

2012

Predicting Traffic on the Maine Turnpike Using Artificial Neural Networks

Robert Swain MS
University of Southern Maine

Follow this and additional works at: <https://digitalcommons.usm.maine.edu/etd>

 Part of the [Computer Sciences Commons](#)

Recommended Citation

Swain, Robert MS, "Predicting Traffic on the Maine Turnpike Using Artificial Neural Networks" (2012). *All Theses & Dissertations*. 94.
<https://digitalcommons.usm.maine.edu/etd/94>

This Open Access Thesis is brought to you for free and open access by the Student Scholarship at USM Digital Commons. It has been accepted for inclusion in All Theses & Dissertations by an authorized administrator of USM Digital Commons. For more information, please contact jessica.c.hovey@maine.edu.

Predicting Traffic on the Maine Turnpike Using Artificial Neural Networks

by

Robert Swain

A thesis submitted in partial fulfillment
of the requirements for the degree of
M.S. of Computer Science
(Department of Computer Science)
at The University of Southern Maine
2012

Thesis Committee:

Associate Professor Clare Bates Congdon, Advisor
Adjunct Professor David Bantz
Professor Charles Colgan

© Robert Swain 2014

All Rights Reserved

The University of Southern Maine

Department of Computer Science

September 2014

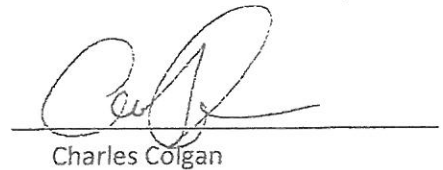
We hereby recommend that the thesis of Robert M. Swain entitled *Predicting Traffic on the Maine Turnpike Using Artificial Neural Networks* be accepted in partial fulfillment of the requirements for the degree of Master of Computer Science.



Clare Congdon, Advisor



David Bantz



Charles Colgan

Accepted



James E. Graves, Dean
College of Science, Technology and Health

TABLE OF CONTENTS

LIST OF FIGURES	v
LIST OF TABLES	vi
ABSTRACT	i
CHAPTER	
I. Introduction	1
II. Background	3
2.1 Comparison of Neural Networks and Statistical Forecasting Models	6
2.2 Highway Traffic Forecasting	7
2.3 Holiday Forecasting	10
III. ARIMAX Methodology	12
3.1 Introduction to ARIMAX	12
3.2 ARIMAX Models in this Thesis	15
3.3 ARIMAX Models in Depth	16
IV. ANN Methodology	19
4.1 Network Type	19
4.1.1 Simplicity	19
4.1.2 Multilayer	20
4.2 Training Algorithm	20
4.3 Activation Function	21
4.4 Number of Hidden Layers	22
4.5 Number of Hidden Nodes	23
4.6 Number of Input and Output Nodes	23

4.7	Holiday Multiplier	24
V.	Description of Experiments	26
5.1	Experiment 1: Hidden Nodes	27
5.2	Experiment 2: Comparison of ANN to ARIMAX	28
VI.	Results	30
6.1	Number of Hidden Nodes	30
6.2	ARIMA vs. ANN	36
6.2.1	ANN MAPEs	36
6.2.2	Actual Traffic Counts vs. Predictions	38
6.2.3	ARIMA MAPEs vs. ANN MAPEs	43
VII.	Discussion	45
7.1	Number of Hidden Neurons	45
7.2	ARIMA vs. ANN	47
7.3	Other Considerations	49
VIII.	Future Work	52
BIBLIOGRAPHY	55

LIST OF FIGURES

Figure

2.1	A simple example of an artificial neural network.	5
6.1	Number of Hidden Nodes: Memorial Day	32
6.2	Number of Hidden Nodes: Fourth of July	33
6.3	Number of Hidden Nodes: Labor Day	34
6.4	Number of Hidden Nodes: Columbus Day	35
6.5	Forecast MAPEs Using ANN	37
6.6	Predictions Vs. Actuals: Memorial Day	39
6.7	Predictions Vs. Actuals: Fourth of July	40
6.8	Predictions Vs. Actuals: Labor Day	41
6.9	Predictions Vs. Actuals: Columbus Day	42

LIST OF TABLES

Table

3.1	Northbound ARIMAX Model Summary	16
3.2	Southbound ARIMAX Model Summary	16
6.1	Forecast MAPEs: Memorial Day	43
6.2	Forecast MAPEs: Fourth of July	44
6.3	Forecast MAPEs: Labor Day	44
6.4	Forecast MAPEs: Columbus Day	44

ABSTRACT

Predicting Traffic on the Maine Turnpike Using Artificial Neural Networks

by

Robert Swain

In this thesis, I compare the forecasting accuracy of statistical models with artificial neural networks when predicting daily traffic counts on the Maine turnpike over holiday weekends. I review the relevant literature and discuss the methodological choices behind both modeling techniques. Two experiments are outlined and performed: one to determine the optimal structure of the artificial neural network, and another to compare holdout forecasts of each modeling technique. In the first experiment, my results indicated that a relatively small number of hidden nodes (11) were sufficient to produce accurate forecasts using the artificial neural networks. In the second experiment, I found that the artificial neural networks produced more accurate forecasts than the statistical models for most days of the holiday weekends. Due to the added complexity of artificial neural networks, as well as their “black box” nature, I hesitate to conclude that artificial neural networks are significantly better at forecasting traffic counts than the statistical models. I conclude that artificial neural networks show much promise in the area of traffic forecasting and require more consideration in future work.

CHAPTER I

Introduction

The goal of this thesis is to develop an artificial neural network methodology that produces reasonable forecasts of the daily traffic counts on the Maine Turnpike. The results of said forecasts will then be compared to the accuracy of forecasts produced by ARIMA statistical models for a number of different holiday weekends.

The inspiration for this work comes from the Maine Center for Business and Economic Research (CBER) at the University of Southern Maine. CBER is an University Center that “provides high-quality applied research and technical assistance services to Maine’s private- and public-sector organizations”[11]. One of the projects that CBER has been involved with is an ongoing project with the Maine Turnpike Authority (MTA) to provide a variety of forecasts. The part of the project that this thesis tries to address is the forecasting of daily traffic counts during holiday weekend periods at the York toll plaza. There are many other parts to the project, but this particular forecast has historically been of the most interest as well as one of the most difficult to do accurately.

The forecasting of daily traffic counts during holiday weekend periods is of great interest for a number of reasons. Firstly, these holiday weekends usually have the highest volume of vehicles of any time during the year. The MTA is very interested in just how many cars will coming through in order to schedule toll workers

and construction projects so keep traffic flowing as efficiently as possible. Secondly, the Maine tourism industry uses these forecasts to get an idea of how busy tourist destinations will be during this holiday weekends. Many tourist destinations such as lobster pound restaurants need to schedule extra workers for these weekends and having an idea of how many tourists will be coming into the state is a big help. Thirdly, and from a forecaster's perspective, the holiday weekends are of interest because they are typically the most volatile periods during the year. These periods are the most challenging to forecast and as such offer the greatest capacity for improvement.

In the past, CBER has used a statistical time series modeling approach to forecast the aforementioned holiday weekend periods. Recently, these statistical models have had some difficulty producing forecasts that meet the MTA's expectation of accuracy. This thesis work is an exploration of an alternate forecasting attempt using artificial neural networks. I chose to use a standard 3-layer feedforward neural network and found that while it did not significantly outperform the statistical models, it did show that neural networks are viable options for forecasting traffic on the Maine turnpike and should be explored further.

The following chapter (Chapter II) contains a review of the relevant literature. Chapter III describes the statistical models that CBER currently uses. Chapter IV describes the artificial neural network methodology I employed in this thesis. Chapter V outlines the experiments that I performed to test the accuracy of the statistical model forecasts versus the artificial neural network forecasts. Chapter VI lays out the results of my experiments and I discuss the implications of those results in Chapter VII. Finally, in Chapter VIII, I conclude with a discussion of where I would like to take this research in the future.

CHAPTER II

Background

In this section I will briefly describe the two modeling techniques used in this thesis and then follow with a review of the relevant literature. The following section will outline the two modelling techniques in much more detail.

The statistical modeling technique used to forecast traffic on the Maine turnpike is known as Auto-regressive Integrated Moving Average (ARIMA) time-series forecasting. This modelling technique is specifically designed for time-series applications. A time-series is a set of ordered observations that typically occur at regular intervals. A good example of a time-series are the daily stock prices on the New York stock exchange. An ARIMA model attempts to model the daily fluctuations in price of a stock using only the past values of the stock itself. If the price of a stock always drops on Fridays, an ARIMA model would be able to quickly find this out and incorporate it into the model. There is also a variant of an ARIMA model called an ARIMA model with exogenous variables (ARIMAX). This type of model functions exactly like an ARIMA model, but also incorporates other variables such as other stocks. This allows the model to utilize historical trends as well as other correlated variables to aid in predictions.

The modeling technique that I chose to compare and contrast with ARIMA is an artificial neural network (ANN). This modeling technique is a type of artificial

intelligence that was originally designed to mimic the way our brains were thought to work. The key to this modeling technique is creating many artificial neurons with weighted connections. An artificial neuron is essentially just an equation that takes a few numbers as inputs, adds them together, scales the sum, and outputs the result. The connections between the neurons are weighted; each connection multiplies the number that is passing through the connection by a fixed value that is typically less than one. An example of an artificial neural network is shown in Figure 2.1. In this example, there are two input variables. These values coming from these input variables are multiplied by the weights (w_1 , w_2) as they travel to the hidden node. The hidden node adds the weighted values together, scales the sum so that it stays between zero and one, and then passes the result on to the output. The connection weights are determined by a training algorithm (of which there are many).

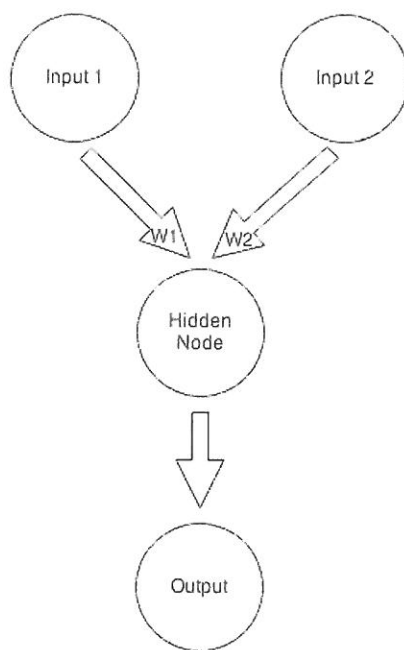


Figure 2.1: A simple example of an artificial neural network.

2.1 Comparison of Neural Networks and Statistical Forecasting Models

The primary motivation for this thesis was the thought that neural networks might be able to improve upon the traffic forecasts produced by CBER's ARIMA models. The next logical step was to review the existing literature and see how neural networks fared against ARIMA in various time series forecasting situations. Fortunately, the current literature contains a number of papers that compared the performance of neural networks and ARIMA models.

More and Deo [9] compared feed forward and recurrent neural networks to ARIMA as applied to forecasting wind. Wind speed forecasts were performed using the three methods at the daily, weekly, and monthly levels for two coastal locations in India. The authors found that both types of neural networks significantly outperformed the ARIMA model for all of their various different experiments.

Zhang and Qi [19], Castellano-Mendez et al [3], and Prybutok et al [12] all produced forecasts with both feed forward neural networks and ARIMA models in a variety of situations. Zhang and Qi used a number of different monthly time series and ultimately found that the neural networks performed better than the ARIMA model for most of the time series. Castellano-Mendez et al modeled the monthly and daily behavior of the runoff of the Xallas river in Spain and found that the performance of the neural network was superior to the performance of the ARIMA. Prybutok et al predicted daily maximum ozone concentrations in Houston using ANNs, ARIMA, and regression. They found that the ANN outperformed both statistical methods.

These papers make the point that ANNs have the capability to outperform ARIMA models when forecasting time series. It certainly isn't guaranteed that ANNs will always be a better solution than an ARIMA model, but the literature does suggest that ANNs are definitely a viable alternative to more traditional statistical methods.

2.2 Highway Traffic Forecasting

The existing literature also provided some interesting patterns within the focus of traffic prediction research. While there is a substantial amount of research relating to traffic flow prediction, the vast majority is concerned with short-term forecasts. In these works, short-term is usually defined as between 5 and 30 minutes. These short-term forecasts are used in real-time systems to predict congestion and to give drivers information about what kind of travel times to expect in the near future.

Within the set of short-term traffic forecasting literature, there was a vast array of different approaches that the researchers took. Approximately half of the literature found employed some type of ARIMA methodology. A slightly smaller group used neural networks. The rest of the literature found used other statistical and artificial intelligence techniques.

Of those that used ARIMA, the one that most resembled the work that CBER has done was Williams and Hoel [17]. In this work, the researchers used a seasonal ARIMA model to forecast traffic in 15-minute intervals in both London, England and Atlanta, Georgia. The authors found that there were significant seasonal effects at the daily and weekly level. In contrast to the CBER ARIMA model described in this thesis, Williams and Hoel did not include any exogenous variables. The authors performed holdout forecasts and compared the results of their model to the results of several heuristic approaches (random walk, historical average, and deviation from historical average). The seasonal ARIMA model outperformed the heuristics, though the “deviation from historical average” heuristic came in a close second.

Another paper that utilized the ARIMA methodology was Yu and Zhang [18]. Similar to Williams and Hoel, Yu and Zhang were primarily interested in forecasting 15-minute traffic volumes in Beijing. The authors of this work decided that different periods of each day might benefit from having their own ARIMA models. The authors

determine that there are four different patterns within a typical day; a bottom pattern, an ascending pattern, a peak pattern, and a descending pattern. For each of these patterns, an independent ARIMA was created. When forecasting, the overall model determines which ARIMA is appropriate to use for each 15-minute interval forecasted. The authors found that their “switching” ARIMA model outperformed the random walk, historical average, and informed historical average heuristics.

Van Der Voort, Dougherty and Watson [15] implemented a similar solution in their 1995 paper. In that work, they were forecasting 30-minute and one hour time intervals in Beaune, France. Like the previous paper, the authors constructed separate ARIMA models for different situations. Unlike that previous paper, instead of deciding empirically how many different ARIMA models there should be, the authors used a Kohonen self-organizing map to classify which time periods were most similar and therefore warranted an ARIMA model. The authors concluded that their model performed as well as existing models but that they need to test it with more data before they could make any concrete conclusions.

Stathopoulos and Karlaftis [13] used state space modeling to forecast three-minute time periods on roads in and around Athens, Greece. State space modeling is similar to ARIMA modeling, but allows for multiple dependent variables. The authors built a state space model as well as several ARIMA models and found that the state space model was superior. The authors’ reasoning for this was that since state space models are able to model multiple time series concurrently, the model was able to consider the traffic flow throughout the entire road system, which enabled the individual forecasts to be more accurate. The authors did note that the ARIMA models performed very well at the entrances of the road system.

Neural networks were also featured prominently in a number of papers regarding traffic forecasting. Ishak and Alecsandru [6] created a number of different types of neural networks in order to forecast time periods ranging from 5 to 20 minutes on

Florida roads. They examined multilayer feed-forward networks in addition to some more complex neural networks such as modular networks, hybrid principal component analysis networks, and co-active neuro-fuzzy inference systems. The authors concluded that no one neural network type was optimal for all of the different tests that they did. However, all of the different neural networks outperformed the basic statistical heuristic benchmarks that they included.

Zheng, Lee, and Shi [20] compared stand alone back propagation and radial basis function neural networks with a Bayesian combined model that incorporated both of the stand-alone neural network techniques into it. The combined model was able to switch back and forth between the two stand alone models in order to get the best of the two forecasts depending on which time periods were being forecast. This allowed the combined model to outperform both stand alone models overall.

Dougherty and Cobbett [5] implemented a back-propagation neural network to produce short-term forecasts in the Netherlands. Because of limited computing power, the authors used elasticity testing to cut down the number of inputs. The elasticity testing consisted of the authors varying the magnitude of the inputs in order to see the effect on the output. This allowed the authors to get an idea of which variables were most significant. Once the authors cut down the number of inputs, they concluded that the neural networks showed a lot of promise, but didn't produce any outstanding results.

Papers within the current literature that didn't fit either the statistical or neural network categories used a wide variety of different techniques. Chrobok et al [4] used an advanced cellular-automation micro-simulator model, Li et al [7] used a type-2 fuzzy logic approach, Sun et al [14] used a Bayesian network, and Van Lint et al [16] used a state space neural network. All of these approaches were focused on short-term traffic forecasting. All of them showed promise and beat various heuristic approaches, but none proved to be dominant.

This portion of the literature review served to reinforced that both ARIMA and neural networks are viable options for forecasting traffic. These works also provided some valuable examples of techniques to try in future research. The papers that utilized some sort of neural network methodology prove that there are a number of impressive techniques that work quite well for traffic forecasting, however before trying something more exotic, in this thesis, it was decided to start with the most basic neural network and possibly use these other papers as inspiration for future work.

2.3 Holiday Forecasting

During this literature review, it became apparent that there was a lack of literature regarding the effect of holidays when it comes to traffic forecasting. This is most likely the case because the vast majority of the papers deal with short-term forecasting, and holidays are not as much of a factor during short-term forecasting. It is also possible that with the focus of these papers being real-time traffic prediction with the goal to reduce momentary congestion, it doesn't matter too much if the predictions are off during holidays as long as the models work well during the majority of days which exhibit normal traffic patterns.

The paper that was the most relevant to holiday forecasting was Liu et al. [8] which forecasted hourly time periods for several days. They use a non-parametric regression technique called k-nearest neighbors. With this technique, the authors manage to average around 8% MARE (mean absolute relative error) with their hourly forecasts over holiday periods. The non-parametric regression technique also allowed the authors to deal with missing data values quite effectively. The authors concluded that non-parametric regression is a powerful technique that warrants more investigation. This paper did not give much insight into how to deal with the effects of holidays in this thesis as the non-parametric regression took care of the holidays seemingly

without any explicit intervention from the authors. Despite that fact, the paper did provide insight into how accurate a holiday model could be.

The only other paper in the current literature that dealt with the impact of holiday effects was Bakirtziz et al. [1]. This paper utilized a three-layer feed-forward neural network to forecast short-term loads on the Greek power system. The way in which the authors chose to deal with holidays was to compute an average deviation that holidays exhibited compared to non-holidays. This average holiday deviation was then applied to the forecasts in order to account for the holiday effect. The authors concluded that the model performed well for both normal days and holidays but that the errors for holidays were significantly higher than normal days.

Due to the lack of holiday forecasting literature, there appears to be a need to develop techniques that can account for the effect that holidays have. The “holiday multiplier” technique laid out in this thesis could fit into this void.

CHAPTER III

ARIMAX Methodology

3.1 Introduction to ARIMAX

When CBER got the forecasting contract from the MTA, ARIMAX was chosen as the statistical modelling technique to use. ARIMAX models are a good choice when you have a non-stationary dependent time series and variables that you think would be good predictors of the dependent variable. This is the case for the MTA project. The traffic counts on the Maine turnpike are good examples of non-stationary dependent variables because the number of cars travelling on the turnpike has been steadily increasing over the years and it is intuitive to think that variables such as rainfall, snowfall, and the price of gas may all affect the number of cars on the road for a given day. CBER chose to create one ARIMAX model for forecasting northbound traffic and another separate ARIMAX model for forecasting southbound traffic. The models used in this these are outlined in tables 3.1 and 3.2.

In total, 154 quantitative and qualitative variables were made available to the ARIMAX models. The quantitative variables were: year (the current year), Day_actual (the day of the month), precip (the number of inches of liquid precipitation that day), snow (the number of inches of snow that day), maxtemp (the high temperature for the day), petrolREGULAR (the average price of a barrel regular gasoline that week), petrolALLGRADE (the average price of all grades of gasoline that week), and

crudeprice (the price of a barrel of crude oil). Each of these quantitative variables is then lagged to create 10 total lagged versions of each variable. Lagging is a technique of shifting a variable so that past values are made available to the model. In this case for example, the current price of regular gasoline is available, yesterday's price is available (lag 1), the price two days ago (lag 2), etc. For the ARIMAX models in this paper, I generated lags of 1-8, 364, and 365. These lags coincide with the lags that CBER has used in their statistical models. In the tables below, variable names are made up of the original coded name, followed by an underscore and a number indicating the lag. For example, snow_8 refers to the inches of snowfall 8 days ago.

The qualitative variables used in this thesis are all in the form of binary dummy variables. Qualitative dummy variables are variables that one creates to stand in for phenomena that are not inherently quantitative. An example of this is a dummy variable for Monday. If the day in question is a Monday, then the variable is 1. If it is a different day of the week, then the dummy variable would be 0. This is a good way to let the model know about phenomena such as the day of the week, holidays, seasons, etc. The qualitative variables used in this thesis were: memorial (has a value of 1 for the Friday, Saturday, Sunday, and Monday of Memorial Day weekend), memorialmon (has a value of 1 for just Memorial Day), july4d (has a value of 1 for July 4), fourth (has a value of 1 for the four days of the July fourth weekend), labor (has a value of 1 for the four days of the labor day weekend), labormon (has a value of 1 for Labor Day), columbus (has a value of 1 for the four days of the columbus day weekend), colmon (has a value of 1 for Columbus Day), day1 - day6 (binary dummy variables for each day of the week except Saturday which is the absence of the other variables being 1). As with the quantitative variables, these qualitative binary variables also have associated lagged versions for lags 1-8, 364, and 365. For example, if the variables labormon_8 has the value of 1, this indicates that Memorial Day was 8 days ago.

In addition to the quantitative and qualitative variables, the ARIMAX models utilize a number of ARIMA terms. ARIMA terms are auto-regressive (AR) and moving-average (MA) terms. AR terms are lagged versions of the dependent variable and MA terms are lagged versions of the forecasting error. For example, an AR term of 1 corresponds with the traffic counts from a day ago. An MA term of 1 is the forecasting error of the ARIMA model a day ago. These terms are typically very helpful for time series that exhibit a random-walk type of pattern. This is when future values of the dependent variable are somewhat impacted by past values of the dependent variable. For example, if the traffic was 30,000 on Tuesday, and no other information was available, for Wednesday, it would be reasonable to forecast something close to 30,000 rather than, for example, 300. This isn't to say that the past values of the dependent variable cause the future values of the dependent variable, but they can be good indicators.

There are two ways to specify AR and MA terms in ARIMA models. Many textbooks use the "order" technique in which only the highest lag AR and MA terms are specified and all lower order lags are automatically included. This can work well when using ARIMA models with low order AR and MA terms, but is not always the optimal strategy. In the case of the traffic forecasting in this thesis, it was more appropriate to use a subset methodology. This is when individual AR and MA terms are specified. This allows for very large AR and MA terms to be included without including all of the lower order terms. This is important when dealing with seasonal data, as there are likely to be daily, weekly, monthly, and yearly patterns present. It is essential that the AR and MA terms are allowed to account for these large lags, without also bringing in the intervening insignificant lags.

The final piece of the ARIMAX methodology is the integrated element which refers to time series differencing. Differencing a time series is similar to taking the derivative in mathematics. It is a way of changing a variable from raw values into the change

over time of those values. With time series, this is accomplished by subtracting a prior value from the current value. An example of this would be a 1-day differencing scheme in our traffic data. For each daily traffic count in our data, we would subtract the prior days value. The new variable then describes by how much did the traffic change since the prior day. Differencing in this way is often used in time series analysis to change a non-stationary variable into a stationary variable; to turn a variable with a changing mean into a variable with a stable mean.

With non-stationary time series, it is often necessary to perform one or two differences on the data before beginning ARIMA modelling. Differencing can transform the input series into a stationary version of itself. For example, with a time series that exhibits a linear trend, a difference of one, in which each value is subtracted from the next value in the series can result in a time series devoid of linear trend. Differencing is necessary for two reasons. The first reason is that only stationary time series can be modeled with AR and MA terms. There are some statistical reasons for this, but the main reason is that ARIMAX forecasting is based on past values and assumes that the mean and variance of the dependent series in the past will be the same in the future. The model basically roots its predictions in the past mean and variance and will fail if the mean and variance are changing over time. The second reason to difference is to make the model's job easier. If the original time series exhibits a linear trend or a quadratic trend, this is just more variation that the model needs to figure out. Differencing and removing that large amount of variation allows the model to work on a simpler time series.

3.2 ARIMAX Models in this Thesis

There are two ARIMAX model used in this thesis. The first is a model tailored to the northbound traffic time series of the Maine Turnpike at the York toll plaza, and the second is a model tailored to the southbound traffic time series.

Table 3.1: Northbound ARIMAX Model Summary

Model Component	Description
Difference	1,7
AR terms	1, 2, 3, 4, 365
MA terms	7, 364
Exogenous Variables	year_365, fourth_364, snow, precip, july4d, columbus, memorial_2, memorial, memorialmon, labor_2, labor, labormon, columbus_2, colmon_1, memorialmon_1, memorial_1, maxtemp_8, fourth_2, fourth

Table 3.2: Southbound ARIMAX Model Summary

Model Component	Description
Difference	1,7
AR terms	1, 2, 3, 4, 364
MA terms	5, 7, 365
Exogenous Variables	snow, labormon, memorialmon, july4d_1, colmon, snow_1, labormon_364, precip, labor_7, memorialmon_364, fourth_364, labor, fourth_365, july4d, memorial_2, colmon_1, labormon_2, july4d_2, snow_8, memorialmon_1, memorialmon_2

The two models are summarized in the tables below as well as described in greater detail below.

3.3 ARIMAX Models in Depth

Difference The northbound ARIMAX model uses a differencing scheme of 1 and 7. This differencing scheme was found to make the dependent time series trend

stationary using the augmented Dickey-Fuller test (a standard statistical test for confirming stationarity of mean and variance in a time series). This makes sense intuitively as one would expect a steady increase in automobile traffic over time. This is the reason for differencing by 1. The difference by 7 makes sense because of the strong weekly pattern within the time series. Weekday traffic consists of very regular commuter traffic for the most part, whereas weekend traffic is much more irregular and at different times than weekday traffic.

Like the northbound model, the southbound model employs a differencing scheme of 1 and 7. This was also verified to bring the dependent time series to stationarity using the augmented Dickey-Fuller test.

Auto-Regressive and Moving Average Terms The northbound ARIMAX model has quite a number of AR and MA terms. All terms were significant to the 0.05 level. The total list of terms is shown in the table above. It is often difficult to come to an intuitive understanding of why certain terms are significant and others are not as these significance levels must be considered within the presence of all of the other elements of the model. For example, it is difficult to tell a story about why the 4th AR lag is significant and not the 5th. It would be incorrect to try to come to an understanding about the significance of these two terms without considering all of the other aspects of the model. It's entirely possible that 4 is only significant because it's trying to slightly offset the influence of 2. Or perhaps these terms are interacting with some of the exogenous variables. It is often sufficient to say that all of the terms are significant.

As seen in the above table, it is evident that the southbound model has many AR and MA terms in common with the northbound model. Once again, all terms were significant to the 0.05 level. Even though it is difficult to attribute intuitive sense to individual terms within a model, it is reassuring to see that the northbound and

southbound model share many similar terms. The two dependent time series look fairly similar, and it's nice to see that ARIMAX treats them somewhat similarly.

Exogenous Variables Both the northbound and southbound models include many exogenous variables. The majority of these exogenous variables are binary dummy variables that only help explain one day out of the year. It is perhaps easy to understand how a July Fourth binary dummy variable might be significant. Even though the variable only helps with one day out of an entire year, the irregularity of that one day is significant enough that it can use an entire variable to itself. There are enough days like this throughout the year, that the model finds a great number of these variables to be significant. In particular, the dummy variables that coincide with the co-occurrence of a holiday and a weekend are of particular use to the model. This is intuitively attractive since regardless of what day of the week the holiday happens to fall on, the nearest weekend is when traffic usually spikes.

The remaining exogenous variables are various continuous variables such as inches of precipitation, inches of snow, and maximum temperature. These variables are the typical explanatory variables that modelers like to see. These variables make a lot of easy, intuitive sense. They also provide a brief "sanity check" that yes, snow does significantly affect highway traffic.

CHAPTER IV

ANN Methodology

After deciding to use an artificial neural network (ANN), there are many subsequent questions that must be answered about how exactly to implement it. I explore some of the major ANN methodology choices below.

4.1 Network Type

For this thesis, the multilayer feed-forward neural network was deemed most appropriate. There are many other types of neural networks and new types are being created all the time. The multilayer feed-forward architecture was one of the first developed [2] and has a number of advantages. In particular, there are a couple of good reasons to use this type of neural network for the problem presented in this thesis.

4.1.1 Simplicity

One reason for choosing the feed-forward type is its simplicity. Among the various types of neural networks, feed-forward is relatively easy to understand. All of the information that is passing through the network passes in one direction: forward. This is in contrast to other network types such as recurrent neural networks. In these types of networks, information can travel forward through the network but can

also loop backwards from a later neuron to an earlier neuron. These network types have their uses, and have even been used for time series prediction, however they are much more complicated to understand and implement correctly. The relative ease of implementation, coupled with the fact that the feed-forward type is proven to be effective in time-series prediction, made this type of network a good choice for this thesis.

4.1.2 Multilayer

My reason for choosing the multilayer feed-forward over a single-layer feed-forward comes down to the math behind these networks. A single-layer feed-forward network is, essentially, a regression [2]. The inputs to the neural network are multiplied by their respective weights and combined to produce the output. This is exactly what happens in a regression. If you use a linear activation function, the single-layer network will behave exactly like a multiple regression. If a sigmoidal activation function is used, the network will function in a manner similar to a logistic regression. This can be quite useful in some applications, but lacks the power of a multilayer network. As a single layer network would only be capable of matching an ARIMAX model, for this thesis, a multilayer network will be used in order to produce an interesting contrast with the ARIMAX model and possibly exceed the capabilities of an ARIMAX model.

4.2 Training Algorithm

Once the type of ANN to be used has been established, the method of training must be decided upon. In the case of feed-forward neural networks, the most commonly used training algorithm is Backpropagation. This training algorithm starts out by assigning random values to the weights. It then uses a training dataset to compare predicted values with the ideal values from the data set. Backpropagation uses the

difference between the predicted value and the ideal value to adjust the weights in the neural network in order to get the predicted value closer to the ideal value. Over many iterations of this weight adjustment, the neural network gets better and better at matching the ideal output values.

Backpropagation requires a learning rate and a momentum rate. These rates dictate how much the algorithm can change the weights during any given training iteration. If rates are chosen poorly, the algorithm can get stuck in a local minima, or can shoot right past the global minima. In either case, the algorithm will not be able to find the best combination of weights to produce accurate predictions. Choosing these rates used to be largely trial and error. Relatively recently researchers have come up with an automated version of backpropagation called Resilient Propagation. Resilient Propagation behaves almost exactly like backpropagation, however it adjusts the learning and momentum rates as it goes. This is very convinient as choosing the appropriate learning and momentum rates could be a paper in and of itself.

4.3 Activation Function

With resilient propagation chosen to be the training algorithm in this thesis, the choice of activation function is the next logical decision to make. The use of resilient propagation places some constraints on which activation functions can be used in the neural network. Resilient propagation, like backpropagation, requires that the activation function be differentiable. Even though a linear activation function is differentiable, because the derivative is a constant, it is not usable with resilient propagation. The two most popular options that are left are the hyperbolic tangent and sigmoid activation functions. The main difference between the two is that the hyperbolic tangent function can go negative and the sigmoid cannot. In this thesis, because the dependent variable (traffic counts) is always positive, the sigmoid function is a good choice.

4.4 Number of Hidden Layers

The number of hidden layers is the next important question to answer. Hidden layers and nodes are the key parts of a feed-forward neural network that allow the network to be more powerful than a regression. Hidden nodes allow the input variables to interact with each other when producing output. This is very similar to the idea of interaction terms in regression modeling. Hidden nodes allow a neural network to take the idea of interaction terms even farther by allowing all of the input variables to interact with each other an arbitrary number of times. This is what makes feed-forward neural networks so powerful.

When choosing the number of hidden layers in a feed-forward neural network, there are several options.

- 0 Hidden Layers
- 1 Hidden Layer
- 2 Hidden Layers
- >2 Hidden Layers

As previously mentioned in the first question, 0 hidden layers essentially leaves you with a regression. More than 2 hidden layers is usually unnecessary. For the vast majority of problems that can be solved with a feed-forward neural network, either 1 or 2 hidden layers is/are best. As stated on page 116 of Bishop [2], a feed-forward neural network with just one hidden layer is “capable of approximating any continuous functional mapping”. Since the problem of forecasting traffic fits into this category, 1 hidden layer seems appropriate. Using 2 hidden layers can be appropriate for some problems, but in the case of this thesis, it seems unnecessary.

4.5 Number of Hidden Nodes

Integrally linked to the question of how many hidden layers to use is the question of how many hidden nodes to use. Since one hidden layer is appropriate, that reduces the question of how many hidden nodes to how many hidden nodes in the single hidden layer.

When searching for an answer to the question of how many hidden nodes to use, it becomes clear that there is no good answer. The optimal number of hidden nodes can be completely different with different numbers of inputs, different numbers of hidden layers, different numbers of outputs, and what the data is that you are trying to use. A significant portion of the literature suggests some kind of trial and error approach to this question. There is a rule of thumb that the appropriate number of hidden nodes is most likely between the number of outputs and the number of inputs. In Chapter VI, a variety of different numbers of hidden nodes will be tried to determine which is closest to optimal.

4.6 Number of Input and Output Nodes

The number of input and output nodes is much easier to determine than the number of hidden nodes. The number of input nodes is equal to the number of inputs that are present in the dataset. In this thesis, I chose to use the same data that the ARIMAX models were using, although I cut down on the number of lags made available to the ANN. The ARIMAX models had access to lags 1-8, 364, and 365 for all of the variables. Providing this number of inputs to the ANN was computationally infeasible on the computer that I had available to me. Therefore, instead of the original 10 lags, I chose to use 1, 2, 7, 8, 364, and 365 with the ANN. This does mean that the ANN could possibly be at a slight disadvantage against the ARIMAX, but hopefully not too large. After cutting down on the number of lags of each input

variable, the ANNs in this thesis ended up having 168 variables made available to them, and therefore required 168 input nodes.

The number of output nodes is the number of outputs that you want. In this thesis the goal is to produce separate forecasts for the northbound and southbound lanes, so I chose two output nodes. These two output nodes allow the neural network to produce simultaneous forecasts for the northbound and southbound lanes of the Maine turnpike one day at a time.

4.7 Holiday Multiplier

The final piece of neural network methodology worth noting in this thesis deals with holiday bias, or the lack thereof. The data of interest in this thesis is daily traffic counts going back to 2001. The vast majority of the observations in this time series are normal (not a holiday) days. As a result, the neural network, when trained over this data, will be biased towards normal days. This means that the network will, most likely, be better at predicting normal days than predicting holidays. During training, there is not much incentive to improve the forecasting of holidays. Because of this, in early testing runs, it was evident that the error when forecasting normal days was extremely low, but holiday errors were very high. Since the holiday errors make up a very small portion of the overall error, the training algorithm could do a great job by concentrating on the normal days. A novel technique used in this thesis was to duplicate holiday observations. This naturally forces the training algorithm to put additional emphasis on holidays rather than normal days. I am calling this process the Holiday Multiplier. Essentially, each holiday weekend is marked as a period of interest in my dataset. Right before training, copies are made of all of these periods so that there are much greater numbers of holiday periods in the training set than before. This technique forced the resilient propagation training algorithm to focus more on holidays and great improvement in the forecasting of holiday weekend time

periods resulted.

CHAPTER V

Description of Experiments

As previously stated, my goal for this thesis is to explore the application of artificial neural networks to the problem of predicting daily traffic counts on the Maine Turnpike during holiday weekend periods. To this end, I devised two experiments that achieve this goal:

1. Explore the effects of differing numbers of hidden nodes in an ANN on holdout forecast accuracy of holiday weekends
2. Compare neural network performance to the ARIMAX model performance

For each experiment, the same core neural network methodology is employed (as described in Chapter IV):

- The network is a 3-layer feed-forward neural network.
- The hidden and output nodes use the sigmoid activation function.
- The resilient backpropagation training algorithm is used.
- The input layer consists of 168 nodes.
- The output layer consists of 2 output nodes.

5.1 Experiment 1: Hidden Nodes

The goal of this experiment is to examine the effect that different numbers of hidden nodes have on the forecasting performance of the neural network. In order to accomplish this goal, I perform holdout forecasts for each of the four holiday weekends in 2011. For each of these four holiday weekends, 10 different networks with differing numbers of hidden nodes are used.

I created 10 neural networks for each of the four holiday weekends in 2011 (Memorial Day, July 4th, Labor Day, and Columbus day). The 10 neural networks each use a different number of hidden nodes ranging from 2 to 84 following a logarithmic distribution (e.g., 2, 3, 5, 7, 11, 16, 24, 37, 55, 84). The reason for 10 different networks comes from the fact that computational power and time is a limiting factor. From preliminary experiments, it looked as though a relatively small number of hidden nodes would be adequate. This is the reason both for the logarithmic distribution of hidden nodes in the trials, as well as the reason for stopping at 84 (half the number of inputs).

Each of the 4×10 neural networks are trained for 5,000 training iterations using the resilient backpropagation training algorithm or until they reach an error of 0.5. From preliminary testing, 5,000 seemed to be a large enough number of training iterations to get an idea of how the network will perform. Once all of the networks are at 0.5 error, they are on an even playing field in order to gauge their performance during the holdout forecasts.

Regarding the data used, for each of the 4×10 neural networks, I have restricted the training data set to data up to the holiday of interest. For example, for Memorial day, the 10 networks are trained on data through the Thursday before the holiday. This simulates a real forecast because the network will have seen all of the data up until the holiday weekend, but not what will be part of the weekend forecast.

The holdout forecast for the three Monday holiday weekends (Memorial Day, Labor Day, Columbus Day), are made up of the forecasts from the Monday holiday as well as the 3 preceding days for both northbound and southbound. For example, the forecast for Memorial day weekend is made up of the forecasts from Friday, Saturday, Sunday, and Memorial day. The forecast for the July 4th holiday weekend is coincidentally the same as the other three holidays because in 2011, July 4th happened to fall on a Monday.

Once all of the forecasts are performed, a number of different performance metrics are calculated. There is an overall MAPE (mean absolute percent error), a Northbound MAPE, a Southbound MAPE, a Fri/Sat Northbound MAPE, and a Sun/Mon Southbound MAPE. These five metrics should give a good picture of how each network has done with its forecast.

Once all of the results are assembled, I will first try to determine, for each of the four holidays, which number of hidden nodes was most successful using the above metrics. I will then try to determine which number of hidden nodes was most successful for all of the holidays, if there was in fact one number of hidden nodes that outperformed the others.

5.2 Experiment 2: Comparison of ANN to ARIMAX

The goal of this experiment is to compare the performance of the neural networks to the performance of the ARIMA models. This is accomplished by producing holdout forecasts for the four holiday weekends in 2011 using both models. For the neural network, results of the previous experiments are used to pick the best number of hidden nodes. For the ARIMA model, forecasts from the two models that were built for this thesis are used.

The specific metrics that are used to compare the two models are as follows:

- Overall holiday weekend MAPE that includes both directions for Friday, Saturday, Sunday, and Monday
- Northbound holiday weekend MAPE
- Southbound holiday weekend MAPE
- MAPE of just Friday and Saturday of the Northbound holiday weekend traffic
- MAPE of Sunday and Monday of the Southbound holiday weekend traffic

For each of the four holiday weekends in 2011, these five metrics are produced for both models. The goal is that these metrics will give an accurate picture of what each model is capable in terms of forecasting performance.

CHAPTER VI

Results

In Chapter V, the experiments that were to be run were described. In this chapter, the results of those experiments will be shown. For discussion and conclusions, see Chapter VII.

6.1 Number of Hidden Nodes

The first set of experiments was designed to find what the optimal number of hidden nodes was for the neural network. Due to computational limitations, the number of runs and the number of hidden nodes were limited. For each of the four holiday weekends of interest in 2011, 10 different numbers of hidden nodes were tried. These numbers were 2, 3, 5, 7, 11, 16, 24, 37, 55, and 84. For each holiday and number of hidden nodes, 10 neural networks were created and trained.

For each of the four holiday weekends, five figures were created. Each of these figures displays how well the various networks performed on a holdout forecast of that weekend. The number of hidden nodes is on the x axis and the mean absolute percent error (MAPE) is on the y axis. The weekend in question is indicated at the top of the figure along with what numbers were included in the calculation of the MAPE. The figures that have “Northbound” at the top used all four of the northbound predictions (Friday, Saturday, Sunday, and Monday) in the calculation

of the MAPE. The “Southbound” figures do the same for all four of the southbound predictions. The “Overall” figures display MAPEs that took into account all four northbound predictions as well as all four southbound predictions for that particular weekend. The “Fri/Sat Northbound” figures display MAPEs that were calculated only using the northbound predictions from the Friday and Saturday of the weekend in question. The “Sun/Mon Southbound” figures display MAPEs using the southbound Saturday and Sunday predictions. Most of the hidden neuron numbers will have 10 dots plotted above them on each figure. If they do not, that is because the Resilient Backpropagation training algorithm was not able to get the network to converge within the allotted training time.

Figure 6.1: Number of Hidden Nodes: Memorial Day

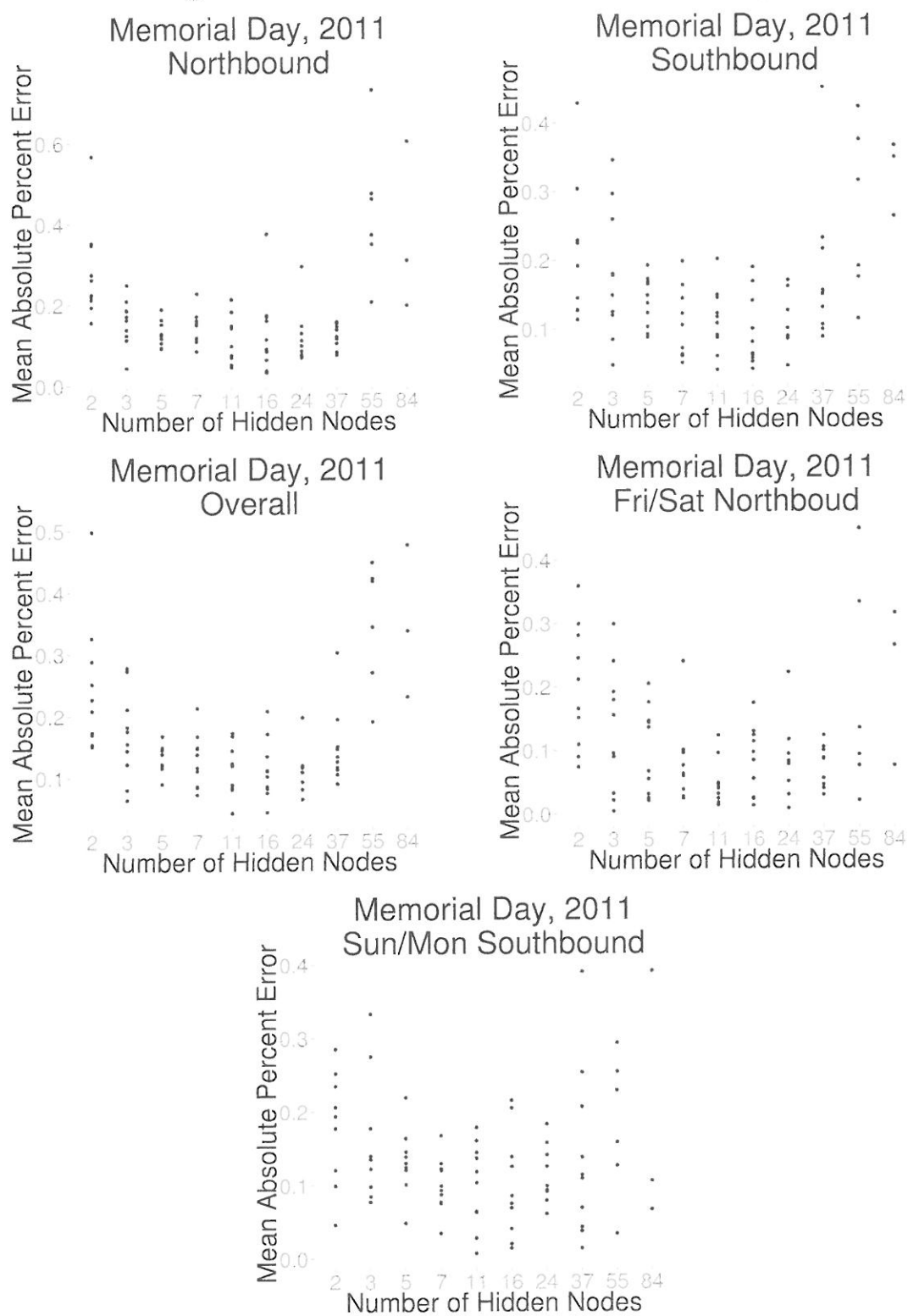


Figure 6.2: Number of Hidden Nodes: Fourth of July

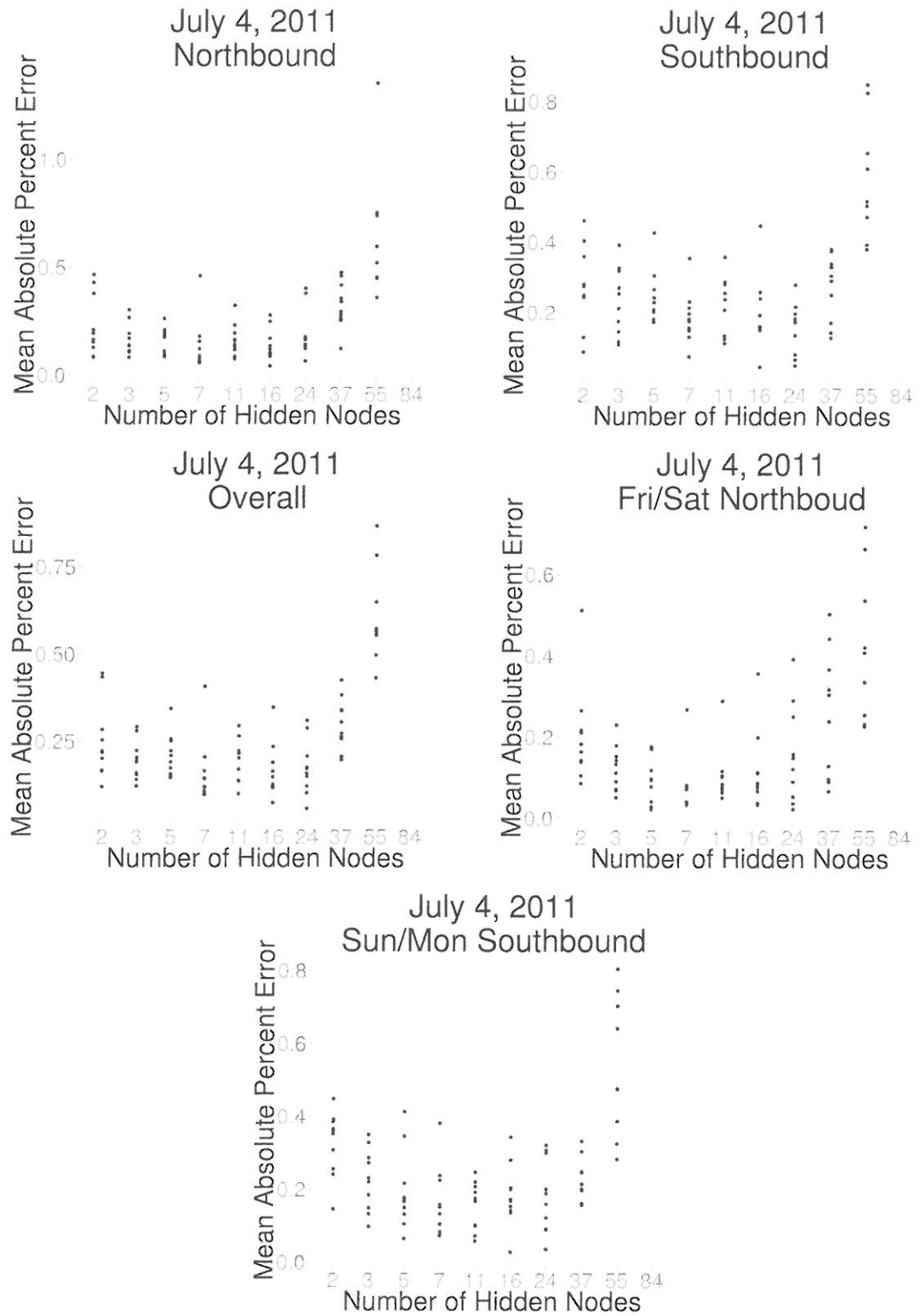


Figure 6.3: Number of Hidden Nodes: Labor Day

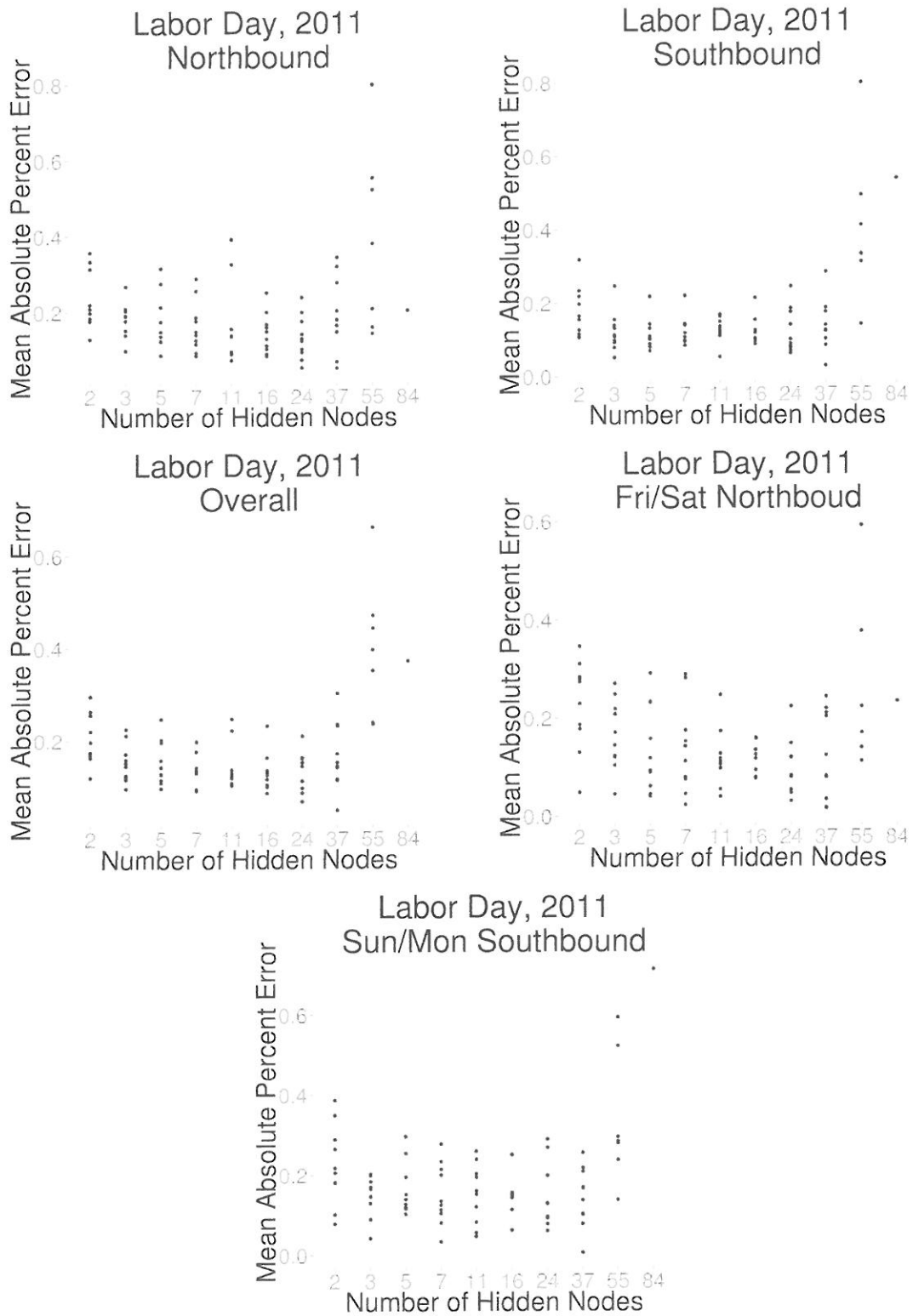
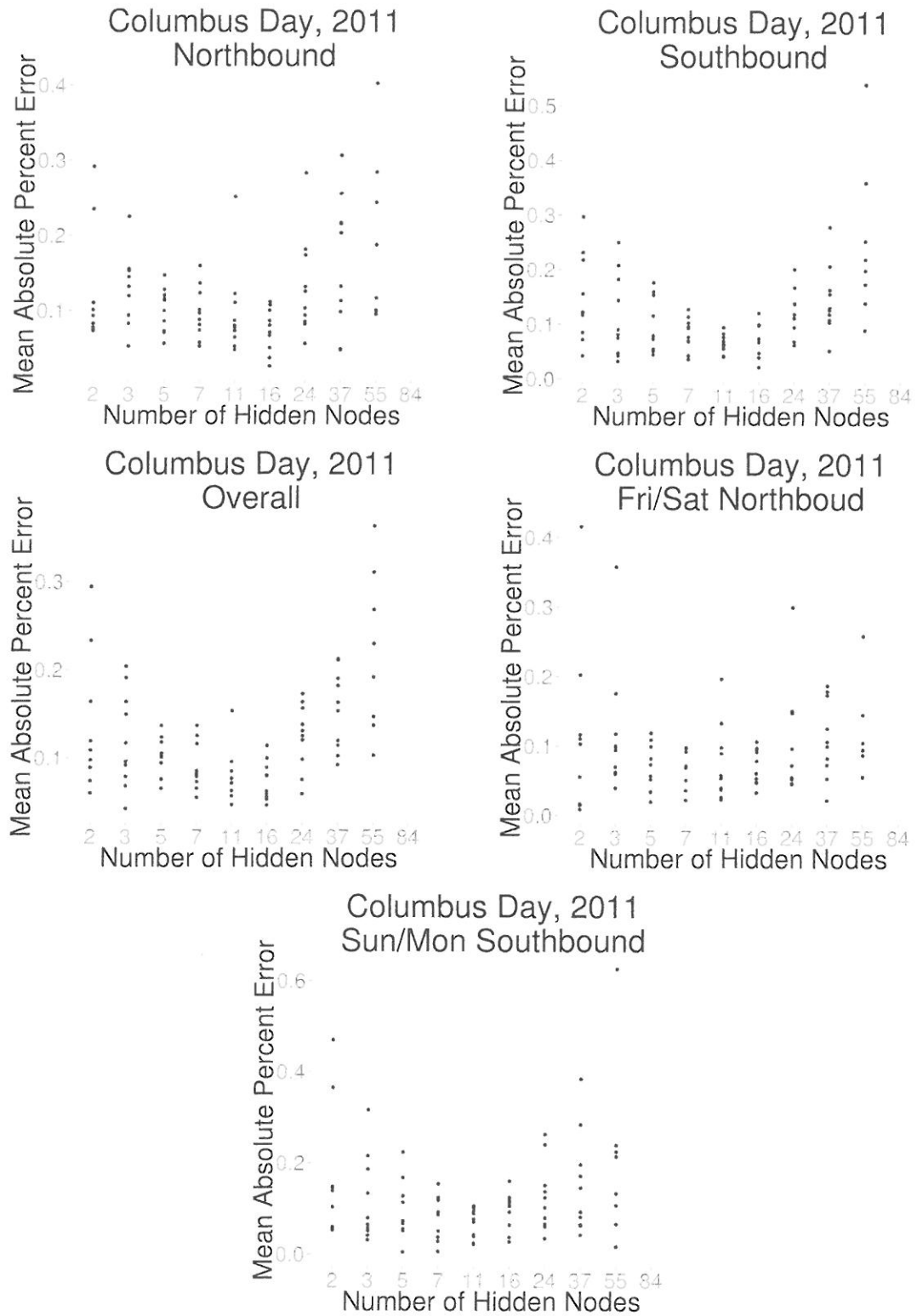


Figure 6.4: Number of Hidden Nodes: Columbus Day



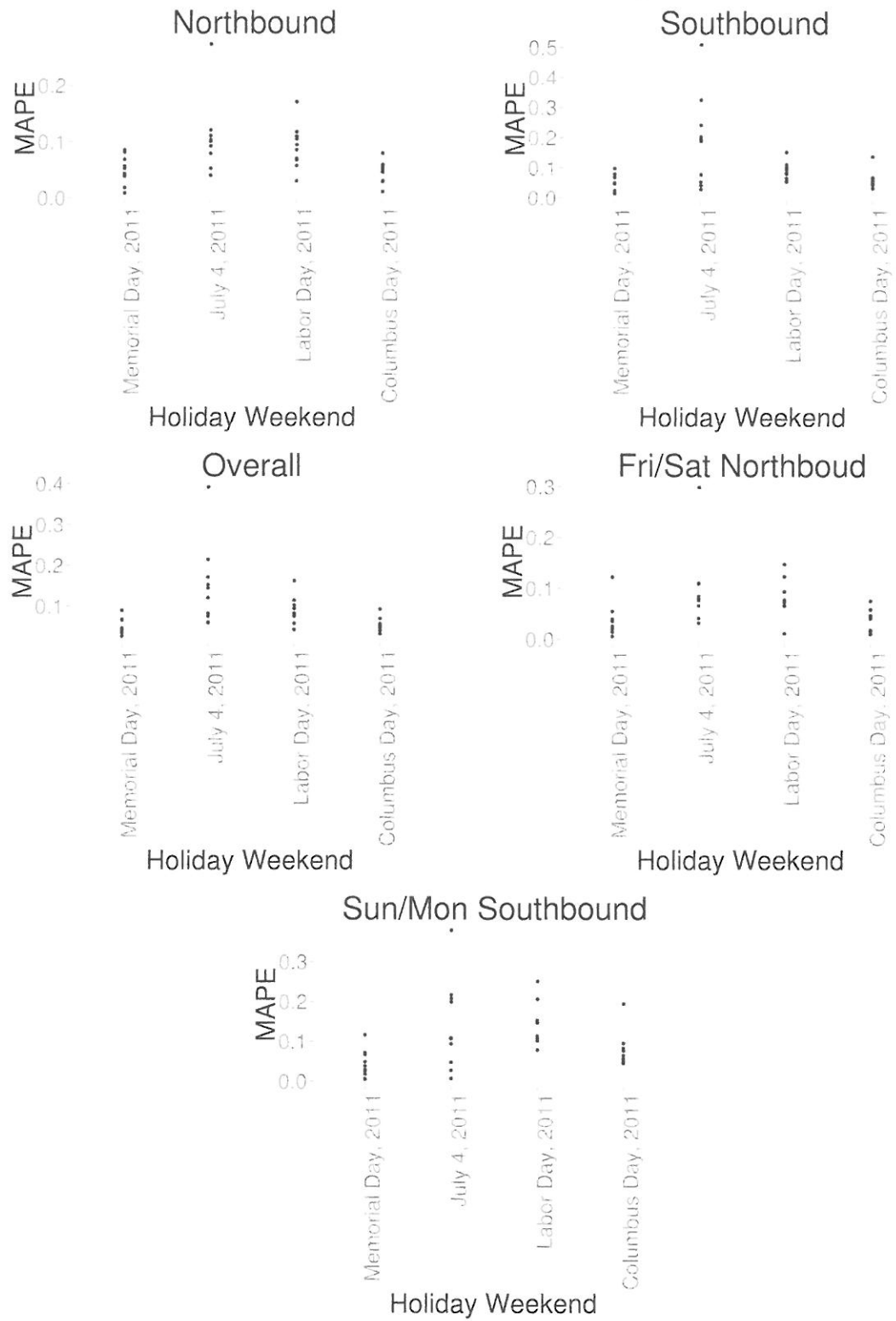
6.2 ARIMA vs. ANN

The second set of experiments was designed with the purpose of trying to determine which technique (ARIMA or ANN) is the superior forecasting technique for holiday traffic forecasting. This consisted of the development of two ARIMAX models: one for southbound and one for northbound. These two ARIMAX models were used to produce holdout forecasts of the four holiday weekends of interest in 2011. Using the results of the previous set of experiments, neural networks were created using 11 hidden nodes and trained to perform holdout forecasts on the four holiday weekends. For each holiday weekend, 10 neural networks were created and trained.

6.2.1 ANN MAPEs

The first set of figures below display the results of the neural network holdout forecasts. Each of the five figures below show the MAPEs of the holdout forecasts of the four holiday weekends. The 10 dots above each holiday weekend represent the 10 independent neural networks used to produce holdout forecasts.

Figure 6.5: Forecast MAPEs Using ANN



6.2.2 Actual Traffic Counts vs. Predictions

The next set of figures display the actual traffic counts, the predictions from the ARIMAX models, as well as the predictions from the ANNs. As there were 10 neural network runs for each holiday weekend, the neural network predictions shown in the figures below are the averages of all 10.

Figure 6.6: Predictions Vs. Actuals: Memorial Day

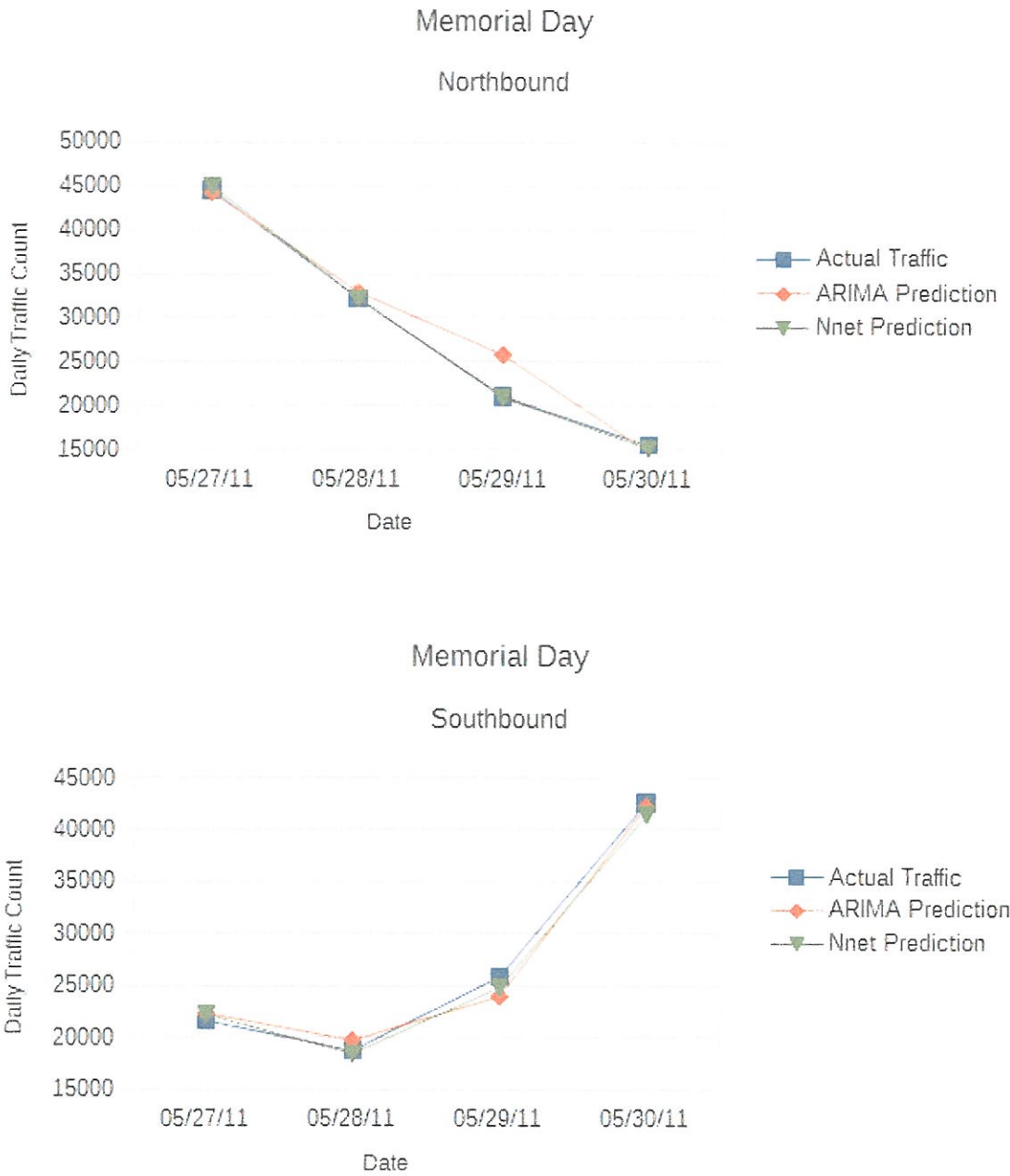


Figure 6.7: Predictions Vs. Actuals: Fourth of July

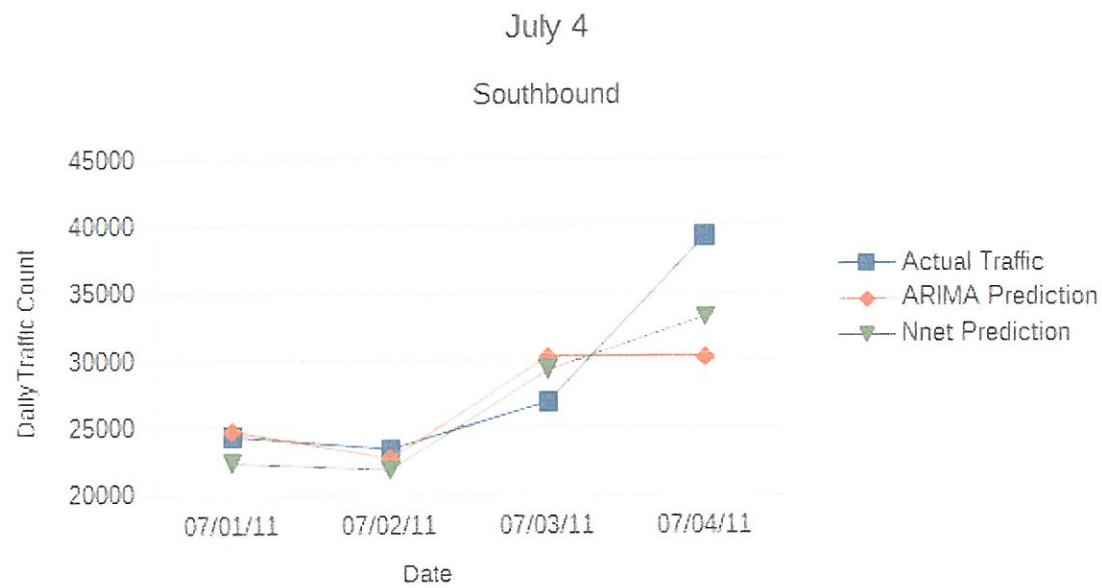
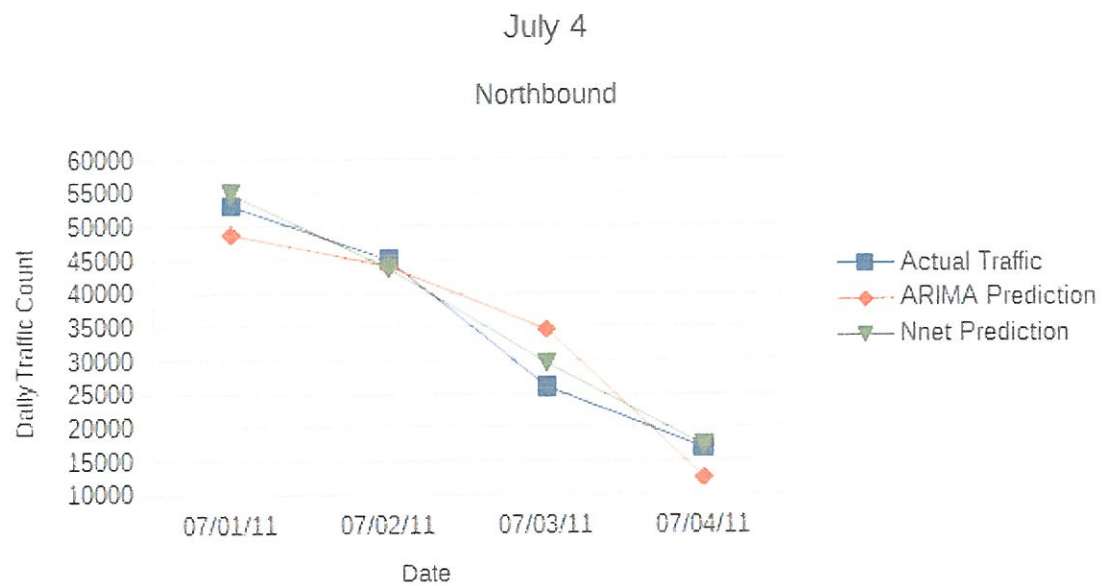


Figure 6.8: Predictions Vs. Actuals: Labor Day

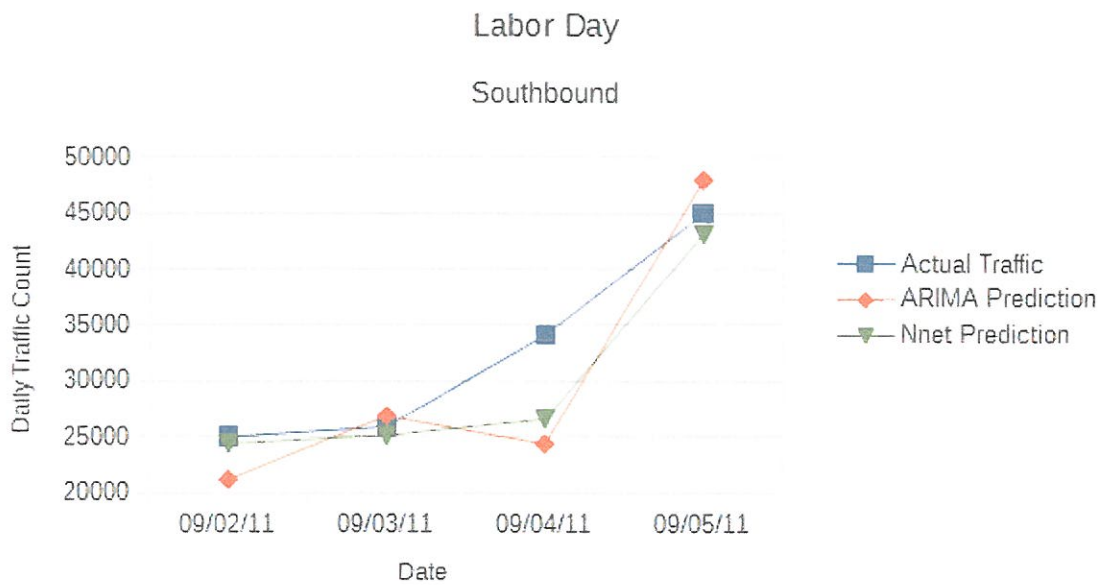
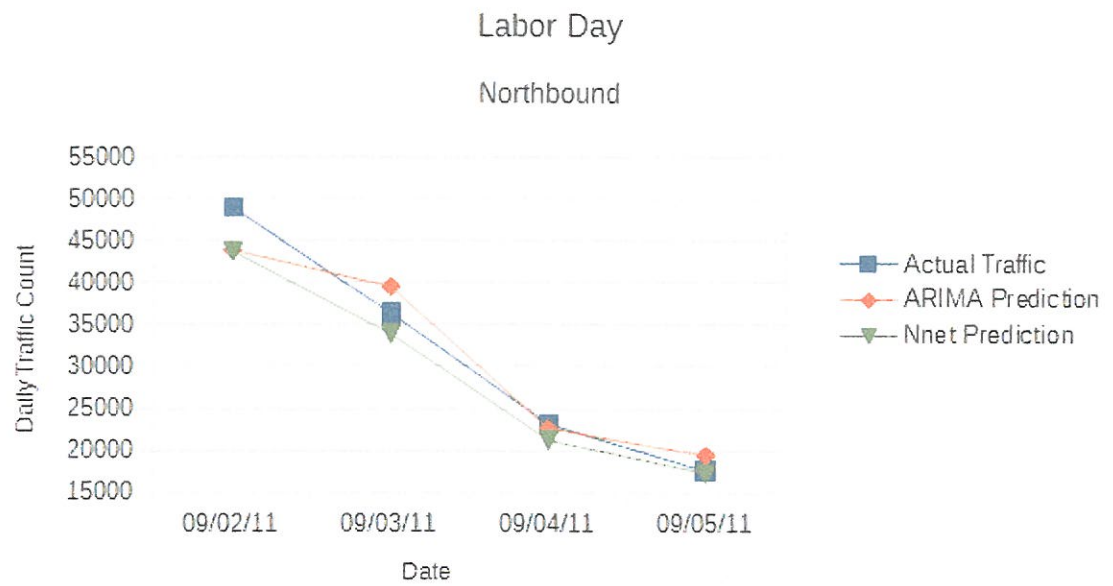
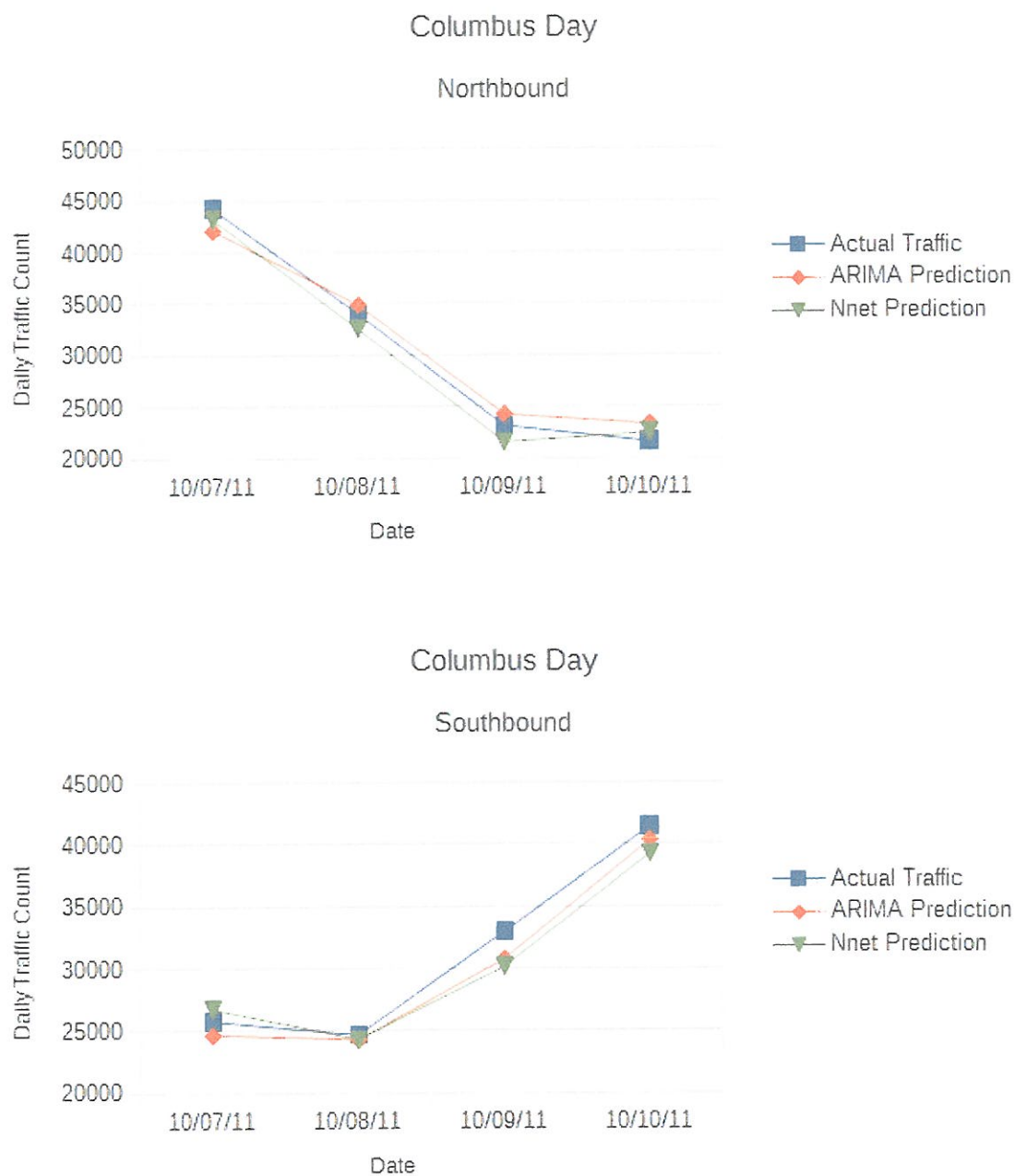


Figure 6.9: Predictions Vs. Actuals: Columbus Day



6.2.3 ARIMA MAPEs vs. ANN MAPEs

The following tables display the relevant MAPEs for the four forecasts in 2011 from both the ARIMA models and the ANNs. The ANN MAPEs are calculated using the average forecast of the 10 neural networks.

Table 6.1: Forecast MAPEs: Memorial Day

Memorial Day, 2011	ARIMA	ANN
Northbound MAPE	7.46%	0.96%
Southbound MAPE	4.17%	2.77%
Overall MAPE	5.82%	1.87%
Fri/Sat Northbound MAPE	1.44%	0.55%
Sun/Mon Southbound MAPE	4.00%	3.33%

Table 6.2: Forecast MAPEs: Fourth of July

July Fourth, 2011	ARIMA	ANN
Northbound MAPE	17.23%	5.38%
Southbound MAPE	10.01%	9.73%
Overall MAPE	13.62%	7.56%
Fri/Sat Northbound MAPE	5.08%	3.08%
Sun/Mon Southbound MAPE	17.63%	12.04%

Table 6.3: Forecast MAPEs: Labor Day

Labor Day, 2011	ARIMA	ANN
Northbound MAPE	7.93%	6.79%
Southbound MAPE	13.52%	7.86%
Overall MAPE	10.73%	7.33%
Fri/Sat Northbound MAPE	9.52%	8.70%
Sun/Mon Southbound MAPE	17.57%	13.05%

Table 6.4: Forecast MAPEs: Columbus Day

Columbus Day, 2011	ARIMA	ANN
Northbound MAPE	5.01%	4.35%
Southbound MAPE	3.72%	4.78%
Overall MAPE	4.37%	4.56%
Fri/Sat Northbound MAPE	3.69%	3.32%
Sun/Mon Southbound MAPE	4.63%	6.91%

CHAPTER VII

Discussion

In this chapter, I will first discuss the results of the set of experiments that were designed to determine the correct number of hidden nodes to use in the neural networks in this thesis. I will then discuss the forecasts of both the ARIMA and ANN models, as well as the resulting MAPEs of those holdout forecasts. Finally, I will discuss the relative advantages and disadvantages of both approaches with regards to things other than just forecast accuracy.

7.1 Number of Hidden Neurons

The first set of experiments were designed to determine what the ideal number of hidden nodes was for the neural networks to produce the most accurate holdout forecasts possible. As stated previously, there isn't a good way to empirically determine what this ideal number is. The most reliable way is to simply resort to trial and error. In this case, that means constructing and training neural networks with a range of numbers of hidden nodes.

Ideally, I would have been able to try every single number of possible hidden nodes from the number of output nodes (2) to the number of input nodes (168). Also under ideal circumstances, I would have been able to do a thousand runs with each number of hidden nodes to get a very good idea of how each number of hidden nodes does

on average. As is often the case, I was unable to perform as many runs as I would have liked to due to computing constraints. Specifically, I only had one computer to work with and a limited amount of RAM. This meant that I could not run many runs in parallel, or run them for a large number of trianing iterations. As it was, it took several days to do the runs that are in this thesis. Because of how time consuming the runs were, I decided to limit the number of hidden nodes to the range 2-84, and only perform 10 iterations of each number. Due to previous informal testing, I knew that the ideal number of hidden nodes was unlikely to be greater than about 30. I also knew that the ideal number of hidden nodes would be on the lower end and therefore decided to sample the number of hidden nodes using a logarithmic distribution. Thus, I ended up doing ten runs using each of the ten munbers of hidden neurons that I chose (2, 3, 5, 7, 11, 16, 24, 37, 55, 84). I believe that these runs were sufficient to make the decision of the ideal number of hidden nodes.

From examining the figures in Section 6.1 and starting on page 32, it is evident that the ideal number of hidden nodes is approximately 11. In most cases, the neural networks using less than 11 hidden neuron produce significantly larger and more varied MAPES then those networks with 11 hidden neurons. It is also apparent that those network containing more hidden neurons than 11 consistently produced larger and more varied MAPES as well.

More hidden neurons typically equates to more computational power for the network. Because of this, it is relatively easy to understand why a network with few hiddden neurons would perform poorly. It is because the network does not have enough power to find the hidden relationships among all of the different variables. It is slightly more difficult to understand why networks with greater and greater numbers of hidden neurons don't continue to perform better and better. This gets back to the issue of overtraining as discussed previously. Because a network with a great many hidden neurons has a lot of computational power, the network is able to figure

out the training data very well. The network becomes so good at predicting the training data, that it no longer has the flexibility to deal with new data in a reasonable way. The end result is a network that performs very well on data that it has already seen, but fails completely on anything that it has not seen before. This may be why the networks with 24, 37, and 55 hidden neurons didn't perform nearly as well as the networks with fewer neurons even though all of the networks were trained until they had the same performance on the training data.

Ultimately, I decided to go with 11 hidden neurons. In those experiments, 7, 11, and 16 all did fairly well, and since 11 is in the middle, that was the logical choice.

7.2 ARIMA vs. ANN

With the appropriate number of hidden neurons determined to be 11, it was time to produce holdout forecasts using both the ARIMA and ANN models and compare their performance. I chose to use the four holiday weekends in 2011 to do the holdout forecasts. The northbound ARIMA model produced the northbound forecasts and the southbound ARIMA model produced the southbound forecasts. The ANN forecasts were produced using 40 different neural networks. For each of the four holiday weekends, 10 neural networks were created and trained. In Section 6.2.1 starting on page 37, the various maps are displayed for the individual neural network runs for the four different holiday periods.

As is evident in the "Overall" figure on page 37, the July fourth holiday weekend was the most difficult to forecast. This is to be expected as the July fourth weekend often sees the highest volume of traffic of the entire year, as well as the fact that July fourth does not always fall on the same day. The other three holidays always occur on Mondays, but July fourth can occur on any day of the week. Depending on the day of the week that it does happen to fall on, the traffic patterns can change dramatically from year to year, whereas the other holidays exhibit much more consistent traffic

patterns from year to year.

It's also apparent from these figures that the forecasts for the northbound traffic on Fridays and Saturdays is usually much easier to forecast than the southbound traffic on Sundays and Mondays. I believe that the reason for this is simply that forecasts degrade as you get farther and farther out. For example, if it is Friday and you are asked to forecast the weather for tomorrow, you can say with a reasonable amount of confidence that it will most likely be somewhat similar to today. However, if you are asked to forecast the weather a week away, it is not nearly as easy to do so because weather patterns can change dramatically in a week's time. The same is true when forecasting traffic counts.

In the next set of figures in Section 6.2.2 and starting on page 39, the actual traffic counts are shown along with the predictions of the ARIMA and ANN models. The ANN predictions are the result of averaging the 10 predictions for each day of each holiday. These plots aren't terribly informative, but are useful in creating a frame of reference. MAPES are more useful in determining which forecast is more accurate, especially when they are close, but they aren't very good at describing the traffic patterns of the weekend or imparting a sense of what actually happened during the period in question.

These plots of the actual traffic and predictions do show a few things. The first is that there is definitely a pattern that a lot of people travel north at the start of the holiday weekend, and then a lot of people travel south towards the end of a holiday weekend. This may seem somewhat obvious, but it's interesting to see how pronounced the pattern is. The other thing to notice is that when there is a large discrepancy between the predictions and actuals, that the neural network appears to do usually do better. This is evident in the July 4 plots on page 40.

The results that really tell the story of how the ARIMA models compared to the ANNs are the tables that begin on page 43. These four tables provide the head to

head comparisons between the ARIMA models and the ANN models. As is evident in the tables, the ANN has a lower error than the ARIMA in all but a few instances during the Columbus day holiday weekend. Many of the differences in MAPEs are not huge, but some are fairly significant. Particularly during the July fourth weekend, the ANN does quite a bit better than the ARIMA models.

Overall, I would not say that the neural networks significantly outperformed the ARIMA models. I do think that these results indicate that ANNs are entirely viable for traffic forecasting and do have the ability to perform better than ARIMA.

7.3 Other Considerations

Despite the neural networks outperforming the ARIMA in the majority of the forecasts, it is still not a black and white issue when trying to determine which to use. There are always advantages and disadvantages of any methodology that don't necessarily involve its performance.

For the ARIMA, the advantages are many. ARIMA models are tried and true. There is a great deal of literature regarding their development and use. They also have the advantage of being much more transparent. Once an ARIMA model is built, it can be examined to see which variables are significant and which are not useful. It's much easier to get a better sense of the underlying relationships between the variables with an ARIMA model. ARIMA models also don't require much in the way of computational resources. It's true that large models can take a bit of time to run, but nothing like a large neural network.

ARIMA models do have their disadvantages however. To make a statistically valid ARIMA model, it is necessary to know a great deal of statistics. There are many traps that a modeler can fall into along the way that will invalidate the entire model. ARIMA models are also limited in the type of relationships that they can model. For instance, it was necessary to build two different ARIMA models; one

for southbound, and one for northbound. This is because an ARIMA model cannot handle multiple dependent variables. Of course, there are other statistical models that can, but ARIMA cannot. Because of this, it is impossible for the ARIMA model to gain certain insights. For example, as seen in some of the previous figures, the northbound traffic at the beginning of the holiday weekend may have an impact on the southbound traffic at the end of a weekend. If the ARIMA model was predicting a huge number of northbound traffic, that information may be of use to the southbound model, however, with the two separate models, that information sharing cannot occur.

Neural networks have a number of advantages. They are quite flexible in the kinds of things that they can model. In almost an instance where you have a set of data that you are trying to use to predict another set of data, you can use a neural network. This is a nice advantage to have because they are almost a “one-stop shop”, whereas when using statistical models, you must know the right type of model to choose for your particular situation. They also have the advantage of being automated to a large degree. For instance, you don’t need to know which variables are significant or not, the neural network should be able to sort that out on its own. Neural networks can also figure out relationships between variables that one might never think of, whereas often those possible relationships need to be specified in a statistical model.

Of course, neural networks also have disadvantages as well. They use a significant amount of computational resources. This proved to be a hurdle in this thesis. I would have liked to use the full set of variables that were available and to do many more runs of the neural networks, but could not due to time and RAM limitations. Neural networks are also “black boxes”. This means that it is almost impossible to look at a trained neural network and use it to gain an understanding of the relationships between the different variables. It is possible to do so to a limited extent, but there are often so many different connections between the neurons, that it is quite difficult to gain any useful insights.

At the end of the day, both modeling techniques have pros and cons, and they both do a reasonable job of forecasting traffic counts. The decision on whether to use one or the other will often come down to computational limitations, as well as whether the modeler has expertise with statistical models or neural networks. Purely based on my own opinion, I would say that neural networks have greater potential for more accurate forecasts than ARIMA models, simply because of their flexibility.

CHAPTER VIII

Future Work

At the end of any research project, there are always additional lines of inquiry that have been left unexplored. This thesis is no exception. There are several things that would have been interesting to try out, and potentially profitable, but that would have been beyond the scope of this thesis.

The first thing that would have been nice to explore would have been the “holiday multiplier”. Originally I had outlined a number of experiments that would have examined how effective this was as well as what the right number to use was. Ultimately I had to cut these experiments from this thesis because of time constraints. I ended up only doing a few informal experiments in order to figure out that the holiday multiplier was a very useful idea that improved the performance of the neural network dramatically when producing forecasts for holiday weekends.

Another avenue of exploration would be to try out additional variables and lags of variables. Due to computational constraints, I had to cut down the number of input variables dramatically. If I could get a hold of a larger computer or cluster, this would be one of the next logical steps to take.

A more substantial extension to this work would be to try a different type of neural network. One that I think would be particularly well suited to this problem would be a recurrent neural network. This is a neural network that has connections between

neurons both forward and backward. The advantage of such connections is that they can internally act as a type of lag. This would potentially mean that the lagged versions of variables would not have to be explicitly input. The neural network could “remember” past values of the variables and decide for itself which lags were important. The trouble with using a neural network with recurrent connections is that you cannot use a backpropagation training algorithm. Typically the only way to train one of these networks is to use an evolutionary algorithm, which takes significantly more expertise than the standard feed-forward network used in this thesis.

The final thing that would be interesting to try would be to combine an ARIMA with a neural network into a hybrid model. Either the ARIMA model could go first on the data, and then the neural network could clean up the errors, or vice-versa. This hybrid model would most likely outperform either model on its own.

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] A.G. Bakirtzis, V. Petridis, S. J. Kiartzis, and M. C. Alexiadis. A neural network short term load forecasting model for the greek power system. *IEEE Transactions on Power Systems*, 1996.
- [2] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., New York, NY, USA, 1995.
- [3] Maria Castellano-Mendez, Wenceslao Gonzalez-Manteiga, Manuel Febrero-Bande, J. M. Prada-Sanchez, and Roman Lozano-Calderon. Modelling of the monthly and daily behavior of the runoff of the xallas river using box-jenkins and neural networks methods. *Journal of Hydrology*, 2004.
- [4] Roland Chrobok, Andreas Pottmeier, Sigurdur F. Hafstein, and Michael Schrechenberg. Traffic forecast in large scale freeway networks. *International Journal of Bifurcation and Chaos*, 2004.
- [5] Mark S. Dougherty and Mark R. Cobbett. Short-term inter-urban traffic forecasts using neural networks. *International Journal of Forecasting*, 1997.
- [6] Sherif Ishak and Ciprian Alecsandru. Optimizing traffic prediction performance of neural networks under various topological, input, and traffic conditions. *Transportation Research Board*, 2003.
- [7] L. Li, W.-H. Lin, and H. Liu. Type-2 fuzzy logic approach for short term traffic forecasting. *IEE Proceedings of Intelligent Transportation Systems*, 2006.

- [8] Zhaobin Liu, Satish Sharma, and Sandeep Datta. Imputation of missing traffic data during holiday periods. *Transportation Planning and Technology*, 2008.
- [9] Anurag More and M.C. Deo. Forecasting wind with neural networks. *Marine Structures*, 2003.
- [10] Robert Nau. Introduction to arima, November 2012.
- [11] University of Southern Maine. Overview, 2012.
- [12] Victor R. Prybutok, Junsu Yi, and David Mitchell. Comparison of neural network models with arima and regression models for prediction of houston’s daily maximum ozone concentrations. *European Journal of Operational Research*, 2000.
- [13] Anthony Stathopoulos and Matthew G. Karlaftis. A multivariate state space approach for urban traffic flow modeling and prediction. *Transportation Research Part C*, 2003.
- [14] Shiliang Sun, Changshui Zhang, and Guoqiang Yu. A bayesian network approach to traffic flow forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 2006.
- [15] Mascha Van Der Voort, Mark Dougherty, and Susan Watson. Combining kohonen maps with arima time series models to forecast traffic flow. *Transportation Research*, 1996.
- [16] Hoogendoorn S.P. van Zuylen H.J. van Lint, J. W. C. Accurate freeway travel time prediction with state-space neural networks under missing data. *Transportation Research Part C*, 2005.

- [17] Billy M. Williams and Lester A. Hoel. Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results. *Journal of Transportation Engineering*, 2003.
- [18] Guoqiang Yu and Changshui Zhang. Switching arima model based forecasting for traffic flow. in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*
- [19] G. P. Zhang and Min Qi. Neural network forecasting for seasonal and trend time series. *European Journal of Operation Research*, 2005.
- [20] Weizhong Zheng, Der-Horng Lee, and Qixin Shi. Short-term freeway traffic flow prediction: Bayesian combined neural network approach. *Journal of Transportation Engineering*, 2006.