66 %). Although Table 6.3 suggests that the various model's can't be expected to produce very accurate results, the results from Table 6.4 indicate that the various ANN models should be able to estimate within reasonable range of the actual schedule deviation values. The higher levels of absolute error for the various model architectures suggests that perhaps the model structure defined as IA may not be appropriate for the schedule behavior modeling problem.

The performance of the three ANN architectures for this model structure are shown in Figures 6.4a-6.4c. In general, the models performed a moderate job of tracking the fluctuations in the schedule deviation of buses during different times of the day. The models performed poorly at tracking large schedule deviation values.

6.2.4.2 Model IB

Table 6.3 indicates that the Elman network performed better than the other two ANN architectures for this model structure. However, the average percentage errors' were still high (29.68 % to 32.11%) for this model structure. But when compared to model structure IA, the various ANN architectures under model structure IB showed improved performances. The average percentage timepoint for model structure IA. In addition, the

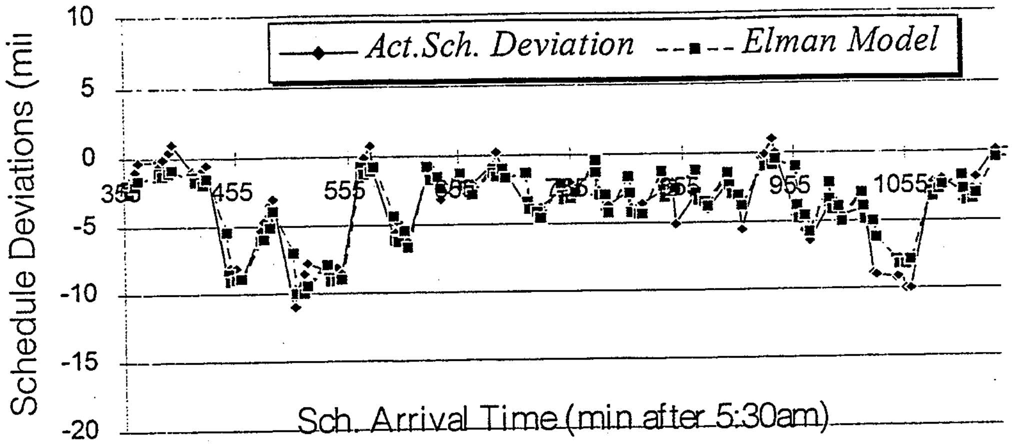


Figure 6.4a : Elman Network Performance on Test Data :Model IA

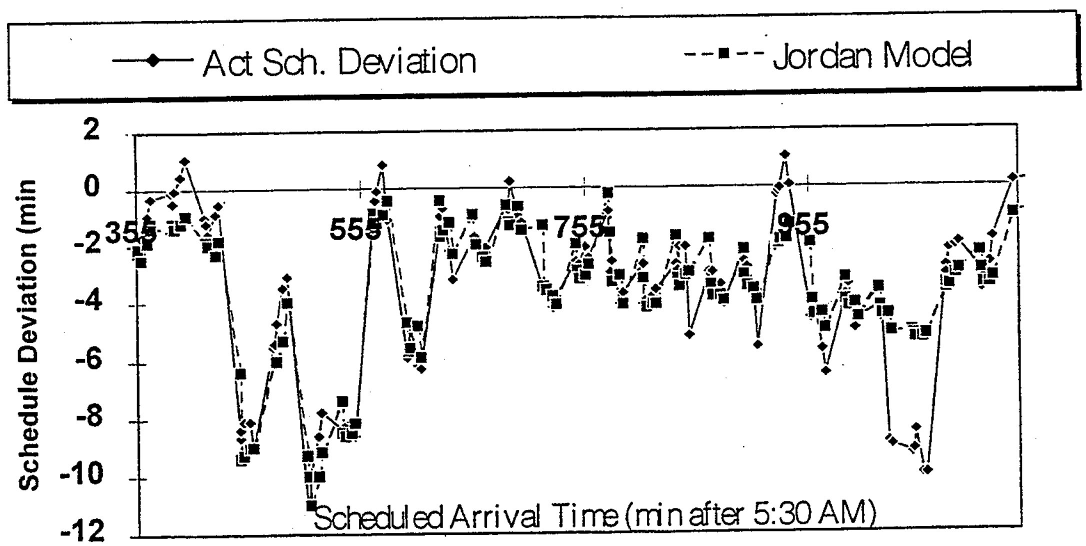


Figure 6.4b : Jordan Network Performance on Test Data: Model IA

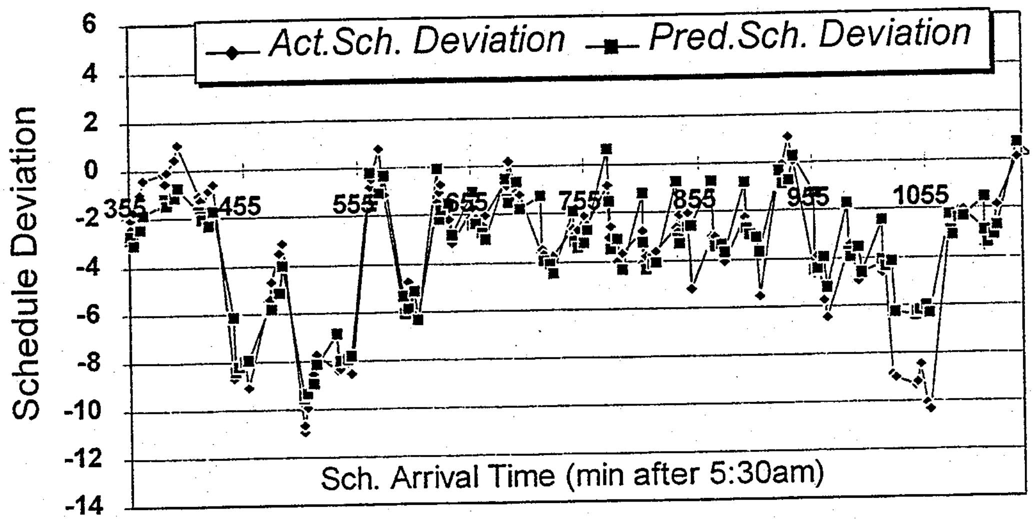


Figure 6.4c : FFN Performance on Test Data : Model IA

better performance of the Elman network error for the Elman network decreased to 29.88 % from 32.15 % for the model structure IA. The Feedforward network showed very insignificant change in the average percentage error for model structure IB compared to structure IA. The Jordan network model also showed a decrease in the average percentage error from 33.98 % to 31.01 %. This can be explained by the fact that the networks are provided with a longer input length series. In other words the networks know about the schedule deviation information from the previous two timepoints instead of just one previous suggests that a partial recurrent architecture is able to capture and use more information from the previous two timepoints while training is in progress.

Table 6.4 shows that the Jordan network using structure IB showed a lower tendency (7.42 % of cases) to err by more than 2 minutes compared to structure IA (11.64 % of cases). The Jordan network also showed a lower percentage of cases (7.42%) with wrong estimates of more than 2 minutes compared to the Elman (8.98%) or the Feedforward networks(9.76%).

Figures 6.5a-6.5c illustrate the performance of the three ANN architectures under model structure IB. When comparing Figures 6.4a-6.4c with 6.5a-6.5c it can be inferred that the networks performed better especially while estimating large schedule deviation values. The ANN networks are able to capture the fluctuations in the schedule deviation values much better using the model structure IB compared to IA.

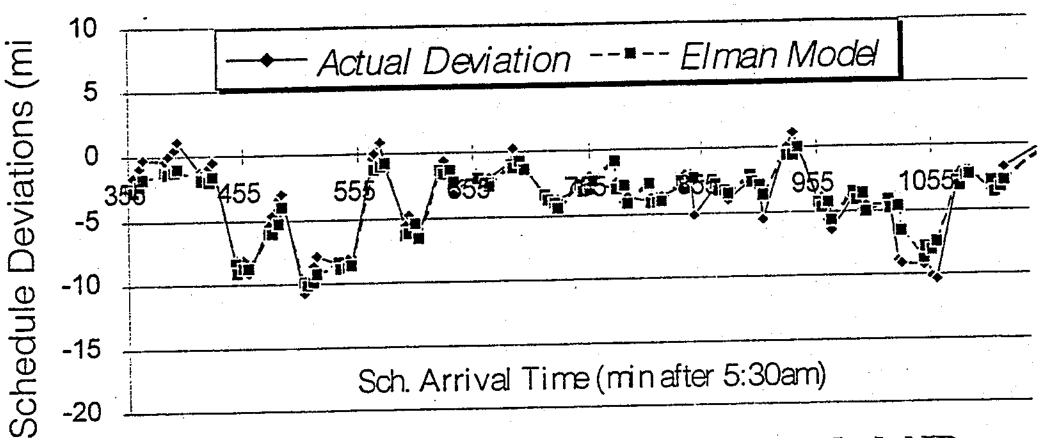


Figure 6.5a : Elman Network Performance : Model IB: Test Data

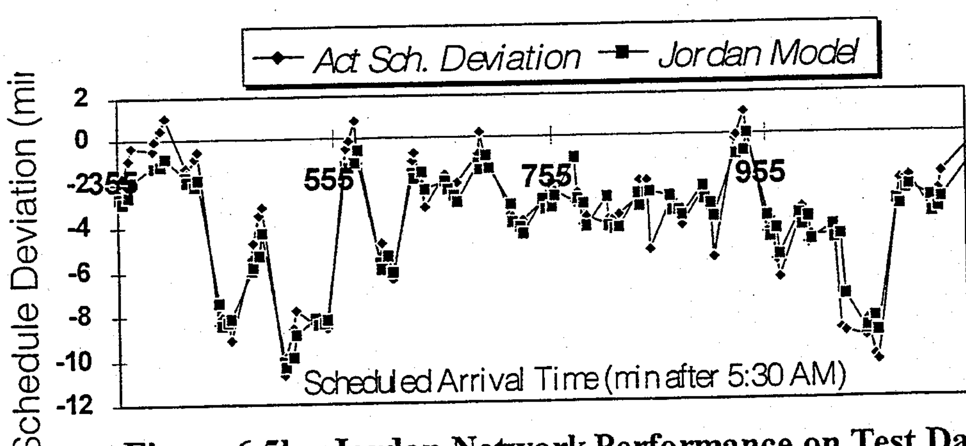


Figure 6.5b : Jordan Network Performance on Test Data : Model IB

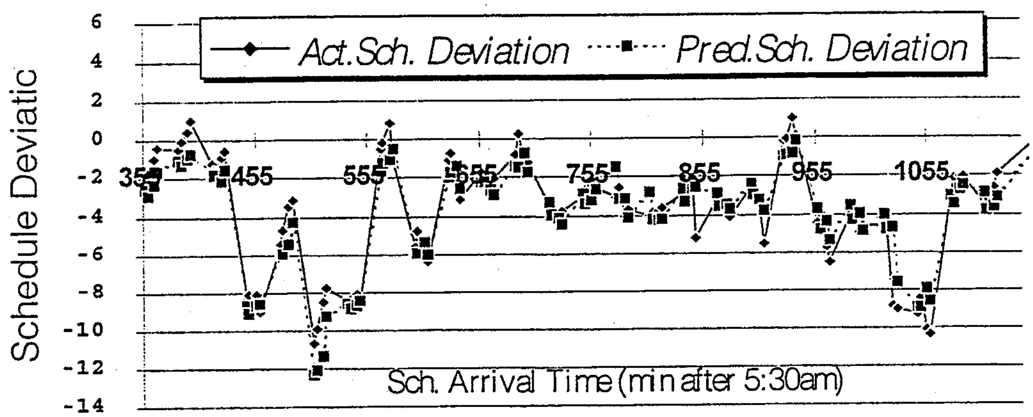


Figure 6.5c : FFN Performance on Test Data : Model IB

6.2.4.3 Model IC

Table 6.3 indicates that the models developed using model structure IC were the most accurate of the five considered. Hence it can be concluded that the model structure IC is the most appropriate structure for modeling the schedule behavior of buses. With an average absolute error of 0.905 minutes (20.55 % average percentage error) the Jordan network model was the most accurate of all the models developed under model set I. Figures 6.6a-6.6c illustrate the performance of the three ANN architectures for this model structure. The Figures 6.6a-6.6c indicate that the Jordan network model performed the best while the Feedforward network performed the next best. Table 6.5 shows that the Jordan model was the preferred model based on the Wilcoxon test. However, the performance of the various models did not differ from each other significantly at the 95 % confidence level.

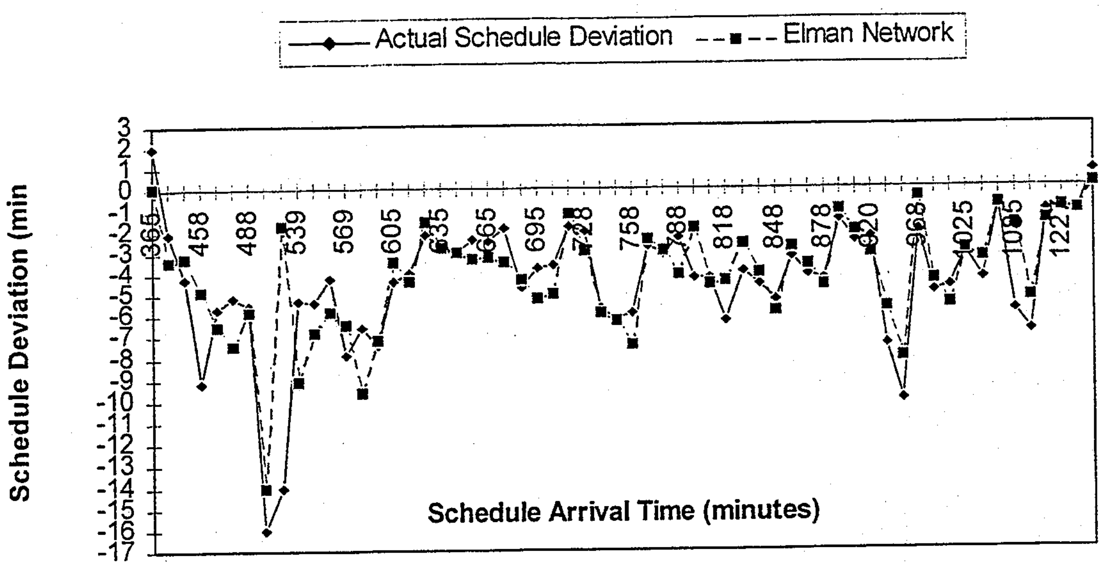


Figure 6.6a : Performance of Elman Network on Test Data : Model IC

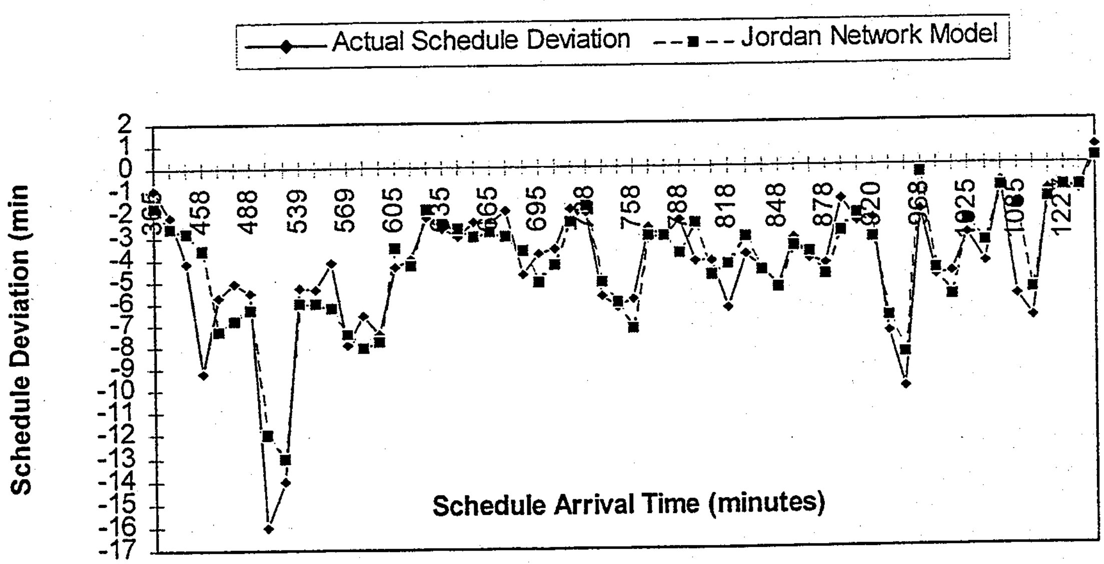


Figure 6.6b : Performance of Jordan Network on Test Data : Model IC

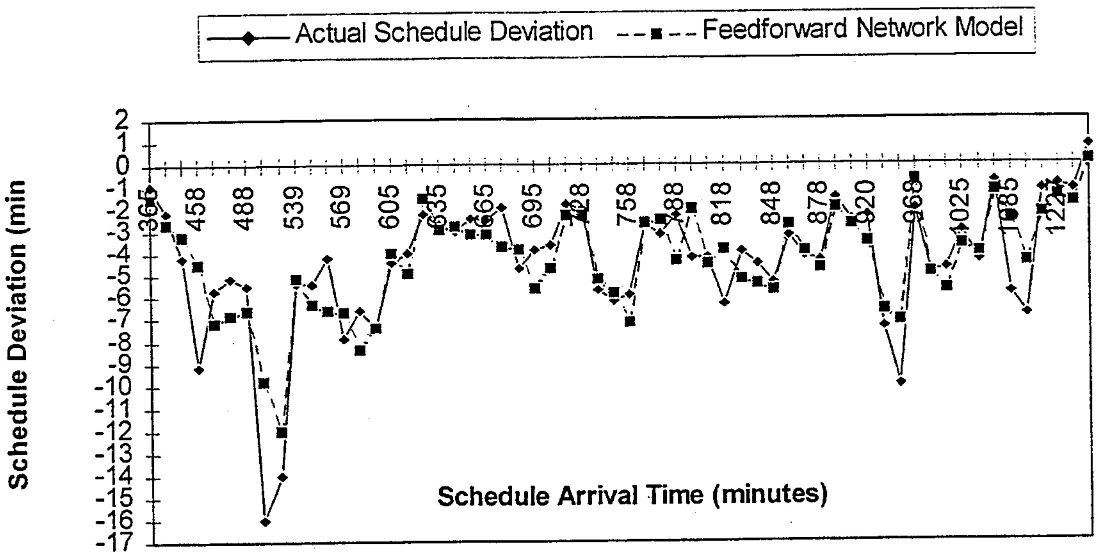


Figure 6.6c : Feedforward Network Performance on Test Data : Model IC

Comparing Figures 6.4a-6.4c, 6.5a-6.5c and 6.6a-6.6c, it can be pointed out that the models under IC outperformed the IA and IB. Table 6.3 shows that the average absolute error for the models using IC structure were significantly less than IB and IA. There was a marked improvement in the accuracy of the models for model structure IC when compared to the IB structure.

As seen in Table 6.4, the tendency of the models to provide erroneous estimates of greater than 3 minutes dropped significantly with model structure IC. For example, the percentage cases for which the Jordan model erred in its estimates by greater than 3 minutes dropped from 5.47% for model structure IB to 2.96% for model structure IC. Hence it can be inferred that the models with structure IC had a significantly lower tendency to show large errors (greater than 3 minutes) in their estimates. This is important because a lower percentage of cases with wrong estimates of greater than 3 minutes will mean that while the models are not the most accurate in terms of average percentage error they can be relied in practical applications to provide results estimates within a reasonable range of the true schedule deviation values. From a practical standpoint, what is important to operational management and schedule planning are schedule deviation values greater than 3 minutes. If the models are not able to estimate such large values of schedule deviation correctly in a significantly greater percentage of the cases than the models cannot be effectively used for practical use as a tool to support the operational management and schedule planning strategies.

6.2.4.4 Model ID

As seen from Table 6.3, the performance of the Elman network was the best using structure ID. The table also indicates that the average percentage error of the Jordan network model increased significantly to 27.52 % for model structure ID from 20.55 % for model structure IC. The performance of the Elman network improved slightly in terms of the average percentage error for model structure ID compared to IC. There was only a insignificant increase in average percentage error for the Feedforward network model when the structure changed from IC to ID.

Table 6.4 suggests that the trend towards lower error in estimates of large schedule deviation values that was seen as the model structures changed from IA to IC was no longer valid. In fact, the percentage of cases with wrong estimates of greater than 3 minute schedule deviations nearly doubled for all the three ANN model architectures. For example, the Elman model using ID experienced errors of greater than 3 minutes in 6.9 % of the evaluation test data cases while the Elman model using structure IC experienced errors of greater than 3 minutes only in 3.7 % of the cases. Based on the Average percentage error and the distribution of error it can be inferred that ID is not an appropriate structure for modeling the schedule behavior of buses.

6.2.4.5 Model IE

Tables 6.3 and 6.4 show that the performance of the three ANN architectures deteriorated significantly when model structure IE was applied. For example, the average percentage error of the Feedforward network model was 51.79 % compared to 22.44 % for the Feedforward network model using structure ID. One plausible reason for such a dramatic drop in the performance can be that the number of training data sets that provided data on schedule deviation information for the previous 5 timepoints was significantly lower. In addition adding more units to the input layer of the ANN can lead to longer training time as well as the potential for larger errors due to increase in number of connections which result in the desire of larger training data.

6.3 Results : Model Set II

Table 6.6 displays the measures of error for the two model structures and for the different ANN architectures. This model set was developed in order to assess the effect of including spatial information on performance of the models. In order to study this effect only two model structures were developed using the data from the case study. The results were compared to the model structures IA -IB which have the same structure besides the timepoint information.

MODEL SET II			
ANN	MEASURE	IIA	IIB
FeedForward Network	Avg. Absolute Error(min)	1.16	1.34
	Avg. % Error	35.01	39.80
Elman Network	Avg. Absolute Error(min)	1.22	1.06
	Avg. % Error	34.8	31.55
Jordan Network	Avg. Absolute Error(min)	1.15	1.08
	Avg. % Error	34.71	32.32

Sample Size = 135

Table 6.6 : Model Set II : Measures of Performance on Test Data Set

6.3.1 Analysis

Table 6.7 shows the results of the Wilcoxon signed-rank tests for comparing the model sets IA with IIA and IB with IIB. This comparison statistical test was used to assess the effect of including the location information (spatial information represented by timepoint on a transit route). As one can note from the table, there was no statistical significance of including the timepoint information. Also by including this extra set of inputs to the network, the training time increased due to increase in complexity of the ANN architecture. Therefore it can be inferred that the inclusion of spatial information in the form of timepoints was not useful in improving the performance of the models. Hence the timepoint information was excluded from the inputs for investigating larger input series lengths (n = 3 to n = 5).

MODEL	**Null Hypothesis**	**Alternative Hypothesis**	**Z - Statistic**	**Significance at 0.05 level?**	**Preferred Model**
FeedForward	$\mu_{IA} - \mu_{IIA} = 0$	$\mu_{IA} - \mu_{IIA} > 0$	-0.5039	No	IA
	$\mu_{IB} - \mu_{IIB} = 0$	$\mu_{IB} - \mu_{IIB} > 0$	-0.6506	No	IB
Elman Net	$\mu_{IA} - \mu_{IIA} = 0$	$\mu_{IA} - \mu_{IIA} > 0$	-1.586	No	IA
	$\mu_{IB} - \mu_{IIB} = 0$	$\mu_{IB} - \mu_{IIB} > 0$	-2.535	Yes	IIB
Jordan Net	$\mu_{IA} - \mu_{IIA} = 0$	$\mu_{IA} - \mu_{IIA} > 0$	-0.2393	No	IA
	$\mu_{IB} - \mu_{IIB} = 0$	$\mu_{IB} - \mu_{IIB} > 0$	-0.635	No	IB

Table 6.7 : Wilcoxon Signed-Rank Tests for Comparing Model IA with IIA and Model IB with IIB.

6.4 Statistical Modeling : Test Results

The exponential smoothing technique was first applied on the data set to see if a simple model can explain the underlying schedule behavior model structure. The exponential smoothing parameter α was close to zero indicating that the exponential smoothing method simply predicts the overall mean and does not use information from the most recent observations. Hence it is of little use and a more sophisticated technique such as the ARIMA method is perhaps a more appropriate modeling technique for the schedule behavior data set.

The results of the ARIMA model development approach described in Chapter 4 are discussed below.

6.4.1 ARIMA Models: Results and Analysis

The ARIMA models were developed for the two model structures described in the previous section. The various steps in the ARIMA model development process and analysis of the results are discussed in the following sub-sections.

6.4.1.1 Identification of ARIMA Model Process

The ARIMA models were developed using the modeling process defined in Chapter 4. The first step in the process involves the identification of the processes underlying the data series. Figure 6.7 illustrates the series plot of the schedule behavior data. The plot shows no evidence of trend or seasonality. The schedule deviation series tends to wander which in other words implies that the short-term mean level is not constant but varies over the course of the series. Hence it can inferred that the series is nonstationary. In order to transform the series to show stationary behavior differencing was necessary. Figure 6.8 illustrates the series after differencing. The plot indicates that the nonstationary behavior of the series has been transformed into a stationary behavior as indicated by the fact that the short term average is always the same.

The next step is to obtain the plots of the ACF and PACF. The ACF and PACF plots show the correlation coefficients and also the 95 % confidence limits (vertical lines on the plot). As illustrated in Figure 6.9, the ACF shows a spike (sudden drop in the values) at lag 2 with a few marginally significant correlations scattered through the rest of the plot. The PACF as shown in Figure 6.10 attenuates rapidly after lag 2. The ACF and PACF plots indicate an AR(2) and MA(2) process. Since the series was differenced preliminary identification of the model can be judged to be ARIMA(2,1,2). However it is important to note that mixed AR and MA models are more complex and identification often requires that alternate ARIMA processes be estimated and diagnosed. In this study this approach of considering alternate ARIMA processes was deemed necessary and therefore was adopted in order to obtain the best fit model. The results and analysis of the model identification are discussed below for both the structures- Model A and Model B. The AIC criteria discussed in Chapter 3 was used to identify the best fit model.

Figure 6.7 : Plot of Schedule Deviation at Timepoint 6, RT23 : Before Differencing

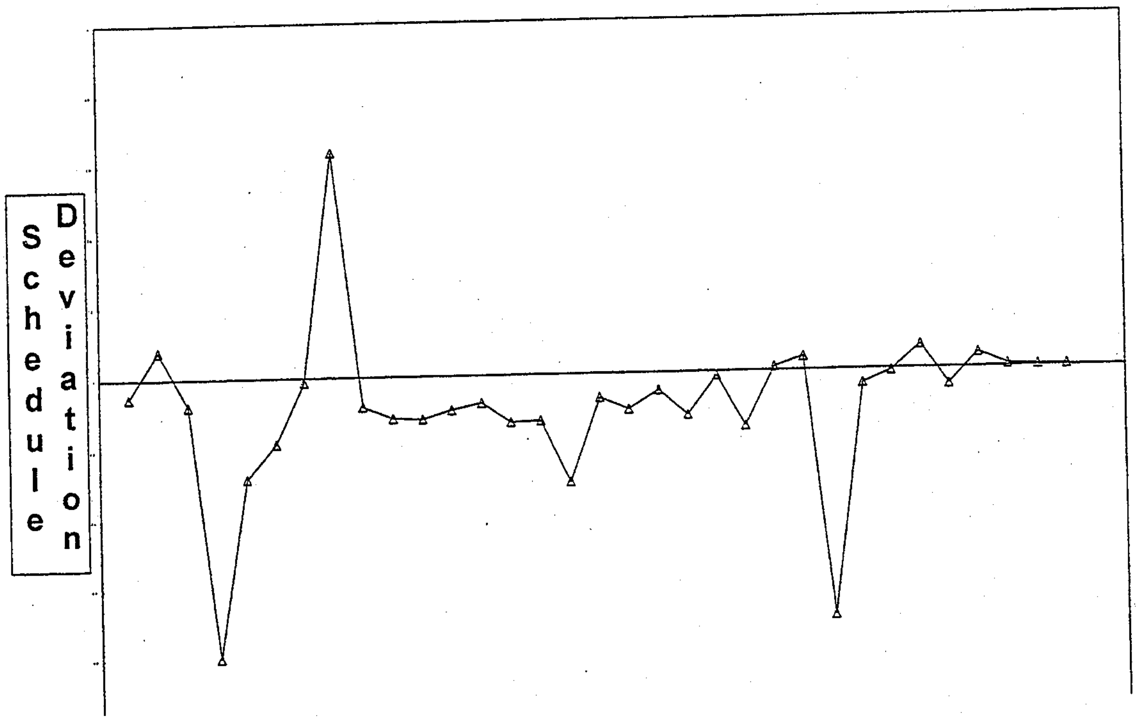
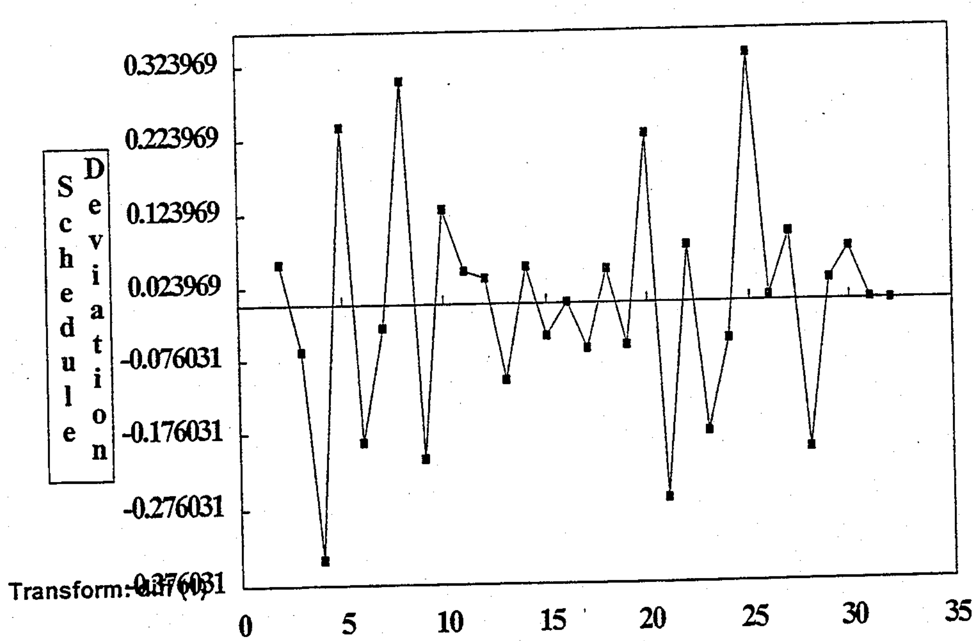


Figure 6.8 : Plot of Schedule Deviation at Timepoint 6, Rt 23 - After Differencing



Note : Normalized Schedule Deviation values

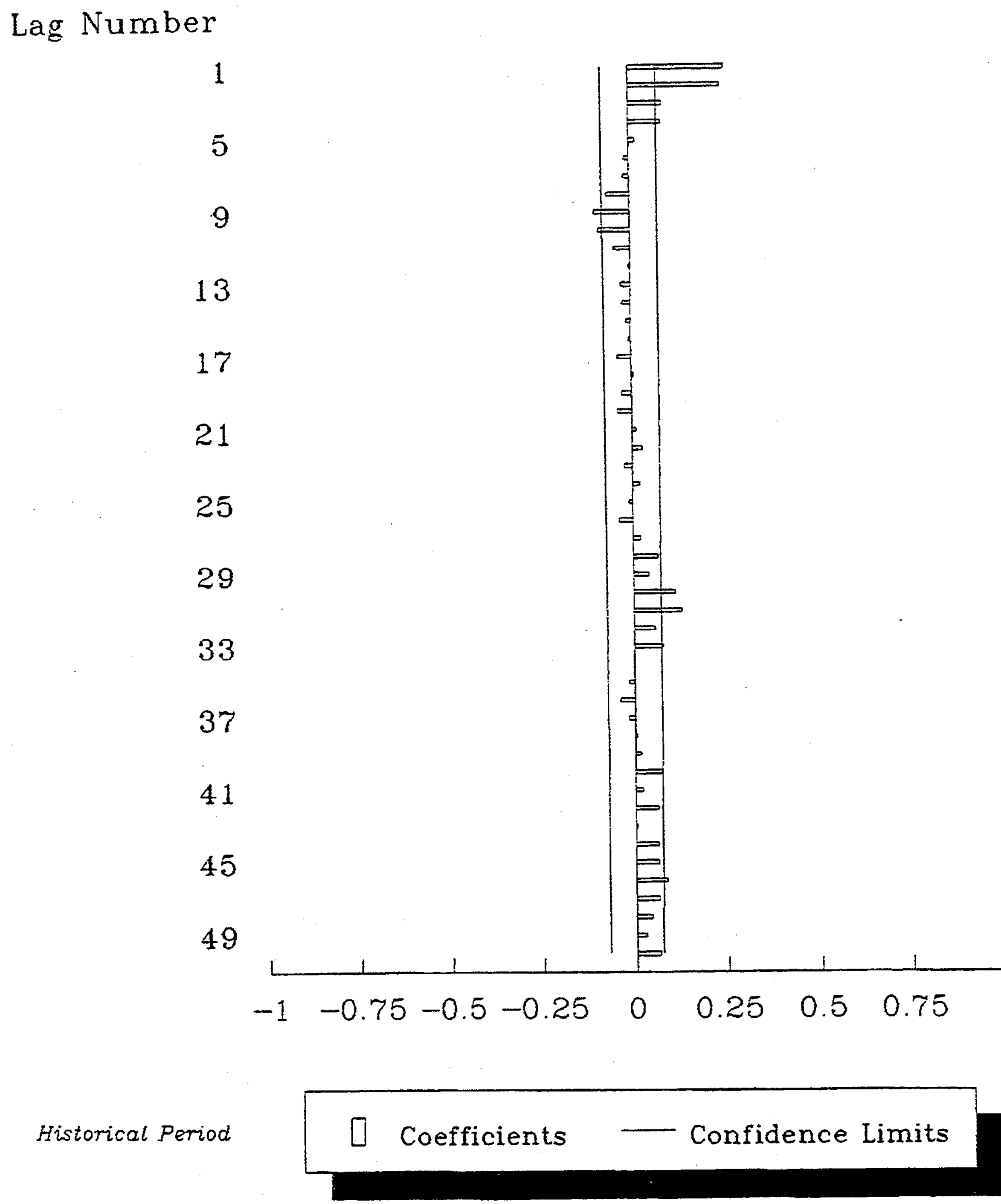


Figure 6.9 : Autocorrelations of Schedule Deviation Series

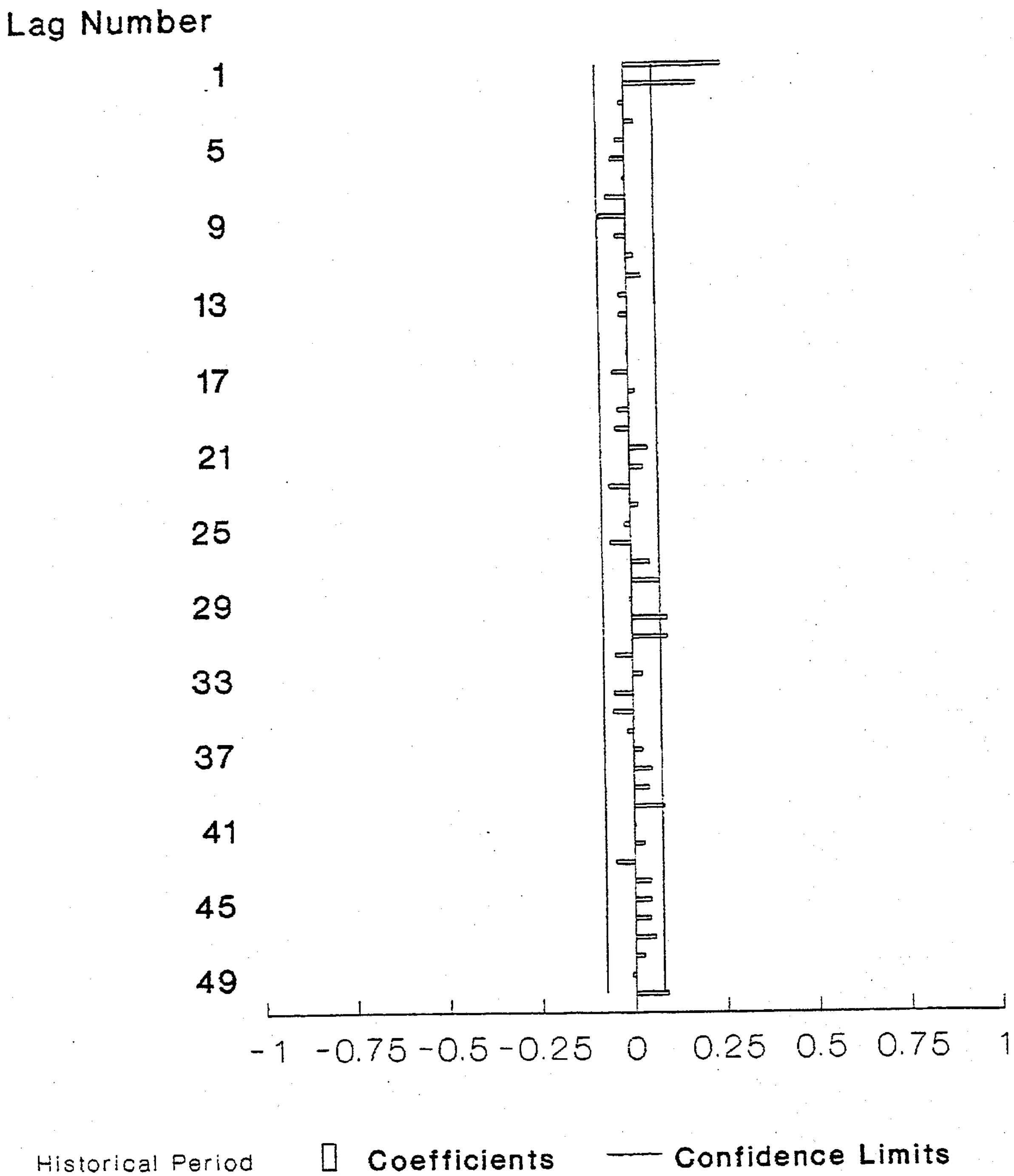


Figure 6.10 : Partial Autocorrelations of the Schedule Deviation Series- SD(k)

Identification of ARIMA process : Model A

Table 6.8 shows the various ARIMA processes investigated for identifying the model process for model structure A. The ARIMA(0,1,1) process produced the best fit as it had the lowest value for AIC. Although the preliminary identification using ACF and PACF indicated an ARIMA(2,1,2) process, Table 6.8 indicates that the ARIMA(0,1,1) process was a better fit due to its lower AIC value. The ARIMA(2,1,2) process was also rejected because the estimates for the AR1 and AR2 parameters for the ARIMA(2,1,2) are not statistically significant at the 95 % confidence limits which leads us to conclude that the autoregression component may not be required for identification of the process.

Identification of ARIMA process : Model B

A summary of the model fits for model identification and diagnosis is given in Table 6.9. for the model structure B. Comparing the AIC values for the different ARIMA processes given in Table 6.9, the ARIMA(2,1,1) model provides the lowest value of AIC and hence can be judged as the best fit model. Although the preliminary identification using ACF and PACF indicated an ARIMA(2,1,2) process, Table 6.8 indicates that the ARIMA(2,1,1) process was a better fit due to its lower AIC value. The ARIMA(2,1,2) process was also rejected because the estimates for the MA2 parameter for the ARIMA(2,1,2) was small (0.38) and not statistically significant at the 95 % confidence limits which leads us to conclude that the MA(1) more closely represents the moving average process rather than MA(2).

Model	Parameters	t-statistic	AIC
(0, 1, 1)	MA1 = 0.994 SD(k-1) = 0.411 SD(k-2) = 0.421 SD(k-3) = 0.002	189.3* 10.64* 10.49* -0.05	-610.49
(1, 1, 1)	AR1 = 0.028 MA1 = 0.999 SD(k-1) = 0.395 SD(k-2) = 0.427 SD(k-3) = 0.0069	0.58 43.05* 10.21* 10.71* 0.176	-605.72
(2, 1, 1)	AR1 = 0.0289 AR2 = -0.011 MA1 = 0.973 SD(k-1) = 0.388 SD(k-2) = 0.425 SD(k-3) = 0.003	0.64 -0.29 72.59* 8.03* 8.68* 0.08	-600.43
(2, 1, 2)	AR1 = 0.2288 AR2 = -0.025 MA1 = 1.205 MA2 = -0.2064 SD(k-1) = 0.398 SD(k-2) = 0.432 SD(k-3) = 0.0054	0.323 -0.6609 1.93* -0.301 10.306* 10.826* 0.1378	-605.02
(0, 1, 2)	MA1 = 0.475 MA2 = 0.174 SD(k-1) = 0.131 SD(k-2) = 0.389 SD(k-3) = 0.086	10.49* 4.47* 3.35* 8.16* 2.22*	-467.18

Table 6.8: Summary of ARIMA Models Fits : Model Structure A

Note : Model Notation is using standard ARIMA convention. ARIMA(p,d,q) where p is the order of autoregression, d is the degree of differencing, and q is the orderof moving average. SSE denotes sum of squares error, AIC denotes Akaike's Information criterion.

* indicates the parameter is significant at the 95 % confidence level.

Model	Parameters	t-statistic	SSE	AIC
(2, 1, 0)	AR1= -0.656 AR2= -0.213 SD(k-1)= 0.377 SD(k-2) = 0.228 SD(k-3)=-0.034 SD(k-4)=-0.021 SD(k-5)=-0.033	-16.08* -5.48* 8.52* 5.84* -0.858 -0.589 -0.893	18.275	-540.18
(2, 1, 1)	AR1= 0.1168 AR2= 0.2032 MA1= 0.984 SD(k-1)= 0.365 SD(k-2)= 0.230 SD(k-3)= -0.019 SD(k-4)= -0.036 SD(k-5)= -0.01	2.798* 5.31* 130.39* 8.27* 5.89* -0.48 -0.99 -0.28	15.18	-665.55
(2, 1, 2)	AR1=0.046 AR2=0.2122 MA1=0.910 MA2=0.072 SD(k-1)=0.365 SD(k-2)=0.2294 SD(k-3)=-0.019 SD(k-4)=-0.038 SD(k-5)=-0.0094	0.24 4.75* 4.67* 0.38 8.28* 5.87* -0.47 -1.07 -0.25	15.21	-663.62

Table 6.9: Summary of ARIMA Models Fits : Model Structure B

Note : Model Notation is using standard ARIMA convention. ARIMA(p,d,q) where p is the order of autoregression, d is the degree of differencing, and q is the order of moving average. SSE denotes sum of squares error, AIC denotes Akaike's Information criterion.

* indicates the parameter is significant at 95 % confidence level.

6.4.1.2 Diagnosis of ARIMA Models

The models were diagonised using the residual series. To check the residuals, plots of the ACF and PACF of the error series was created. The error series plots for the two models A and B are discussed below.

Model A

The error series plots for ARIMA(0,1,1) model,as illustrated in Figures 6.11-6.12, show that the ACF and PACF appear randomly distributed with only a few scattered correlations that exceed the 95 % confidence limits. The 95 % confidence limits are shown as vertical lines. This is also consistent with the null hypothesis that the population autocorrelation function is 0 (indicated by the p value in the ARIMA(p,d,q) representation). The parameters of the model are shown in Table 6.8. The table shows the t-statistic for the various parameters and it indicates that the parameters for the regressors SD(k-2) and SD(k-1) are statistically significant at the 95 % confidence level. It can be inferred that the schedule deviation values at the previous two timepoints (k-1,k-2) have a significant effect on the schedule deviation values at timepoint k. Figure 6.13 illustrates the performance of the ARIMA(0,1,1) model on the test data set. The figures shows that there is a lag of the model specified values to the actual schedule deviation values. This is clearly a distinct weakness of the ARIMA modeling approach.

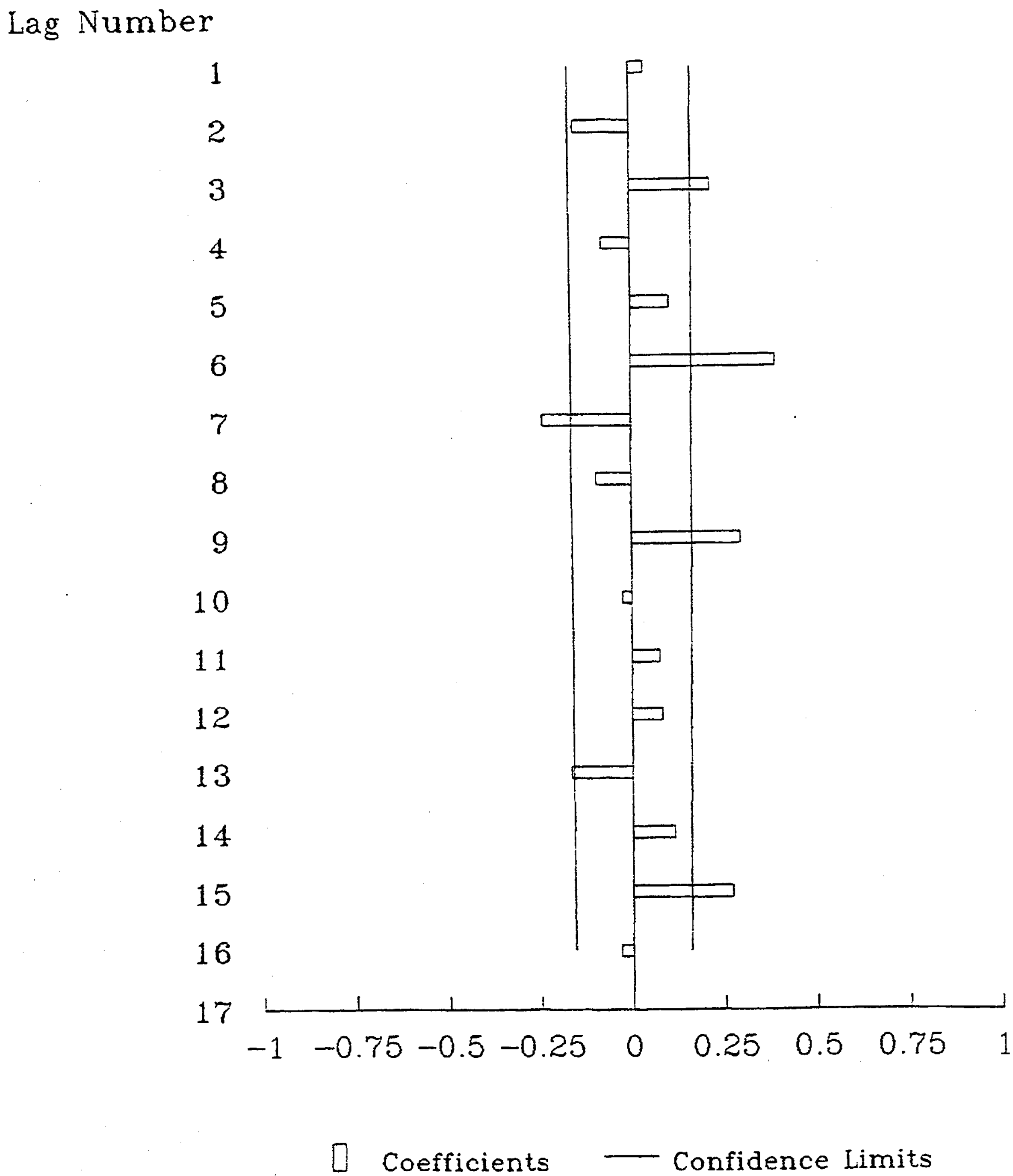


Figure 6.11 : Autocorrelations of the Residual Series ARIMA(0,1,1) Process for Model A

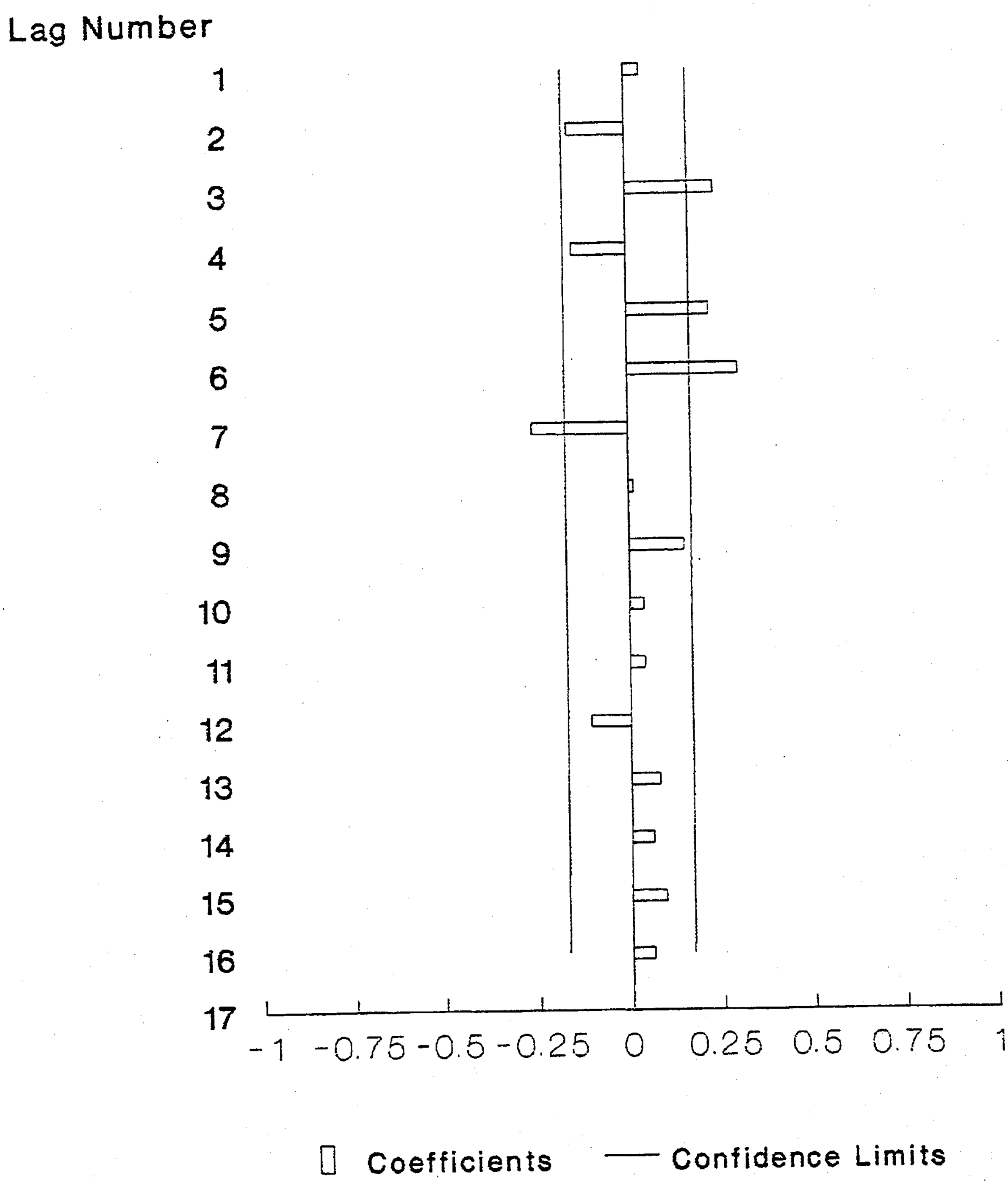
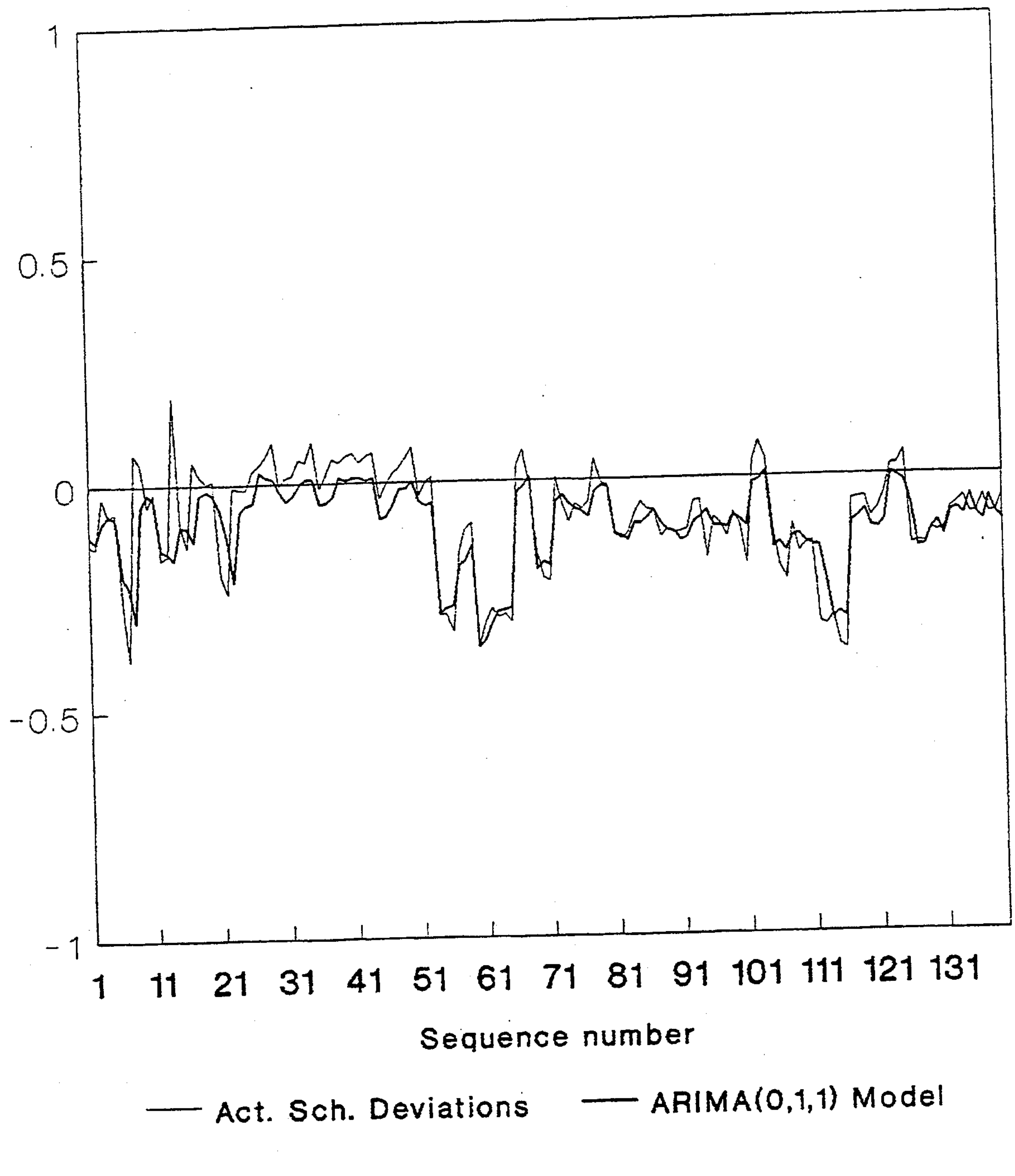


Figure 6.12: Partial Autocorrelations of Residual Series ARIMA (0,1,1) Process for Model A

Figure 6.13 : Performance of the ARIMA(0,1,1) Model A : Test Data



Note : Normalized Schedule
Deviation Values

Model B

The residual plots as illustrated in Figures 6.14-6.15, show that the ACF and PACF appear randomly distributed and no correlation exceeds the 95 % confidence limits. Furthermore, the Box-Ljung statistic for the ACF function is not statistically significant at any lag. From Table 6.9, it is interesting to note that the estimates of parameters corresponding to SD(k-1) and SD(k-2) are significant at the 95 % confidence level with regards to the t-statistic. Thus it can be concluded that the schedule deviation at a timepoint k, is significantly affected by the schedule deviations at the previous two timepoints SD(k-1) and SD(k-2). Thus the ARIMA(2,1,1) model is the best fit model to describe the schedule behavior of buses at a timepoint on a specific route using the model structure B. Figure 6.16 illustrates the performance of the ARIMA(2,1,1) model on the test data set. The ARIMA model exhibits a lag in addition to poorly estimating large schedule deviation values.

Table 6.10 shows the distribution of error for the two ARIMA models. The results suggest that the in the case of model structure B, there was a greater tendency (21.87% of cases) of the ARIMA model to grossly err in its estimate of schedule deviations greater than 3 minutes. It is clear that the ARIMA model using structure B is poorly suited to model the schedule behavior of buses on a route. The ARIMA(0,1,1) model also showed a greater tendency (4.16% of cases) to wrongly estimate schedule deviation values greater than 3 minutes when compared to the values (2.96 % -3.9%) for the ANN models shown in Table 6.4.

Lag	Auto-Corr.	Stand. Err.	-1 -.75 -.5 -.25 0 .25 .5 .75 1	Box-Ljung	Prob.
			-----------------x-+-x-----------------		
1	.002	.038	. * .	.002	.967
2	-.007	.038	. * .	.038	.981
3	-.010	.038	. * .	.114	.990
4	.045	.038	. *.	1.535	.821
5	-.005	.038	. * .	1.555	.907
6	-.019	.038	. * .	1.797	.937
7	.051	.038	. *.	3.651	.819
8	-.044	.038	.* .	5.006	.757
9	-.047	.038	.* .	6.567	.682
10	-.051	.038	.* .	8.391	.591
11	-.003	.038	. * .	8.397	.677
12	.050	.038	. *.	8.190	.599
13	-.061	.038	.* .	12.803	.463
14	.023	.037	. * .	13.191	.512
15	.015	.037	. * .	13.342	.576
16	.005	.037	. * .	13.364	.646

Figure 6.14 : ACF Plot of the Residual Error Series: ARIMA(2,1,1) Model

Note: The . . . vertical lines indicate the 95 % confidence limits. The * indicate the plot values.

Figure 6.15: Partial Autocorrelations (PACF) of Residuals for ARIMA(2,1,1) Process : Model B

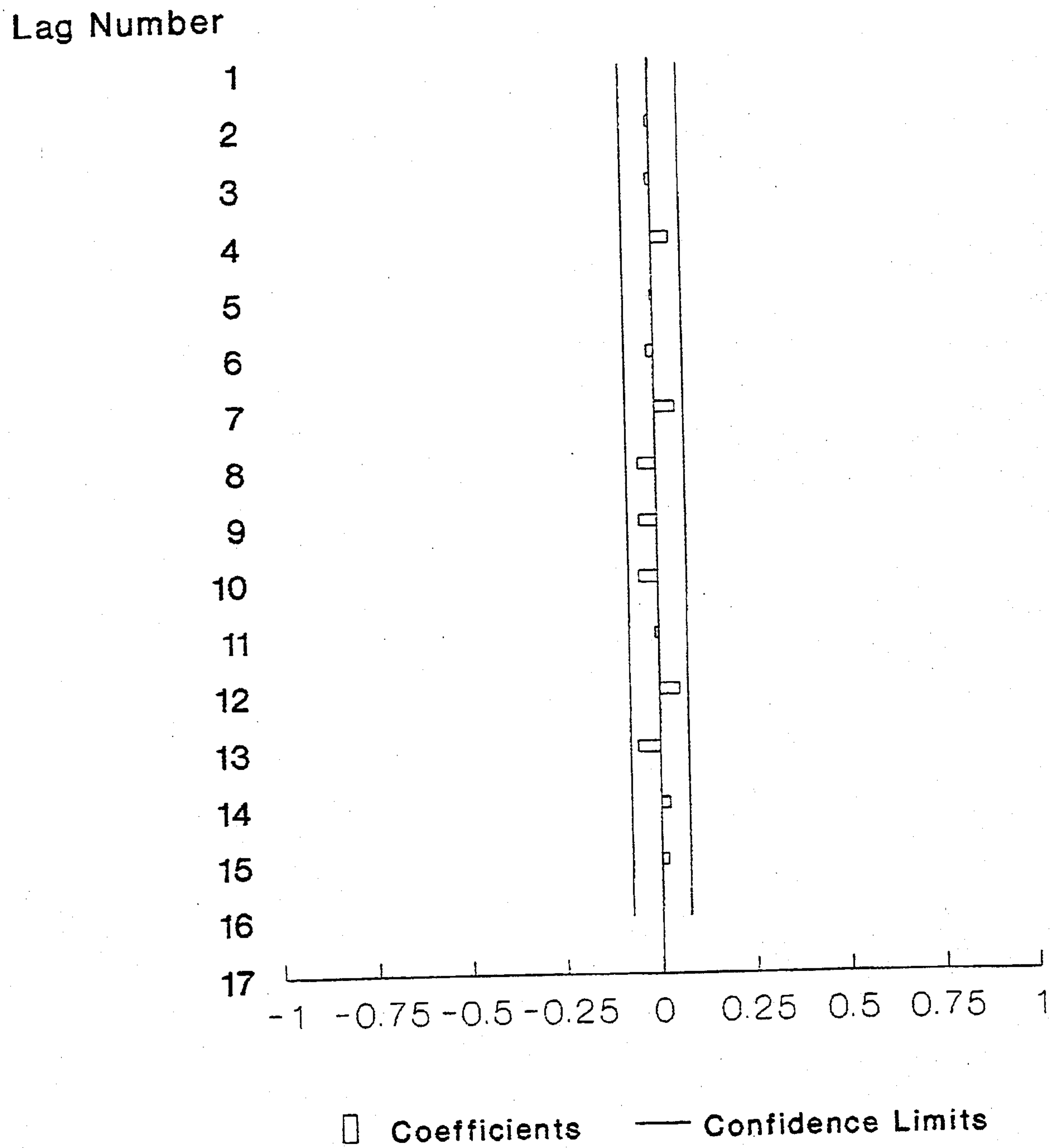
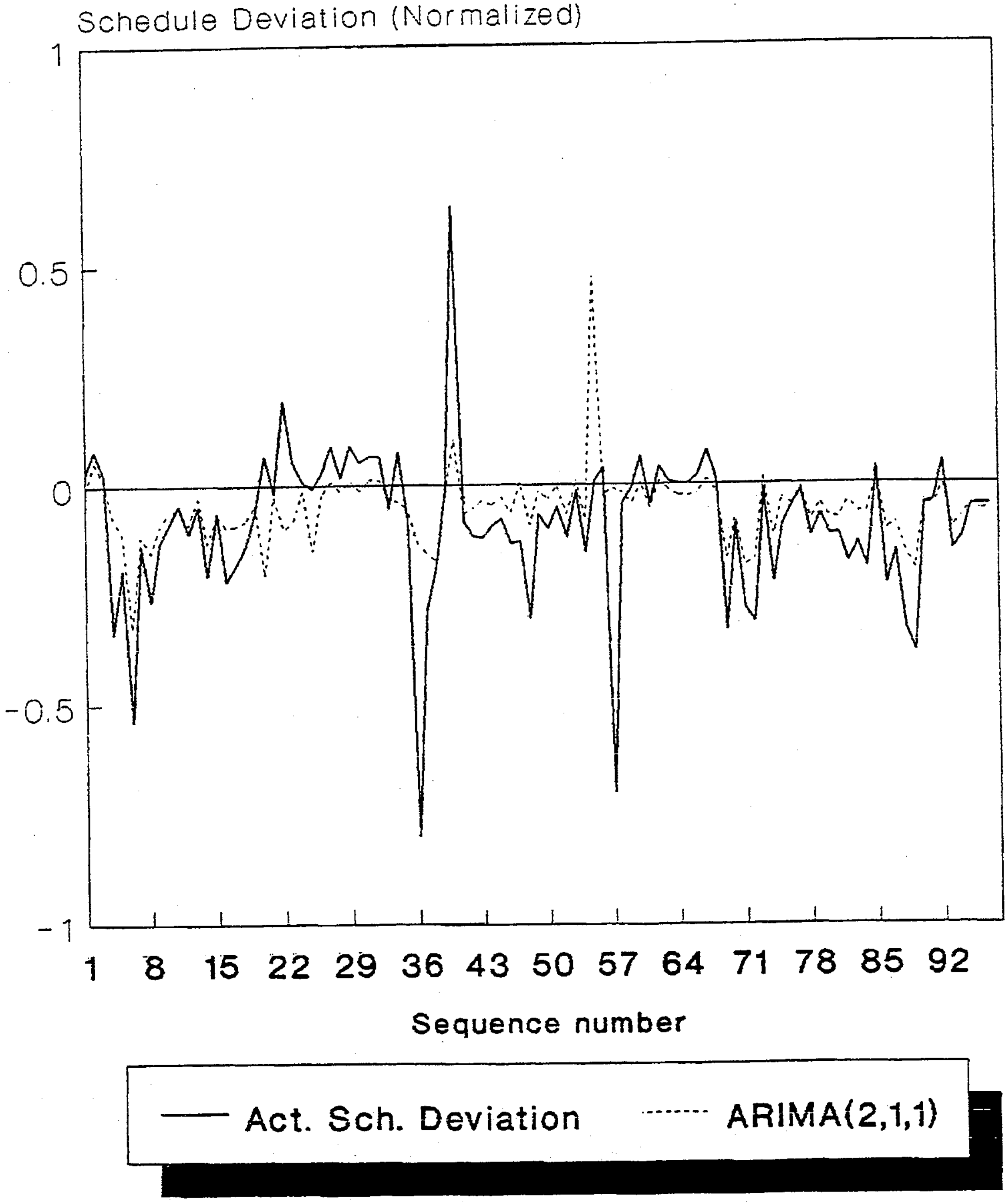


Figure 6.16:Performance of ARIMA(2,1,1) Process : Model B : Test Data Set



		DISTRIBUTION OF ABSOLUTE ERROR % Cases with Absolute Error of			
Model Structure	ARIMA Model	0 - 1 (min)	1 - 2 (min)	2 - 3 (min)	> 3 (min)
A (IC)	ARIMA(0,1,1)	57.29	33.33	5.2	4.16
B (IE)	ARIMA(2,1,1)	43.75	28.12	6.25	21.87

Table 6.10 : Distribution of Absolute Error for ARIMA Models

Note : The model symbols given in the brackets (IC & IE) are the corresponding model structures for the ANN models.

6.5 Comparative Analysis of Models

The various models developed using ANNs and Statistical methods were evaluated based on the measures of error. Table 6.11 summarizes the results of error measures for the various models. It can be seen that the Jordan network model performed the best among all the models for model structure IC. The ARIMA(0,1,1) models' performance based on average absolute error was the worst among the models for model structure IC. All the models performed poorly with respect to model structure IE. The Wilcoxon test results comparing the ARIMA model with the ANN models for model structure IC are shown in Table 6.12. The Z- statistic for the ARIMA-Jordan paired comparison was significant at the 95 % confidence level. The result indicates that the average absolute error for the ARIMA model was significantly greater than the average absolute error for the Jordan model.

		MODEL STRUCTURE	
ANN	MEASURE	IC	IE
FeedForward Network	Average Absolute Error(min) Average % Error	0.966 21.9	1.77 51.79
Elman Network	Average Absolute Error (min) Average % Error	1.006 22.84	2.02 59.12
Jordan Network	Average Absolute Error (min) Average % Error	0.905 20.55	2.28 67.23
ARIMA	Average Absolute Error Average % Error	1.072 24.34	2.28 66.97

Table 6.11 : Comparison of Error Measures for Test Data Set

Null Hypothesis	**Alternative Hypothesis**	**Z - Statistic**	**Significance at 0.05 level?**	**Preferred Model**
$\mu_{BJ} - \mu_{JO} = 0$	$\mu_{BJ} - \mu_{JO} > 0$	-2.47	Yes	Jordan
$\mu_{BJ} - \mu_{EL} = 0$	$\mu_{BJ} - \mu_{EL} > 0$	-1.42	No	Elman

Table 6.12 : Wilcoxon Signed-Rank Tests Comparing the Average Absolute Error of ARIMA Model with ANN Models for Model Structure IC

where, μ_{FF} : Average Absolute Error for Feedforward Network

μ_{JO} : Average Absolute Error for Jordan Network

μ_{EL} : Average Absolute Error for Elman Network.

μ_{BJ} : Average Absolute Error for ARIMA .

Although similar results were obtained for the other paired comparisons, the Z statistic shows that the difference is not significant at the 95 % confidence level.

6.6 Summary

The model development and testing phase of this research effort yielded some interesting results. More importantly, it provided some useful and insightful solutions for realizing the objectives defined in Chapter 1. First, on the basis of the results a potentially suitable structure for the schedule behavior model could be defined. What this means is that the results provided a potentially answer to the basic question of how many upstream timepoints should be included in estimating the schedule deviation at any particular timepoint. The model structure IC proved to be the most appropriate structure for the schedule behavior modeling problem. From the model structure IC, it can be inferred that the schedule deviation values at a timepoint k on the route is influenced by the schedule deviation values at the previous three timepoint locations. Secondly, the results from the ANN modeling experiments showed that ANNs can be successfully trained to provide a reasonably accurate, one timepoint ahead, schedule behavior forecast. The testing and analysis of various modeling techniques did not prove in any way that one technique was better than the other in a statistically significant sense. However, it does illustrate that the Jordan Model holds promise to produce some interesting and reasonable performance. What is interesting and important is the distribution of error, especially average absolute errors of greater than 3 min. The results with regard to the distribution of error indicate that the ANN models performed very well in that they were able to predict large values of schedule deviation much better than the ARIMA modeling approach. Thus it is reasonable to conclude that the ANN models performance with respect to large values of schedule deviation is encouraging. What can be inferred from the results is that the models can be suitable for practical application since their performance with respect to larger values of schedule deviation (greater than 3 minutes) is good. Large values of schedule deviation are the ones that effect operational performance and require the development of schedule control strategies. Hence the ability of the schedule behavior model to estimate these values is of utmost importance. The Jordan model performcd well in this regard and only 2.96 % of cases the errors were greater than 3 minutes. One important observation is that no model was able to outperform the others in a statistically significant sense.

The investigation of schedule behavior modeling using ANNs revealed one important disadvantage of the approach. The accuracy of the ANN models was less than expected perhaps indicating that the amount of training data used was not sufficient. This leads us to one of primary problems of modeling using ANNs. The difficulty of determining in advance how much data is needed. Although it is difficult to estimate *a priori* how much data is needed, it stills forms an important step in developing models using ANNs. One approach is to use an iterative process at developing schedule behavior models using ANNs. First start with a data set that is available and develop the models. If the models perform poorly then collect additional data and increase the size of the data set. The collection of additional data should be possible for transit systems since they are continuously receiving data on the schedule performance using the AVL system. This iterative process can be stopped once a reasonable performance is obtained from the ANN models. One reason such an approach seems rationale is because of the fact that ANNs are data driven models. Hence robust ANN models can be developed only by using adequate data sets.

LEFT INTENTIONALLY BLANK

CHAPTER 7

APPLICATION TO ADVANCED PUBLIC TRANSPORTATION SYSTEMS

7.1 Introduction

The development of a schedule behavior model for a transit route is an important and significant accomplishment. It is intuitively evident from the role of system behavior modeling in many complex physical systems that it can play a key role in the task of scientific operations mangement. However the significance of the schedule behavior model will ultimately depend on the design and development of an intelligent decision support system for efficient and effective bus transit operations management. This chapter will describe a general framework that will illustrate the role and utility of schedule behavior models. In particular, it will show how modeling the schedule behavior of buses will support the transit operations management task.

7.2 Intelligent Transit Management System Architecture

The architecture developed in this research effort is illustrated schematically in Figure 7.1. The figure illustrates the role of the schedule behavior models in the overall system architecture. The output from the schedule behavior model serves as the necessary input to additional modules such as a Knowledge Based System for service control strategy selection, and Advanced Traveler Information System (ATIS) which combine to provide high quality and desired information to the transit operations decision-makers as well as the passenger.

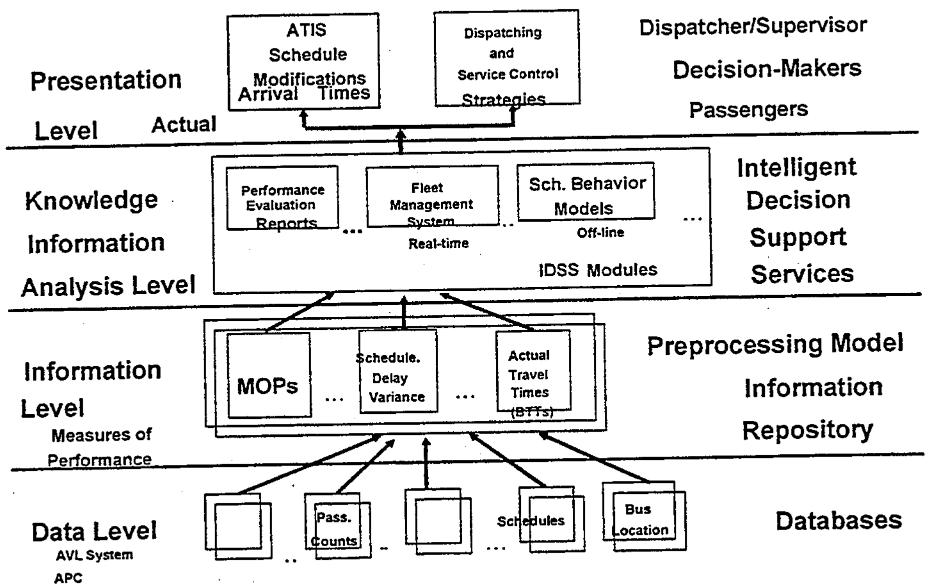


Figure 7.1: Intelligent Transit Management System Architecture

As illustrated in Figure 7.1, the proposed Intelligent Transit Management System has a layered architecture with each layer denoted as a "Level". The Levels have been classified as data, information, knowledge information analysis and decision-making/presentation. A similar architecture was presented by Smith in study on forecasting freeway traffic flow for intelligent transportation systems application [Smith95]. These levels are discussed below.

7.2.1 Data Level

The data level consists of the raw AVL Data collected and stored by the monitoring system. Therefore, this layer consists of the AVL infrastructure and the central database storing the historic AVL data. This level is already present in bus transit systems that have installed an AVL system to monitor the operations.

7.2.2 Information Level

The data obtained using the automatic vehicle location system is preprocessed and stored as information repository in the database. The information repository contains various measures of performance defined by the transit operator and include actual travel times, schedule delay etc. This level is already present to a certain extent in existing systems.

7.2.3 Analysis Level

The analysis level forms the core of the intelligent transit management system. It is the most important layer where information stored in the information repository can be used for analysis usingadvanced information processing techniques such as the schedule behavior modeling. The principal role of this level is to use various techniques to process information stored in the repository and present it as knowledge parameters to the decision level. This level forms the backbone of the intelligent transit management system. Currently only simple processing such as converting information into report formats is being done. This level when developed can assist in the conversion of data into knowledge using automatic processing techniques and ultimately help in operations management. The schedule behavior model can be integrated as an important sub-component of the IDSS.

7.2.4 Decision Level

The decision level can be thought of the layer in the architecture where the results of the analysis level are presented to the decision-making process and appropriate control actions are recommended. The output from the analysis level can be used as input to this decision-making process which can be designed as a knowledge based system for selection of most appropriate service control strategy. The selected strategy obtained from such an autmated decision support process can then be presented to the decision-maker /dispatcher for final implementation. In addition, the modifications to service can be presented to the rider through the passenger information system. At this level, reliance is on human operators (i.e. decision-makers/dispatchers) for final decision making. However, as the system evolves the decision-making process can be automated by developing a suite of software support systems in order to take some load off the transit operators. One example application will be to develop a case-based reasoning system for service control strategy selection. This involves developing a historical database of past cases of service control strategies and the corresponding evaluation of the operations in terms of measures of performance. The intelligent transit management system can then be able to provide automated service control strategy selection without the need for human operators to make decisions. The interaction between various components of the intelligent transit system is discussed in the following section.

7.3 Basic Information Flow Model

The potential benefits of deployment of AVL technology for monitoring of real-time bus operations can be fully accrued only if the following tasks are performed using the AVL information:

- Task 1 : Develop a Basic Intelligent Data Processing Environment: AVL information is intelligently processed for desired operational management task using advanced techniques such as knowledge-based decision support systems and artificial neural networks.
- Task 2 : Develop a Customer Information System that can be integrated into the Advanced
Traveller Information System (ATIS) component of ITS.

The implementation of AVL technology has led to the important problem of information overload on the dispatcher. The information overload problem can be addressed by developing aautomated decision support system that utilize this information and assist the dispatcher in real-time control task. Within this framework of decision support will be an advanced performance evaluation tool that utilizes the real-time location data. The importance of these critical tasks can be illustrated using a AVL information flow model discussed in the next section. This model provides an insight into the level of automation that is needed for intelligent and effective decision-making regarding real-time bus transit operations.

Figure 7.2. illustrates the five component conceptual AVL information flow model

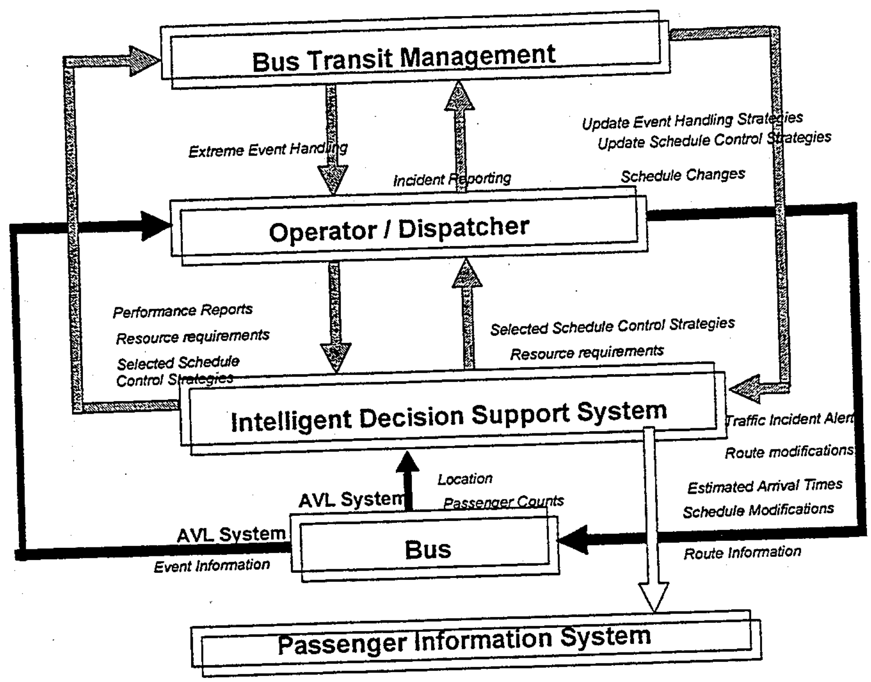


Figure 7.2 : Information Flow Model for Bus Transit Operations

[Kalap92]. The five basic components of this model are :

- Transit Managers
- Operator/Dispatcher
- Intelligent Decision Support System (IDSS)
- Bus (inclusive of the AVL System).
- Passenger Information System (Advanced Traveller Information System)

Transit management is in direct link or information transfer with the operator and the IDSS. The two-way direct transfer helps senior transit management in operational decision making both in real-time and off-line. The real-time link between the transit managers and the operator/dispatcher can aid in real-time management of bus operations. Problems related to operations can be addressed in real-time with the help of information links from the IDSS to the transit management and also the dispatcher. The direct link between the transit management and the dispatcher aids in providing and execution of decisions made by management relating to real-time critical problems associated with operations. The IDSS acts as an intelligent front-end whose information link can assist management make real-time decisions for mitigating real-time problems related to operations. The information link between the bus and (i) IDSS and, (ii) the dispatcher is direct and is through the AVL system. The information is both in the form of data and voice messages. The information link between the bus and the IDSS is one way where in AVL information from the buses on the transit network is transferred to the IDSS for intelligent processing. The one way link suggests that the control of information transfer is not fully automated and critical decision making is with the transit management. The role of the IDSS is therefore of aiding the management in decision making relating to operations, and the IDSS itself does not have the direct control to execute or transfer the problem solving strategies to the buses. The one-way direct link between the IDSS and the passenger information system can provide real-time operational information to the travelling public. This important and critical information link can act as a catalyst for improved public perception and reliability in the bus transit systems.

The information flow model provides a general framework for developing an automated decision support system for bus transit management. The various sub-systems need be designed individually and integrated into an overall intelligent transit management system. In addition the information flow model helps in the identification of the important components of the overall transit management system architecture.

7.4 Potential Application : An Illustrative Example

In section 7.3 an information flow m odel using various advanced public transit system components was discussed. The flow model illustrated a suitable relationship between the monitoring component and other Intellient Transportation System (ITS) related services such as ATIS (Passenger Information System), and an intelligent transit m anagement system for automated operational management. This section will illustrate through an example a pontential application of the bus schedule behavior models within the framework of an intelligent transit management system.

Figure 7.3 shows an illustrative example of the utility of system behavior models to bus transit operations. Service control strategies which essentially involves schedule control, headway control and load control can be designed and developed for real time implementation thus assisting in real-time management of operations. The system behavior models developed can be potentially used for predicting the behavior of the bus transit system as measured in terms of schedule deviation at any time of the day and at any location on the transit route network. Models of schedule behavior can be potentially useful for effective service control especially under time constraints relating to implementation of service control strategies. The prediction models can also be used offline to develop effective schedule adjustment and headway adjustment plans so that the bus transit resources,

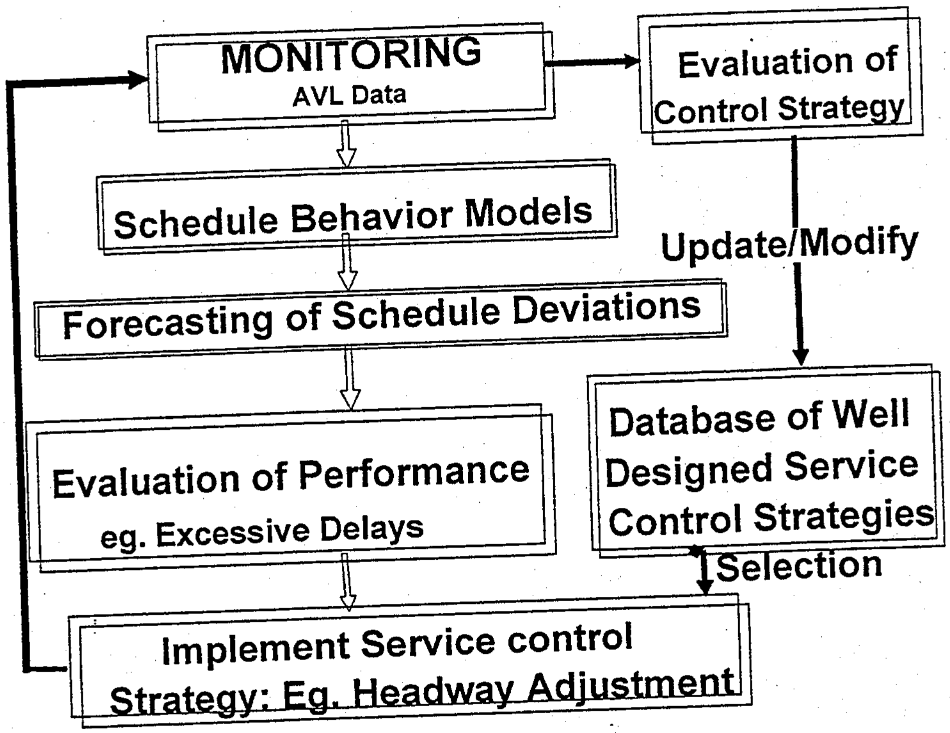


Figure 7.3 : Potential Usefulness of Schedule Behavior Modeling

buses and drivers, are efficiently managed. Since transit service control involves the management of resources such as buses and drivers it is necessary to develop off-line service control strategies that are based on optimal and effective schedule adjustments and headway adjustments so that they can be implemented in real-time conditions. When combined with a knowledge-based decision support system, system behavior models can potentially help in the selection of the best control strategy to minimize delay through efficient and effective schedule control. Changes in operating environments (traffic conditions etc.), unpredictable disturbances such as incidents and failures such as vehicle breakdowns are some of the characteristics that necessitate intelligent control. The best choice may be to achieve adaptive control through the use of ANN models to predict schedule behavior.

System behavior models can be potentially used to improve dispatching control. One important deficiency of current real-time service control practices is that, management has so far not looked into the use of system behavior modeling to optimize real-time dispatching decisions based on the predicted vehicle arrival delays at each bus stop on each route and during different time periods of the day. This lack of predictive capability (also arising from the lack of suitable models of the system behavior) limits management's ability to streamline the real-time decision-making process by making consistent and 'knowledge-based' decisions. The general desire to go in for automation of the man-machine interface of schedule adjustment in order to obtain greater efficiency and also reducing the information load on the dispatcher. In order to achieve greater automation, a system behavior model such as a schedule behavior model can be used as a forecasting tool to help in the design and development of service control strategies that would then be automatically implemented.

7.5 Summary

This chapter described an architecture for developing an intelligent transit mangement system. The role of the schedule behavior model as a subcomponent of the intelligent transit management system was highlighted. The potential utility of the schedule behavior model was enumerated using a simple example. In addition, an information flow model was presented showing the interface between the various components of an overall intelligent transit management system. The flow model assists in understanding of the importance of the various components as well as being useful for the development of the system architecture for bus transit operations.

CHAPTER 8

CONCLUSIONS

This research effort focussed on investigating the feasibility of developing a schedule behavior model. The study attempted to answer the question of how many upstream timepoints schedule deviation information is needed to provide a reasonable estimate for the schedule deviation at a particular timepoint. This was the basis for investigating the different modeling structures. A number of advanced modeling techniques were investigated for developing the schedule behavior models of buses on a particular route of the transit network for the five different modeling structures. While bus transit systems are busy focussing on the implementation of advanced monitoring technologies such as an AVL system, there has been very little or no research on developing automated data modeling techniques for making full use of the large amount of monitoring information provided by the system. The following section highlights the salient contributions of this research to the field of transportation engineering.

8.1 Salient Contributions

The research initiative into the notion of modeling the schedule behavior of buses on a particular route of a transit network has produced some important theoretical as well as practical contributions to the field of transportation engineering. Some of these salient contributions are discussed in this section.

Investigation of the Schedule Behavior Modeling Concept at the Route Level for the First Time

This research has laid the foundation for the application of system behavior modeling concepts to the area of transit operations. The research effort did not provide a robust solution for modeling the schedule behavior of buses. However, this effort provides a basic foundation for the application of system behavior modeling concepts to a dynamic system that for the first time is being monitored in real-time using an AVL system.

There have been no previous attempts to apply system behavior modeling concepts in the area of bus transit systems. By introducing this concept, this research effort as such can be deemed to have made a successful contribution to the body of knowledge in the area of transit operations.

Investigation of the Schedule Behavior Model Structure

The research effort also presented a modeling approach as defined by Equation 4.4 in Chapter 4, that for the first time investigated the effect of the schedule behavior at upstream timepoints on the schedule behavior at any particular timepoint and given time of the day on a route. The modeling approach proposed and investigated in this study presents a theoretical foundation that can be potentially used for future research into the schedule behavior modeling problem using other modeling techniques.

Investigation of Schedule Behavior Modeling using Partial Recurrent Networks

Most research efforts into the application of ANNs in the field of transportation have focussed only on the Feedforward network using the BP algorithm for learning. This research effort has demonstrated that there are other potentially useful architectures such as partial recurrent networks that can be used for developing applications. The ANN modeling approach described here employs simple but powerful architectures. There are several points worth highlighting. The use of partial recurrent network architectures such as the Jordan network and the Elman network combined with the schedule behavior model structures described in Chapter 4, produced some interesting and important lessons. The bus transit schedule behavior problem changed its nature when expressed as sequential events in space and time. The time-varying error signal can be used as a clue to temporal structure. Temporal sequences are not always uniformly structured, nor uniformly predictable. Even when the network has successfully learned about the structure of a sequence, the error may vary. The error signal is a good measure of where structure exists. The representation of time and memory is highly task dependent. The networks depend upon internal representations which have available as part of their input, their own previous state. In this way internal representations intermix the demands of the task with the demands imposed by carrying out that task over time. There is no separate representation of time. There is simply the representation of input patterns in the context of a given output function. It just so happens that those inputs are sequential in space and time. That representation, and thus the representation of time, varies from task to task. The fact that the performance of the Jordan network and the Elman network, presented in Chapter 6, was reasonably good suggests that the schedule behavior modeling approach proposed in this study has contributed positively to the notion that ANN techniques hold promise and that they can be viable tools for schedule behavior modeling. In addition, the results confirm the utility of using networks such as Jordan and Elman networks that embed memory of the previous state in their architectures. However, an important point to note is that the results in no way prove the robustness of the techniques for modeling the behavior of buses. As more research is conducted with larger data sets, it is possible to perhaps arrive at definitive conclusions on the robustness of the various techniques investigated in this study for the purpose of modeling the schedule behavior of buses.

Insight into Potential Strengths and Weaknesses of Artificial Neural Networks as Modeling Techniques.

The results presented in Chapter 6, illustrated some strengths and weaknesses of applying ANNs for modeling the schedule behavior of buses. The strength is that they can reasonably model spatio-temporal sequences that the schedule deviations of a bus along a route represents. This research effort also proved that ANNs can be successfully trained to provide for single timepoint forecasting of schedule deviations.

The single most important weakness as illustrated by the results is the dependence of the models on data sets. During investigation of model development for a real world problem it is often difficult to obtain large data sets. In the case of bus transit operations , the problem is overcome by real-time monitoring that can potentially provide continuous data sets. Since this research was initiated during early part of the implementation of a real-time monitoring system it was difficult to obtain large data sets. Hence the unsatisfactory results can be inferred to be because of this lack of adequate data. The results also illustrated the difficulty of determining in advance how much data is needed to obtain reasonably good ANN models. The implicit modeling approach that an ANN represents presents a practical difficulty in discerning the actual strength of the relationships that are inferred. The other problem illustrated by this research into application of ANNs is the requirement of considerable effort at successfully training a neural network. A number of issues relating to ANN modeling are difficult to resolve because of the lack of sound theoretical knowledge in the ANN literature.

Presentation of Simple ITS Architecture for the Development of Intelligent Transit Management System

The current focus the ITS community is on developing system architectures for various ITS user service groups. The architecture presented in Chapter 7 is an initial attempt at providing a preliminary approach to integrating advanced models with real-time monitoring information to support ITS user services such as public transportation operations management and passenger information systems. The architecture presented illustrates the role of schedule behavior modeling within the framework of an intelligent decision support system for advanced bus transit management.

The lessons learned in this research effort can provide the foundation for a more detailed and comprehensive investigation into the development of a more robust and accurate schedule behavior model.

8.2 Limitations

The research effort had the following limitations.

- The feasibility of schedule behavior modeling was investigated using only one route.
- The problem of schedule behavior modeling focussed on one timepoint ahead prediction. For effective service control strategies to be developed and implemented it would be useful for predicting the schedule behavior at timepoints k+1, k+2 etc.
- The use of Wilcoxon signed test may not be the best approach for performing multiple comparisons. By performing multiple paired comparisons it is likely that the probability of rejecting when the null hypothesis is true increases from 0.05 to 0.15 for three sets of paired comparisons as shown in Table 6.5. One potentially applicable technique is to use the Bonferroni Inequality. The Bonferroni inequality allows for multiple comparisons without introducing any errors that are possible when using the Wilcoxon test. It allows for comparative analysis by not only indicating if any sample performed the best but also provides a statistical rationale that no model can be selected as best. A detailed explanation of the technique for multiple comparisons can be found in [Sheskin97].

8.3 Future Research

The schedule behavior modeling experiments performed in this research effort involved data from only one route. In order for the modeling approaches to become part of a well defined body of knowledge it is imperative that a more detailed study involving the use of data from multiple routes should be attempted. In addition other modeling approaches such as nearest neighbor, inductive techniques should be included in the research effort. In addition multiple data sets are desired for proving the robustness of any applicable technique for modeling the behavior of buses.

The reasons for not very robust results in terms of accuracy can be attributed to the data set. Only a small data set was available due to the fact that during the time the data was collected, TRT's AVL system was still in its initial stages of implementation. Also TRT did not have the mechanism to store the AVL data for more than 14 days. As transit systems gain experience at implementation of more advanced AVL systems, it is possible for attempting a detailed study using a larger data set. Since ANNs are data driven modeling techniques, the lack of adequate training data often leads to poor generalization and hence poor network performance on the test and validation data sets.

Future research should also focus on how to implement the schedule behavior modeling technique into a real world application such as a intelligent transit management system. The architecture described in Chapter 7 should be explored for development and practical implementation in a real world scenario.

This research effort focussed on investigating the schedule behavior of buses using the schedule deviation information as a measure of system performance. Another potential approach is to build models to predict the distance, velocity or travel time of a bus along a link during different times of the day using real-time location information obtained from the AVL system. The event data from the AVL system for each bus gives the location information with respect to time. If we consider a link between two points p1 and p2, then during the bus's journey from point p1 to p2 the AVL data would be collected at intervals dt. Each time t_i has a location p_i associated with it. A distance between the two points can then be computed as d_i. d_i can be termed as a function that measures the distance between two points p_i and p_j. The velocity between the two sampled points can also be computed along with the distance traveled in time t_i. The total distance between point p1 and p2 is known. Therefore the distance, velocity and travel time data are known for each traversal of a bus between the two points during different times of the day. Hence models to predict the these parameters can be attempted using various modeling techniques such as Feedforward network, Jordan network, Elman network, nearest neighbor and ARIMA. This is an alternate method for developing performance models of bus transit systems that can be investigated in the future. The models could be used to predict when a bus would arrive at point p2 given that it was at point p1.

REFERENCES

[Abkowitz83] Abkowitz, M. D., and Engelstein, I. "Factors Affecting Running Time on Transit Routes". In *Transportation Research., 17A* , 1983, pp. 107-113.

[Abkowitz84] Abkowitz, M. D., and Engelstein, I. "Methods for Maintaining Transit Service Regularity". In *Transportation Research Record 961*, TRB, National Research Council, Washington,D.C.., 1984, pp. 1-8.

[Bacha92] Bacha, H., and Meyer, W. "A Neural Network Architecture for Load Forecasting". In *Proceedings of International Joint Conference on Neural Networks* : Baltimore, MD, Vol II, 1992, pp. 442-447.

[Bates86] Bates, J. W. *Transportation Research Circular No. 300 : Definitions of Practices for Bus Transit On-Time Performance: Preliminary Study.* Transportation Research Board, National Research Council, Washington, D.C., February, 1986.

[Beale92] Beale, M., and Demuth, H. "Preparing Controllers for Nonlinear Systems". In *AI Expert*, July, 1992, pp.42-47.

[Bhat90] Bhat, N. and McAvoy, T.J. "Use of Neural Nets for Dynamic Modeling and Control of Chemical Process Systems ". *Computers in Chemical Engineering*, Vol.14, No.5,pp. 573-583, 1990.

[Bhat92] Bhat, N. and McAvoy, T.J. " Determining Model Structure for Neural Models by Network Stripping". *Computers in Chemical*

Engineering,Vol.16,No.4, pp. 271-282, 1992.

[Box70] Box, George E. P. and Jenkins, Gwilym M. " *Time Series Analysis : Forecasting and Control*". Holden-Day,Inc., San Francisco, California, 1970.

[Breiman73] Breiman, Leo. " *Statistics: With a View Toward Applications*". Houghton Mifflin Company, U.S., 1973.

[Bulsari] Bulsari, A. B., and Saxen, Henrik. "A Recurrent Neural Network for Time-Series Modelling".

[Burke93] Burke, Laura. "Assessing a Neural Network : Validation Procedures". *PC AI*, March/April 1993, pp 20-24.

[Caire92] Caire, P., Hatabian, G., and Muller, C. "Progress in Forecasting by Neural Networks". In *Proceedings of International Joint Conference on Neural Networks* : Baltimore, MD, Vol II, 1992, pp. 540-545.

[Casey91] Robert F Casey, Lawrence N. Labell et.al. *Advanced Public Transportation Systems: The State of the Art.* Final Report, DOT-VNTSC-UMTA-91-2, U.S.DOT, Urban Mass Transportation Administration, National Research Council, Washington,D.C, 1991.

[Casey96] Robert F Casey, Lawrence N. Labell et.al. *Advanced Public Transportation Systems: The State of the Art. Update '96*, DOT-VNTSC-FTA-95-13, U.S.DOT, Federal Transit Administration, National Research Council, Washington,D.C, January 1996.

[Castle91] Castle Rock Consultants. *Assessment of Advanced Technologies for Transit and Rideshare Applications*. Final Report, NCTRP Project 60-1A, Washington,D.C, July 1991.

[Caudill93] Caudill, Maureen. " Putting Time in a Bottle". *AI Expert*, Vol 8, Number 2, February, 1993, pp. 22-27.

[CChEng92] " Neural Network Applications in Chemical Engineering". *Computers in Chemical Engineering*, Vol.16, No.4, pp. 271-282, 1992.

[Chakra92] Chakraborty, K., Mehrotra, K., Mohan, C. K., and Ranka, S. "Forecasting the Behavior of Multivariate Time Series Using Neural Networks". In Neural Networks, Vol. 5, No. 6, 1992, pp. 961-970.

[Cheu91] Cheu, Ruey L., Ritchie, Stephen G., Recker, Wilfred W., and Bavarian, Behnam. "Investigation of a Neural Network Model for Freeway Incident Detection". *The Second International Conference on the Application of Artificial Intelligence Techniques to Civil and Structural Engineering*, Oxford, England, 3-5 September, 1991.

[Chitra92] Chitra, Surya P. " Neural Net Applications in Chemical Engineering". *AI Expert*, Vol.7, Number 7, November, 1992, pp. 20-25.

[Connor94] Connor, J.T., Douglas, M.R., and Atlas, L.E., "Recurrent Neural Networks and Robust Time Series Prediction". In *IEEE Transactions on Neural Networks, Vol.5,* March, 1994, pp. 240-254.

[Dough93] Dougherty, Mark S. and Kirby, Howard R. " The Use of Neural Networks To Recognise and Predict Traffic Congestion". *Traffic Engineering and Control*, pp. 311-314, June 1993.

[Elman88] Elman, J.L. "Finding Structure in Time". Technical Report 8801, University of California, San Diego, Center for Research in Language, 1988.

[Etschmair] Etschmair,M. M."Operational Planning and Control in Transportation Systems ". Proceedings of the *5th IFAC/IFIP/IFORS Conference*, Vienna, Austria, 8-11 July 1986, pp. 147-153.

[Faghri92] Faghri, Ardeshir and Hua, J. " Evaluation of Applications of Neural Networks in Transportation Engineering". *71nd Annual Meeting, Transportation Research Board*, Washington,D.C., January, 1992.

[Faghri96] Faghri, Ardeshir and Aneja, Sandeep. " Artificial Neural Network-based Approach To Modeling Trip Production". In *Transportation Research Record 1556*, TRB, National Research Council, Washington,D.C.., 1996, pp. 131-136.

[Fahlman88] Fahlman, S.E. "Faster-learning Variations on Backpropagation: An Empirical Study". In T.J. Sejnowski, G.E. Hinton and D.S. Touretzky, editors, *1988 Connectionist Models Summer School*, San Mateo, CA, Morgan Kaufmann, 1988.

[Foster92] Foster, W.R., Collopy, F.,and Ungar, L.H. " Neural Network Forecasting of Short, Noisy Time Series". *Computers and Chemical Engineering*, Vol 16, No 4, pp. 293-297, 1992.

[Gilmore95] Gilmore, F. John, and Abe, Naohiko. "Neural Network Models for Traffic Control amd Congestion Prediction". In *IVHS Journal*, 1995, Vol.2(3), pp.231-252.

[Ginz92] Ginzberg, I., and Horn, D. Learning the Rules of a Time Series. In *International Journal of Neural Systems*, Vol. 3, No. 2,, 1992, pp. 167-177.

[Guo92] Guo, Zhichao and Uhrig, E. Robert. " Use of Artificial Neural Networks to Analyze Nuclear Power Plant Performance". *Nuclear Technology*, Vol. 99, July, 1992 , pp.36-41.

[Grossberg] Grossberg, S. ed., " *Neural Networks and Natural Intelligence*". MIT Press, Cambridge, Massachussetts, 1988.

[Guenthner] Guenthner, Richard P. and Hamat, Kasimin. "Distribution of Bus Transit On-time Performance". In *Transportation Research Record 1202*, TRB, National Research Council, Washington,D.C.., 1988, pp. 1-8.

[Hallowell] Hallowell F. Susan and Morlok K. Edward. "*Carrier Cost Savings From Advanced Vehicle Monitoring and Telecommunication Systems in Intercity Irregular Route Trucking*". Report, Department of Systems, University of Pennsylvania, Philadelphia, PA, 1990.

[Hecht90] Hecht-Nielsen, Robert. " *Neurocomputing*". Reading, Massachussetts, Addison-Wesley, 1990.

[Hernandez] Hernandez, E., and Arkun, Y. " Study of the Control-Relevant Properties of Backpropagation Neural Network Nodels of Nonlinear Dynamical Systems ". *Computers in Chemical Engineering*, Vol.16, No.4, pp. 227-240, 1992.

[Hertz91] Hertz, J., A. Krogh, and R.G. Palmer. *Introduction to the Theory of Neural Computation*. Addison-Wesley Publishing Company, 1991.

[Hoekstra91] Hoekstra, J., and Kooijman, R. "Recurrence with Delayed Links in Multilayer Networks for Processing Sequential Data". In *Artificial Neural Networks*; T. Kohonen et al. (editors). Elsevier Science Publishers B.V. 1991, pp. 1149-1152.

[Hopfield] Hopfield, J. J., and Tank, W. "Computing with Neural Circuits: A Model". *Science* 1986, 233, pp. 625-633.

[Hoptroff] Hoptroff, R.G. "The Principles and Practice of Time Series Forecasting and Business Modelling Using Neural Nets". In *Neural Computing and Applications, Vol 1,* 1993, pp. 59-66.

[Hua93] Hua, Jiuyi and Faghri, Ardeshir. " Dynamic Traffic Pattern Classification Using Artificial Neural Networks". *Transportation Research Board, 72nd Annual Meeting*, January 10-14, 1993, Washington, D.C.

[Iso90] Iso, K. and Watanabe, T. " Speaker-independent Word Recognition Using A Neural Prediction Model". *Proceedings of ICASSP*, Albuquerque, NM, April, 1990.

[Jordan86] Jordan, Michael I. Attractor Dynamics and Parallelism in a Connectionist Sequential Machine. *Proceedings of the Eight Annual Conference of the Cognitive Science Society*, Hillsdale, NJ, 1986, pp. 531-546.

[Joseph92] Joseph, B., Wang, F.H., and Shen, D.S-S. " Exploratory Data Analysis : A Comparison of Statistical Methods with Artificial Neural Networks". *Computers and Chemical Engineering*, Vol 16, No 4, pp. 413-423, 1992.

[Kalap92] Kalaputapu, R., and Demetsky, M. J. "Intelligent Decision Support Systems for Operational Planning and Management of Bus Transit Services". In *Proceedings of the National Conference on Advanced Technologies in Public Transportation*, TRB, San Francisco, California, Aug. 16-19, 1992.

[Kalap94] Kalaputapu, R., and Demetsky, M. J. " Application of Artifical Neural Networks and Automatic Vehicle Location Data for Bus Transit Schedule Behavior Modeling". In *Proceedings of IVHS America Fourth Annual Meeting*, Atlanta, GA, April 17-20, 1994.

[Kalap95] Kalaputapu, R., and Demetsky, M. J. "Modeling Bus Transit Schedule Deviations Using Automatic Vehicle Location Data and Artificial Neural Networks".In *Transportation Research Record 1497*, TRB, National Research Council, Washington,D.C., 1995, pp. 44-52.

[Kalaputapu] Kalaputapu, R., and Demetsky, M. J. "Modeling Schedule Behavior of Buses". *Proceedings of the 1995 American Society of Civil Engineers*

Transportation Congress,Volume I, pp 257-267. San Diego, October 22-26, 1995.

[Kamijo90] Kamijo, K., and Tanigawa, T. "Stock Price Pattern Recognition: A Recurrent Neural Network Approach". In *Proceedings of the IEEE International Joint Conference on Neural Networks,* I215-I221, San Diego, CA, 1990.

[Kean93] Kean, John. " Neural Nets and Stocks : Training a Predictive System ". *PC AI*, Vol. 7, No. 5, September-October, 1993, pp. 45-47.

[Kimoto90] Kimoto, T., Asakawa, K., Yoda, M., and Takeoka, M. "Stock Market Prediction System with Modular Neural Networks". In *Proceedings of the IEEE International Joint Conference on Neural Networks,* I1-I6, San Diego, CA, 1990.

[Kirby94] Kirby, H.R., Dougherty, M.S., and Watson, S.M. " Should We Use Neural Networks or Statistical Models for Short term Motorway Traffic Forecasting?". Presented at *73rd TRB Annual Meeting,* Washington,D.C, January, 1994.

[Kohonen84] Kohonen, T. "*Self-Organization and Associative Memory*". Springer-Velag, New York, 1984.

[Kumar91] Kumar, V. V., Krishnamurthy, A. K., and Ahalt, S. C. "Phonetic-to-Acoustic and Acoustic-to-Phonetic Mapping Using Recurrent Neural Networks". In *SPIE*, Vol. 1469 : Applications of Artificial Neural Networks II, 1991, pp.484-490.

[Lapedes87] Lapedes, A. and Farber, R. " *Nonlinear Signal Processing Using Neural Networks: Prediction and System Modeling* ". Technical Report. LA/UR 87/2662, Los Alamos National Lab., Los Alamos, NM, 1987.

[LeCun88] Le Cun, Y. "Generalization and Network Design Strategies". In Pfeifer, Schreteer, Fogelman, and Steels, editors, Connectionism in Perspective, 1988, Elsevier, Zurich, Switzerland.

[Levin90] Levin, E. "Word Recognition Using Hidden Control Neural Architectures". *Proceedings of ICASSP*, Albuquerque, NM, April, 1990.

[Lippmann] Lippmann P. Richard. "An Introduction to Computing with Neural Nets". *IEEE Acoustics, Speech, and Signal Processing Magazine*, April 1987, pp 4-22.

[Makri83] Makridakis, S., Wheelwright, S. C., and McGee, V. E. "*Forecasting: Methods and Applications*. Wiley: New York, 1983.

[McClus93] McCluskey, Peter C. "Feeforward and Recurrent Neural Networks and Genetic Programs for Stock Market and Time Series Forecasting". M.S. Thesis, 1993, Department of Computer Science, Brown University.

[Mehra92] Mehra Pankaj and Wah W. Benjamin (Editors). " *Artificial Neural Networks : Concepts and Theory*". IEEE Computer Society Press Tutorial, 1992.

[Morlok91] Morlok K. Edward, Bruun C. Eric and Battle J. Kimberly. " *Advanced Vehicle Monitoring and Communication Systems for Bus Transit: Benefits* *and Economic Feasibility*". Executive Summary, Urban Mass Transportation Administration, U.S. Department of Transportation, Washington,D.C., September 1991.

[Mozer92] Mozer, Michael C. "Neural Net Architectures for Temporal Sequence Processing". In *Time Series Prediction : Forecasting the Future and Understanding the Past.* Proceedings of the NATO Advanced Research Workshop on Comparative Time Series Analysis, Santa Fe, New Mexico, U.S.A, May 14-17, 1992.

[Narendra] Narendra, K.S., and Parthasarathy, K. "Identification and Control of Dynamic System Using Neural Networks". *IEEE Transactions on Neural Networks*, Vol.1, No.1, 1990, pp. 4-27.

[Nasci94] Nascimento Jr., C. L. "*Artificial Neural Networks in Control and Optimization*". Ph.D Thesis, Control Systems Center, University of Manchester, U.K, Feb. 1994.

[NCTRP91] Supervision Strategies for Improved Reliability of Bus Routes. *National Cooperative Transit Research & Development Program: Synthesis of Transit Practice 15,* National Research Council, Washington, D.C., September, 1991.

[Park91] Park, D.C., El-Sharkawi, M.A., Marks II, R.J., Atlas, L.E., and Damborg, M.J. "Electrical Load Forecasting Using An Artifical Neural Network". In IEEE Transactions on Power Systems, Vol. 6, No.2, May 1991, pp. 442-448.

[Pearl90] Pearlmutter, B. A. "*Dynamic Recurrent Neural Networks* ". Technical Report

CMU-CS-90-196, Carnegie Mellon University, Pittsburgh, PA, 1990.

[Pineda89] Pineda, F. J. "Recurrent Backpropagation and the Dynamical Approach to Adaptive Neural Computation". *Neural Computing*, 1989, Vol. 1, pp. 167-172.

[Pollard92] Pollard, J.F., Broussard, M.R., Garrison, D.B., and San, K.Y. " Process Identification Using Neural Networks". *Computers and Chemical Engineering*, Vol 16, No 4, pp. 253-270, 1992.

[Prechelt94] Prechelt, Lutz. "*A Study of Experimental Evaluations of Neural Network Learning Algorithms: Current Research Practice*". Technical Report 19/94, University of Karlsruhe, Karlshrue, Germany, August, 1994.

[Rathburn93] Rathburn, T. " *Advanced Neural Computing* ". NeuralWare, Inc. Pittsburgh, PA, 1993.

[Refenes93] Refenes, A.N., Azema-Barac, M., Chen, L., and Karoussos, S.A. "Currency Exchange Rate Prediction and Neural Network Design Strategies". In *Neural Computing and Applications, Vol 1,* 1993, pp. 46-58.

[Rehbein92] Rehbein, A.D., Maze, M. S., and Havener, P.J. " The Application of Neural Networks in the Process Industry". *ISA Transactions*, Vol.31, No.4,1992, pp. 7-13.

[Reins91] Reinschmidt, F. Kenneth. " Neural Networks: Next Step for Simulation and Control". *Power Engineering*, November, 1991, pp. 41-45.

[Rumel90] Weigend, Andreas S., Huberman, Bernardo A., and Rumelhart, David E. " Predicting the Future : A Connectionist Approach". *International Journal of Neural Systems*, Vol.1, No.3 1990, pp. 193-209.

[Senevi90] Seneviratne, P. N. "Analysis of On-Time Performance of Bus Services Using Simulation". *Journal of Transportation Engineering, Vol 116, No.4*, Jul/Aug. 1990, pp 517-531.

[Servan88] Servan-Schreiber, D., Cleeremans, A., and McCelland, J. L. "*Encoding Sequential Structure in Simple Recurrent Networks*". Carnegie Mellon University,Technical Report CMU-CS-88-183, 1988.

[Sharda] Sharda, R., and Patil, R.B. "A Connectionist Approach to Time Series Prediction: An Empirical Test". In *Neural Networks in Finance and Investing*: Trippi, R.R. and Turban, E. (Editors), Probus Publishing Company, Chicago, IL, 1994.

[Shekar92] Shekar, Shashi, Amin, B.Minesh, and Khandelwal, Prashant. " Generalization Performance of Feed-Forward Neural Networks". *Neural Networks: Advanced and Applications, 2,* E. Gelenbe (Editor), Elsevier Science Publishers B.V. 1992.

[Sheskin97] Sheskin David J. *"Handbook of Parametric and Nonparametric Statistical Procedures"*.CRC Press, 1997.

[Simpson90] Simpson, P. K. *Artificial Neural Systems - Foundations, Paradigms, Applications, and Implementations*, Pergamon Press, New York, U.S.A., 1990.

[Smith95] Smith, Brian. “*Forecasting Freeway Traffic Flow For Intelligent Transportation Systems Application*”. Ph.D Dissertation, University of Virginia, May 1995.

[SNNS3.2] SNNSv3.2. Stuttgart Neural Network Simulator: Version 3.2, User's Manual, University of Stuttgart, Germany, 1994.

[Strathman] Strathman, James G., and Hopper, Janet R. "Empirical Analysis of Bus Transit On-time Performance" In *Transportation Research, Part A, Policy and Practice 27A*, March 1993, pp. 93-100.

[Su92] Su, Hong-Te , McAvoy, J. Thomas and Werbos, Paul. " Long-Term Predictions of Chemical Processes Using Recurrent Neural Networks: A Parallel Training Approach". *Industrial Engineer and Chemical Research*,1992, Vol 31, pp 1338-1352.

[Tally87] Tally, Wayne K., and Becker, Jeff. "On-Time Performance and the Exponential Probability Distribution". In *Transportation Research Record 1108*, TRB, National Research Council, Washington,D.C.., 1987, pp. 22-26.

[Tang91] 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.

[Tebel90] Tebelskis, J. and Waibel, A. "Large Vocabulary Recognition Using Linked

Predictive Neural Networks". *Proceedings of ICASSP*, Albuquerque, NM, April, 1990.

[Terry] Terry, W. R., J. B. Lee, and A. Kumar. 1986. "Time Series Analysis in Acid Rain Modeling: Evaluation of Filling Missing Values by Linear Interpolation." *Atmospheric Environment*. pp. 1941-1945.

[Trippi93] Trippi, R.R., and Turban, E. *Neural Networks in Finance and Investing.* Probus Publishing Company, Chicago, IL, 1993.

[Ulbricht94] Ulbricht, Claudia. *Multi-recurrent Networks for Traffic Forecasting.* Technical Report, Austrian Research Institute for Artificial Intelligence, Schottengasse 3, A-1010, Vienna, Austria, 1994.

[Vemuri94] Vemuri ,V. Rao., and Robert D. Rogers (editors). *Artificial Neural Networks : Forecasting Time Series*. IEEE Computer Society Press, Los Alamitos, CA, 1994.

[Waibel89] Waibel, A., Hanazawa,T., Hinton, G., Shikano, K., and Lang, K. "Phoneme Recognition Using Time-Delay Neural Networks". In *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol.37, 1989, pp. 328-339.

[Watanabe] Watanabe, T., Ikeda, H., Ohkawa, Y., and Satou, A. " Development of SIMTRACS and Its Applications". Proceedings of the *5th IFAC/IFIP/IFORS Conference*, Vienna, Austria, 8-11 July, 1986, pp. 185-190.

[Watson93] Watson, S. M. "An Introduction to Adaptive Forecasting Techniques". *ITS*

Technical Note 233. Institute for Transport Studies, University of Leeds,, Leeds, 1993.

[Weigend92] Weigend, S. Andreas, and A. Neil Gershenfeld (editors). *Time Series Prediction : Forecasting the Future and Understanding the Past.* Proceedings of the NATO Advanced Research Workshop on Comparative Time Series Analysis, Santa Fe, New Mexico, U.S.A, May 14-17, 1992.

[Weiss90] Weiss, Sholam M and Kulikowski, Casimir A. " *Computer Systems that Learn : Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems*". Morgan Kaufmann Publishers, Inc. California, 1990.

[Werbos74] Werbos, P. J. " *Beyond Regression : New Tools for Prediction and Analysis in the Behavioral Sciences* ". Ph.D Thesis, Harward University, 1974.

[Werbos88] Werbos, P. J. " Generalization of Back-propagation with Application to a Recurrent Gas Market Model". *Neural Networks*, 1988, Vol. 1, pp. 339-356.

[Werbos90] Werbos, P. J. "*Neurocontrol and Related Techniques*". Maren, A. J., Ed.; Academic Press: New York, 1990.

[WerbosJ] Werbos, J. Paul. "Backpropagation Through Time: What It Does and How To Do It ". *Proceedings of the IEEE*, Vol. 78, No. 10, October, 1990.

[Williams89] Williams, R. J., and Zipser, D. "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks". *Neural Computing*, 1989, Vol.

1, pp. 270-280.

[Wong90] Wong, F., and Tan, Pan Yong. *"Neural Networks and Genetic Algorithm for Economic Forecasting"*. Institute of Systems Science, National University of Singapore, 1990.

[Wu92] Wu, Lizhong and Fallside, F. "*Fully Vector-Quantised Neural Network-Based Code-Excited Non-Linear Predictive Speech Coding*". Technical Report CUED/F-INFENG/TR.94. Cambridge University Engineering Department, Cambridge, United Kingdom, 1992.

[Wu93] Wu, Lizhong and Niranjan, Mahesan. "*Building Energy Data Analysis and Prediction with Recurrent Neural Nets*". Technical Report CUED/F-INFENG/TR.124. Cambridge University Engineering Department, Cambridge, United Kingdom, April 1993.

[Young] Young, E. F., and Chan, L. W. "Using Recurrent Network in Time Series Prediction". In *World Congress on Neural Networks*, Vol IV., 1993, pp. 332-335.

[Yun96] Yun, Y. Sang, Namkoong, Seong, Shin, W. Seung, Rho, H. Jeong, and Choi, U. Jong. "Application of a Recurrent Neural Network to Traffic Volume Forecasting". In *Proceedings of the Third Annual World Congress on Intelligent Transport Systems*. Orlando, Florida, USA, October 14-18, 1996.

[Zhang97] Zhang, Hongjun, Ritchie, G. Stephen, and Lo, Zhen-Ping. "Macroscopic Modeling of Freeway Traffic Using an Artificial Neural Network". In

Transportation Research Record 1588, TRB, National Research Council, Washington,D.C., 1997, pp. 110-119.