Summer 2017

Wind Turbine Noise and Wind Speed Prediction

Tyler H. Blanchard

WIND TURBINE NOISE AND WIND SPEED PREDICTION

by

TYLER BLANCHARD

(Under the Direction of Biswanath Samanta)

ABSTRACT

In order to meet the US Department of Energy projected target of 35% of US energy coming from wind by 2050, there is a strong need to study the management and development of wind turbine technology and its impact on human health, wildlife and environment. The prediction of wind turbine noise and its propagation is very critical to study the impacts of wind turbine noise for long term adoption and acceptance by neighboring communities. The prediction of wind speed is critical in the assessment of feasibility of a potential wind turbine site. This work presents a study on prediction of wind turbine noise and wind speed using a noise propagation model and artificial neural network (ANN) methods respectively. The noise propagation model utilized Openwind, a software package used for wind project design and optimization, to predict a noise map based on inputs acquired from a potential wind energy demonstration site in Georgia. The resultant noise of the wind turbines and the ambient surroundings were predicted in the neighborhood for different scenarios. The nonlinear autoregressive (NAR) neural network and nonlinear autoregressive neural network with exogenous inputs (NARX) were used to predict wind speed utilizing one year of hourly weather data from four locations around the US to train, validate, and test these networks. This study optimized both neural network configurations and it was demonstrated that both models were suitable for wind speed prediction. Both models were implemented for single-step and multi-step ahead prediction of wind speed for all four locations and results were compared. NARX model gave better prediction performance than NAR model and the difference was statistically significant.

INDEX WORDS: Artificial neural network, Forecasting, Multi-step ahead, NAR networks, NARX networks, Noise maps, Single step ahead, Time series prediction, Wind speed prediction, Wind turbine noise prediction

WIND TURBINE NOISE AND WIND SPEED PREDICTION

by

TYLER BLANCHARD

B.S., Georgia Southern University, 2014

A Thesis Submitted to the Graduate Faculty of Georgia Southern University in Partial

Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

STATESBORO, GEORGIA

WIND TURBINE NOISE AND WIND SPEED PREDICTION

by

TYLER BLANCHARD

Major Professor: Biswanath Samanta

Committee: David Calamas

Minchul Shin

Electronic Version Approved:

July 2017

DEDICATION

To

My family for their continued support.

ACKNOWLEDEMENTS

I would first like to thank my advisor and Professor here at Georgia Southern University, Dr. Biswanath Samanta, for offering the opportunity to work alongside him and other students to conduct research, as well as for his patient guidance, enthusiastic encouragement and useful critiques of research work. I would like to acknowledge the support provided by my family and my significant other during the preparation of this paper.

ABREVIATIONS

ANFIS: Adaptive network based fuzzy interference systems

ANN: Artificial Neural Network

ARIMA: Autoregressive moving average model

BNDA: Bismarck North Dakota Municipal Airport

FFT: Fast Fourier transform

GIS: Geographic information system

ICT: Information communications technology

ISO: International Standardization Organization

JFK: John F. Kennedy International Airport

LMBP: Levenberg-Marquardt Back Propagation

NAR: Nonlinear autoregressive neural network

NARX: Nonlinear autoregressive neural network with exogenous inputs

NCDC: National Climatic Data Center

NN: Neural Network

NWP: Numerical weather prediction

RNN: Recurrent neural networks

SIA: Savannah International Airport

WPMS: Wind power management system

TABLE OF CONTENTS

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

Wind energy has become the world's fastest growing renewable energy source due to its environmentally friendliness and economic viability. However, due to the various conditions at locations for wind energy power generation, accurate information of the dynamic nature of turbines and the wind that drives these turbines is needed for wind farm siting, as well as operations and management of the wind energy conversion systems. Forecasting and prediction methods are based on the available information and the application.

Studies show that the prediction of wind speed, and thus wind power, are critical for the operation of a wind energy conversion site in order to operate at optimal levels. A forecast gives wind farm management the tools to balance maximum reliability with minimal operating costs. Wind forecasting in the order of seconds to minutes are normally applicable to the control of a wind turbine. Forecasting in the order of hours addresses the problem of scheduling with a power system. Forecasts that predict in the range of days address the problem of maintenance and resource planning.

In order to meet the US Department of Energy projected target of 35% of US energy coming from wind by 2050, there is a strong need to look into the aspects of potential environmental impacts of wind energy facilities on human health, wildlife and environment (US Department of Energy 2015, Saavedra and Samanta 2015). Understanding and managing the impact of wind turbine technology on the environment is a critical factor for successful long term adoption and acceptance of the technology by the neighboring community and mitigating potential stress on impacted wildlife. The issues in wind turbines include environmental impact causing concern for humans and wildlife, noise and vibration caused by operation, and visual and aesthetic impacts.

Noise generated by turbines can cause annoyance, sleep disturbance, as well as self-reported instances of stress and quality of life issues (Health Canada, 2014, Michaud, 2013, Saavedra and Samanta 2015). In the selection of any potential wind energy generation site prediction of possible noise generated by wind turbines and noise propagation is very critical to make sure the predicted noise levels are within the acceptable levels to avoid any adverse opinion about the possible noise effects in the neighboring communities.

1.1 Time series forecasting

Time series are a data set with data points indexed in chorological order. Typically time series are taken at equal sequential point in time, known as time steps. Some of examples of time series data includes: daily stock values, energy consumptions values, heights of ocean tides, and hourly temperature measurements. Time series prediction or forecasting uses a model to predict future values from past time series data. A stochastic model will illustrate the fact that data points close together will have more of an effect on each other than data points that are far apart. Machine learning or computational intelligence have been used in recent years to conduct time series forecasting of future data based on historical data (MacKay 2003).

1.2 Objectives and Scope of Present Work

There are two main hypotheses in the work performed: (1) if ambient noise was known at a certain location, sound levels could be predicted with the presence of wind turbines at various distances; and (2) if current and previous wind speed (and temperature) data was available, future wind speed values could be predicted, even in a nonlinear form, by using time series regression analysis techniques.

1.2.1 Wind Turbine Noise Prediction

A model of noise propagation from wind turbines was created. To test the hypothesis presented above ambient noise data was collected from a potential wind energy generation site. Using average wind speeds for the area a noise propagation model was applied in order to predict noise at various distances from the turbines. The distinct objectives of this part of the study were:

- Use sound level meters to collect ambient noise data at a potential wind turbine installation site.
- Develop a noise propagation model using open access software like Openwind®.
- Determine if wind turbine noise would be significant enough to disturb nearby residents based on measured and potential ambient noise levels

In this part of the study relevant techniques for noise propagation modeling were reviewed and learned using Openwind® software. Next sound level meters were used in order to collect ambient noise data. Local wind speed data was used as an input to create an average noise model for a potential wind turbine site. The noise model's data was included with ambient noise data to determine if wind turbine noise would become significant in relation to ambient noise and pose any noise related disturbance in the neighborhood.

1.2.2 Wind Speed Prediction

Two artificial neural networks were used in the prediction of wind speed. To test hypothesis presented above, hourly wind speed and temperature data were obtained from a national meteorological database and artificial neural network (ANN) prediction methods were applied in order to forecast wind speeds in the range of a few hours. The distinct objectives of this part of the study were:

- Develop a nonlinear autoregressive neural network (NAR) model using MATLAB neural network toolbox.

- Develop a nonlinear autoregressive neural network with exogenous inputs (NARX) model using MATLAB neural network toolbox.
- Develop a methodology to optimize the architecture of the two proposed neural networks.
- Compare the results of the two optimized neural networks.
- Implement the two optimized networks by predicting wind speed of several data sets one hour and five hours in the future.

In this part of the study relevant ANN techniques were reviewed and learned using MATLAB neural network toolbox. Next the hourly wind speed and temperature data were collected and normalized in order to have a fair comparison. The data was then used to optimize the two neural networks (NAR and NARX). Multiple runs of the optimization were conducted in order to ensure precision and accuracy. The optimized networks were used to predict wind speed one hour ahead as well as five hours ahead. The statistical significance of the difference of prediction performance of NAR and NARX was investigated.

1.3 Organization of Thesis

The rest of this thesis is organized into the following chapters. Chapter 2 includes the literature reviewed in the time the study was conducted. The first section contains wind turbine noise prediction with sub sections on Openwind®, SPreAD-GIS, calculating noise levels, and buildings and vegetation’s effect on noise propagation. The next section presents wind speed forecasting time frames with sub sections on classification of the time frames, immediate short term forecasting, short term forecasting and long term forecasting. After this forecasting methodology for immediate short term forecasting is presented with sub sections of neural networks (both biological and artificial) as well as a sub section of various other techniques. The final section of chapter 2 recounts various wind speed prediction studies.

Chapter 3 includes the research methodologies used in this study as well as the relevant data sets. The first section discusses the data sets used in the two parts of the thesis with subsection for noise propagation prediction as well as wind speed prediction. The next section presents the Openwind® noise propagation simulation with subsections on standards as well as the noise model. The final section details wind speed prediction with subsection on the NAR and NARX models used in the study.

Chapter 4 contains the results and discussion of both parts of the study. The first section discusses the wind turbine noise propagation prediction results with subsections on the methods used as well as the noise maps generated. The next section analyses the wind turbine noise maps and combines this with ambient noise data taken from the experimental location. The following section details the wind speed prediction results with subsections on the optimization of the network as well as implementation of the optimized networks into one-step ahead and multi-step ahead prediction.

Chapter 5 provides the conclusions of both parts of the thesis as well as the scope of future work. The objectives of the work and the results are summarized in order to ensure the objectives were met.

Appendix A and B will detail the code used in MATLAB 2014a to generate the NAR and NARX networks respectively.

CHAPTER 2

REVIEW OF THE RELATED LITERATURE

2.1 Wind Turbine Noise Prediction Methodologies

There are many different methods, approaches and procedures in order to produce a wind turbine noise propagation model. An easily understood example of turbine noise propagation can be observed in figure 2.1 below (GE 2012). This graphic compares sound levels at different distances from a wind turbine and equates them to various appliances, for instance at 100 meters wind turbine sound is about as loud as a midsize window air conditioner. This section gives an overview of two open-access software platforms along with the advantages and disadvantages. This section also reviews literature in the fields of sound attenuation due to vegetation and urban form.

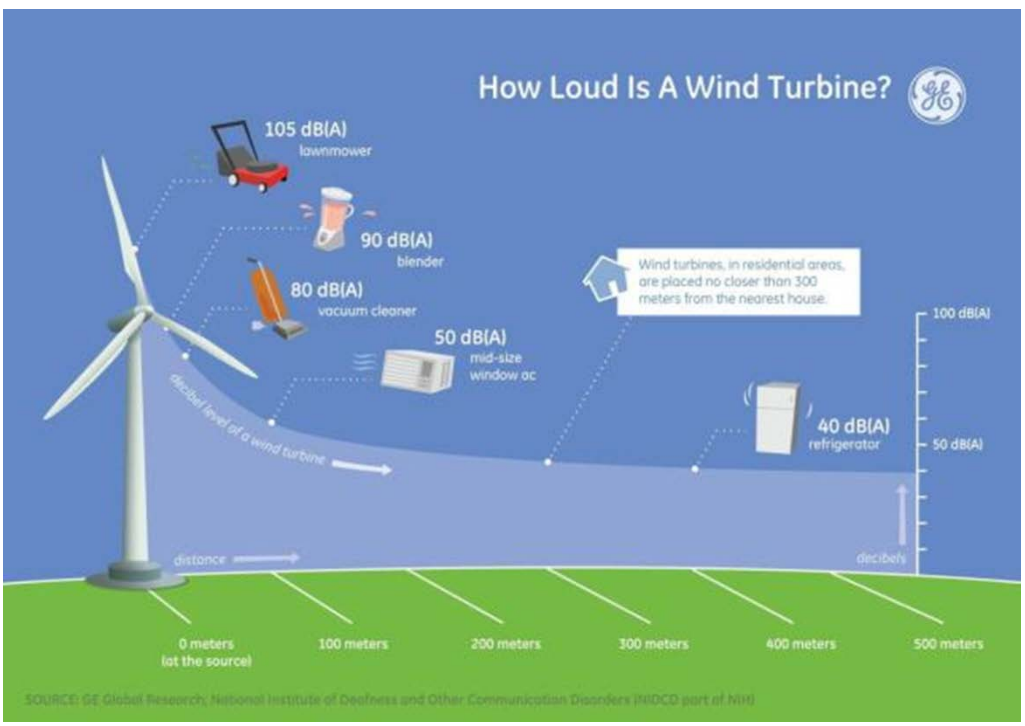


Figure 2.1 Average Turbine Noise levels at various distances (GE 2012)

2.1.1 Openwind®

Openwind® is an open source software package that is used for wind project design and optimization. It is a platform that more intuitively integrates GIS-based site information in order to promote industry collaboration and research in a flexible wind project design platform. Inputs for this software program include site selection and land constraints, wind resource and energy production potential, and turbines and operation characteristics. Site selection entails the construction of a wind resource grid defined within a GIS framework that will be the basis for generating data for the program. Resource and energy potential are based on hub height wind speed maps as well as meteorological data. The characteristic parameters of the turbines to be constructed are also taken into account in the model (Filippeli 2013, AWS Truepower 2014). The noise model in Openwind® is based on ISO 9613-2, which is the international standard for the propagation and attenuation of industrial noise (International Organization for Standardization 1996). The program makes several simplifying assumptions in the generation of a noise map including: all noises are treated as point sources, all propagation is assumed to be in the same direction as the wind, obstacles and blocking effect of terrain can be ignored due to wind turbines being classified as aerial sources (AWS Truepower 2014). This method of calculation results in a fairly conservative picture of wind noise propagation, however could have higher accuracy if all types of attenuation where taken into account.

Openwind® was chosen for this project due to its accessibility, better documentation and its flexibility of using publicly available Google Earth GIS data for coastal site at a potential wind energy demonstration site in Georgia.

2.1.2 SPreAD-GIS

SPreAD-GIS is an open-access software application and is implemented as a toolbox in ArcGIS software, a commercial geographic information system program. The main purpose of SPreAD-GIS is to model patterns of noise propagation caused by manmade sources in natural outdoor environments. This toolbox uses the data sets on land cover, topography, and weather condition from the GIS software to calculate noise propagation and excess noise above ambient conditions for the one third octave frequency bands around one or multiple sound sources (Reed et al. 2012). The use of GIS plays an important role in noise mapping as it can greatly improve the accuracy of results obtained from noise modeling. In the field of research it will be important for standardization in order to optimize quality and efficiency of noise effect studies. Standardization will also be important because results of different studies can only be combined or compared if the same parameters for noise exposure and the same analysis methods are used (Kluijver 2003).

SPreAD-GIS can be a more accurate program to model noise propagation due to the more capable GIS functions embedded within the ArcGIS software. The actual calculation for the noise propagation is the same as in Openwind® and uses ISO 9613-2; if elevation data and vegetation data can be obtained, it can be more thoroughly integrated with SPreAD-GIS than in Openwind® and thus making the model more accurate. SPreAD-GIS was not used in this study due to its need of a third party GIS data in order to create the model and a commercial program module to run it.

2.1.3 Calculating Noise Levels (Noise Prediction)

Noise levels at a receiver point can be calculated as opposed to being measured allowing noise models to be created. In some cases calculation is the preferred method even where noise measurement can be conducted. This can occur in cases such as: where levels are contaminated by high levels of background noise, where future levels need to be predicted, where noise reduction scenarios need to be compared, where noise contour maps need to be produced, and where there is limited access to a measurement position (Brüel & Kjær 2000). Outdoor environments provide a challenge in terms of noise calculation due to the lack of uniformity. Changing meteorological conditions can easily cause fluctuations in sound levels by 10-20 dB over time periods as short as a few minutes. The longer the transmission path the larger will be the fluctuations. Outdoor sound propagation is affected by many factors including: obstructions, terrain type, atmospheric conditions, metrological conditions, and source geometry and type (Lamancusa 2009, National Physical Lab 2006, Larrson and Ohlund 20014).

2.1.4 Vegetation’s Effect on Noise Propagation

While not considered in the approach taken in this study, vegetation with dense foliage can have a major effect on the propagation of sound outdoors. It has been found that ground attenuation and scattering accounts for the highest amount of sound reduction from vegetation (Aylor 1972). It has also been observed from studying tree belts effects on point source noise propagation, which is a negative logarithmic relationship between relative attenuation and the visibility, as well as a positive logarithmic relationship exists between relative attenuation and the width, length, or height of the tree belts (Fang and Ling 2003).

2.1.5 Urban Form’s Effect on Noise Propagation

Obstructions can attenuate noise greatly from noise generation sources; this can be observed by studies done on traffic noise. Barriers can effect sound attenuation based on the angle of the obstruction, the change in path length, the observer height, and the source height (Pamanikabud and Tansatcha 2003). It has been found that urban forms in historical areas with narrower roads, complex road networks, and a higher density of intersections lead to lower traffic noise and thus lower noise pollution (Tang and Wang 2007). Some urban areas have gone so far as to combine barriers with vegetation by introducing greenery on external building elements. This has had an even greater attenuating effect on long-distance noise propagation (Tang and Wang 2007).

2.2 Wind Speed Forecasting Time Frames

There are many studies that have focused on the improvement of wind speed prediction and forecasting techniques. A number of models have been employed on wind farms throughout the globe. The following section outlines existing methods and tools used in wind speed and wind power forecasting and prediction over time. Usually wind forecasting is focused on very short term prediction in terms of immediate time frame of minutes, to a short time frame of hours, to up to one to two days in the long term. (Wang 2011, Zhao 2011)

2.2.1 Classification of Wind Forecasting

Wind forecasting can be classified according to time scale or the methodology used. There are typically three accepted time frames that wind forecasting is attempting to predict.

- The immediate short term (8 hours ahead of present prediction)
- The short term (1 day ahead of present prediction)
- The long term (multiple days ahead of present normally no longer than 48 hours) (Zhao 2011)

There are also typically three different accepted methodologies for wind forecasting.

- The physical approach (deterministic)

  The physical method is based on numerical weather prediction (NWP) using weather forecast data like temperature, pressure, obstacles, and surface roughness.

Generally the wind speeds provided by local meteorological services are used to predict the wind speed in the near future for a wind farm that can be converted into wind power (Zhao 2011).

- The statistical approach

  The statistical method is based on a large amount of historical weather data without considering current meteorological conditions. At its core the statistical approach normally employs artificial intelligence (neural networks, neuro-fuzzy networks) and time series analysis approaches (Zhao 2011)

- The hybrid approach

  The hybrid approach combines physical methods and statistical method. This method is most commonly used in weather forecasts and time series analysis. (Zhao 2011)

2.2.2 Immediate short term forecasting

Models for the immediate short term time frame are generally based on statistical approaches, particularly ANN, because NWP is time consuming. Wind power management systems (WPMS) have been in use by the information and communication technology (ICT) environments of different grid operators and curators of large wind farms. WPMS uses an artificial neural network (ANN) which trained with a large amount of historical weather data. Furthermore, fuzzy logic and adaptive network based fuzzy interference systems (ANFIS) are very promising AI methods which each have their own advantages in forecasting wind power and speed. (Li Shi 2010, Zhao 2011)

2.2.3 Short term forecasting

Several tools have been developed for the wind speed/power prediction in the short term time frame including: WPPT, Prediktor, Zephyr, Ewind, WPFS, AWPPS, etc. These models have also been implemented in case studies in Spain, Denmark, Ireland, Greece, Germany, and France (Li Shi 2010, Zhao 2011).

2.2.4 Long term forecasting

Only a few studies have been done on long-term wind forecasting approaches, thus there are not many prediction tools for this time scale. In order to have a forecast that is greater than a few days in advance, a more complex model must be used, as simpler models cannot meet these requirements. Most wind power prediction tools provide forecasts for a time horizon of several days in advance and are typically based on numeric weather prediction. NWP data can be provided by national weather services or private weather data collection sites. This sets NWP as the main factor for long term forecasting in the future. (Li Shi 2010, Zhao 2011)

2.3 Immediate Short Term Wind Speed Forecasting Methodologies

2.3.1 Neural Networks

Neural networks (NN) are originally associated with human physiology but as time and technology has progressed NNs now also refer to something more artificial. Computational intelligence in the form of artificial neural networks (ANN) are being used more and more frequently in order to predict data through machine learning, similar to how the human mind functions (Hagan 1996). In order to understand ANNs biological NNs must be understood.

2.3.1.1 Biological Neural Networks

In the human brain neural networks consist of three basic building blocks: axon, soma, and dendrites. Dendrites receive inputs from neighboring neurons and then relay the message to soma. The soma congregates and processes the input signals. The axon then processes these input signals into output signals that can be relayed to the next neuron via the synapse (Hagan 1996).

2.3.1.2 Artificial Neural Networks

Information technology use neural networks in a system of hardware and/or software patterned after the biological counterparts found in the brain. There are a variety of applications for ANNs but are typically used for solving intricate signal processing or pattern recognition. ANNs have been used for speech to text software, handwriting recognition and classification, facial recognition, and weather prediction (MacKay 2003).

Artificial neural networks typically contain a high number of processors that function if parallel to each other and organized in tiers. The first step takes in raw input information which could be compared to the eyes and optic nerves in humans each following layer receives an input from the output side of the previous tier- similar to how a neuron far away from the optic nerve receives signals from neurons close to it. The very last tier in the network produces a final output (MacKay 2003).

Each node of network has its own set of rules that it has been programed with or has learned. Each layer or tier is highly connected meaning each node in tier t will be connected to many other nodes in tier $t-1$ (its inputs) as well as nodes in tier $t+1$ (t provides inputs for $t+1$). There can be multiple nodes in the output layer or just one from which the answer can be read (Mackay 2003).

ANNs are noteworthy for being adaptive, this refers to their ability to learn from an initial training session and as subsequent runs are performed more information is learned. ANNs are normally given an initial training on large quantities of data. Training gives the network and input and then tells the network what the output should be. The basic and common learning method is weighting of the inputs. Weighting the inputs is how a node determines the importance of an input from the previous node. If a right answer is obtained that input is weighted higher (Mackay 2003).

Each node makes rules based on its inputs from the previous layer. Neural networks have several principles including: fuzzy logic, Bayesian methods, as well as genetic algorithms. Nodes can have initial conditioning including basic rules and relationships about the data being modeled. Having rules before the training stage can make training faster, however it can also create built in assumptions about the problem that may end up being unsupportive and unrelated or even inappropriate or incorrect. This makes the decision process of what to include (if anything) before training critical. ANNs can be defined by their depth as well, or how many layers (referred to as hidden layers) or neurons/nodes (contained in each hidden layer) they have between the input and the output (MacKay 2003).

2.3.2 Various Techniques

Monafared et al. uses fuzzy logic and artificial neural networks for wind speed forecasting and was shown to outperform the previous methods of the time. It was able to do this because fuzzy methods do not have as much of a rule based initialization as well as its adaptive nature to learn as it is estimating the wind speed. This result was able to outperform with less computational time (Monafared et al. 2009).

Li and Shi used an ANN model in order to estimate wind speed one hour in advance based on wind data from two different locations in North Dakota. This study showed that input data, learning rates, and the mechanisms of the model can affect the accuracy of the forecast. These methods had performance improvements of about 20% according to some metrics. Difficulties arose in this study due to instability problems concerning merging forecasts from other ANN models (Li and Shi 2010). ANN models from the MATLAB toolbox were applied by Mabel and Fernandez in order to estimate wind data in Muppandal, Tamil, India. This study used data from 2002-2005 from seven different locations, and the results were very similar to the actual data (Mabel and Fernandez 2008). ANN methods have been employed in Nigeria where there were no monitoring systems to great effect as well (Fadare 2010).

Mohandes et al. employed the ANFIS technique to predict wind speeds at higher elevations using data collected from lower elevations. This study was able to obtain mean absolute error of 3%, lending to the dependability of the ANFIS technique (Mohandes 2011).

2.4 Wind Speed Prediction

Wind energy conversion has been known to be a successful technique in order to generate power, particularly for isolated regions. Studies and practice have shown that it is greatly beneficial to forecast wind speed, and thus wind power, for the optimal operation of a wind turbine that has significant wind activity and penetration. An accurate forecast of wind speed allows a balance between maximizing reliability and minimizing operating costs. Wind speed is considered one of the most difficult meteorological parameters to forecast because of the interactions between other large weather forces such as temperature and pressure differences, topological surface conditions, as well as the Earth’s rotation. Wind forecasting in the order of seconds to minutes are normally applicable to the control of a wind turbine. Forecasting in the order of hours address the problem of scheduling with a power system. Forecasts that predict in the range of days address the problem of maintenance and resource planning.

Wind speed prediction and forecasting is representative of a time series regression problem. Time series regression problems have been researched plentifully, this leads to many methods and simulations relating to the field of time series prediction. Nagy (Nagy 2016) proposed a generalized additive tree ensemble method in order to predict wind power generation along with solar power generation. Camara et al. (Camara et al. 2016) used autoregressive moving average model (ARIMA) with neural network models in order to predict energy consumption. Torres et al. used the ARIMA model in order to predict hourly average wind speeds (Torres 2005). Doucoure et al. employs artificial wavelet neural network and multi resolution analysis in order to determine time series predictions using wind speed data (Doucoure et al. 2016). Haydari et al. present a time series electric load prediction model using neuro-fuzzy techniques (Haydari et al, 2007).

There have been a number of studies that have reported very good results and success in real world applications of using artificial neural networks (ANNs) (Doucoure et al. 2016, Macas 2016, Kiartzis 1995, Sanchez 2008, Jursa 2007). Experiments comparing ANNs to other techniques have shown that ANNs have often yielded superior outcomes (More 2003, Brand 2002, Fadare 2010, Kariniotakis et al. 1996, Tande and Landberg 1993). A reason that ANNs outperform other techniques is their aptitude in modelling non-linear data sets. Additionally ANNs, once they have been trained, are able to quickly predict with acceptable performance. Welch et al. compared a feedforward and feedback neural network design for short term wind speed prediction (Welch 2009). Sfetsos compared a variety of forecasting techniques including

ANNs for mean hourly wind time series (Sfetsos 2000). Gonzalez and Zamorreño predict short-term electricity through employing a feedback ANN.

There are many non-linear systems that appear in the real world, in other words systems that behave dynamically and are dependent on what state they are currently in. Recurrent neural networks (RNN), as well as nonlinear autoregressive (NAR) and nonlinear autoregressive neural networks with exogenous inputs (NARX) networks can prove useful in predicting this type of data (Cao 2012 and Mohanty et al. 2015). These ANNs can use time series data as dynamic input sets. Forecasting using ANNs is considered non-parametric, meaning that the way the time series is generated cannot be neglected. NAR network uses past data in the time series while RNNs do not as it has recurrent connections in its architecture.

This study is concerned with using NAR and NARX methods in order to predict wind speed. The objective of this study is to use a methodology that will optimize these two neural networks for the problem at hand, and thus determine if these methods are suitable for wind speed prediction. This study then determines if external data can be used to improve performance. One year of hourly weather data was used from several locations around the US in order to test these networks.

CHAPTER 3

RESEARCH METHODOLOGIES

3.1 Datasets

The noise propagation section of the thesis required a wind speed measurement at the experimental site as well as ambient noise measurements. The wind speed prediction required a data set of hourly average wind speed and dry bulb temperature readings over a year's time at multiple locations.

3.1.1 Wind Turbine Ambient Noise Dataset

A relatively less populated area in southern Georgia with the potential of a small wind energy demonstration site was chosen for ambient noise measurement. A sound level meter was obtained from leading international manufacturer of such systems in the field of vibration and noise (Bruel & Kjaer, 2013). The Hand Held Analyzer type 2270 is a "4th generation analyzer that has a dual channel measurement capability and performs a frequency analysis based on the Fast Fourier Transform (FFT) algorithm. The dual channel capabilities allow for use of both channels simultaneously to measure with two microphones, two accelerometers, or one of each." The devices were configured with all necessary software installed and tested. The transducers and analyzers were calibrated as per established procedure. The team visited the site and took measurements of the ambient noise level using Bruel & Kjaer portable noise level analyzer system. Measurements were taken at the proposed locations of wind turbines and the meteorological tower.

The microphone provided measurements for the ambient sounds in dBA. Ten two minute measurements were taken. Five locations were designated and two measurements for two minutes were taken at each location. The designated locations were chosen at the installation points for the meteorological tower and four turbines (fig 3.1).

Figure 3.1 measurement location of the noise dataset

The first two recordings were taken at the Met Tower Location. The second two recordings were taken at the Turbine 4 location. The third set of recordings was taken at the Turbine 3 location. The fourth set of recordings were taken at the Turbine 2 location. The fifth set of recordings were taken at the Turbine 1 location. These locations can be observed in the figure above.

3.1.2 Analysis of Initial Measurement Data

Five specific locations were chosen for data measurement and recording. The five locations are where the meteorological tower and four turbines are to be potentially installed. For each location two data sets were taken. Each data set contains 2 minutes of measurement, resulting in 20 minutes of logged data. Table 3.1 represents the mean value of the two recordings at each location, where LAeq is the A-weighted equivalent continuous noise level; LApeak is the A-weighted equivalent continuous noise level peak. It was a relatively windy day. The average wind speed varied in the range of 2.20-4.60 m/s, the minimum was in the range of 0.30-1.60 m/s and the maximum was in the range of 5.60-12.10 m/s. The ambient noise level was in the range of 42.5-52.1 dBA with peak values in the range of 47.1-65.7 dBA.

Table 3.1 Ambient measurements of installation site

	LAeq Avg (dBA)	Lapeak Avg (dBA)
Site A1	44.5	49
Site A2	44.1	47.1
Site B1	45.3	54.1
Site B2	44.5	53.5
Site C1	42.5	49.6
Site C2	44.2	52.1
Site D1	45.9	51.4
Site D2	45.7	49.8
Site E1	52.1	65.7
Site E2	43.6	52.1
Max	52.1	65.7
Min	42.5	47.1
Average	45.2	52.4

To distinguish between the different decibels measurements their definitions are listed as follows:

- LA_{eq} is the A-weighted equivalent continuous sound level.
- LA_{peak} is the A-weighted maximum (peak) sound level.

3.1.3 Wind Speed Prediction Data Set

Historical weather data from Savannah International Airport (GA), the Bismarck Municipal Airport (ND), Logan International Airport (MA), and the John F. Kennedy International Airport (NY) were obtained through the National Climatic Data Center climate data online (NCDC CDO 2010). The data included hourly mean wind speed in MPH and hourly dry bulb temperature in degrees Fahrenheit from January 1st 2010 until December 31st 2010. The latter three airports are considered to have some of the worst year round weather according to the NCDC. The data was normalized to be in a range between zero and one in order to compare results between different data sets and prevent local maxima from skewing results. The data was normalized using the equation 3.1 below. A sample set of the normalized data from Logan International Airport in Boston can be seen in figure 3.2 below.

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \tag{3.1}$$

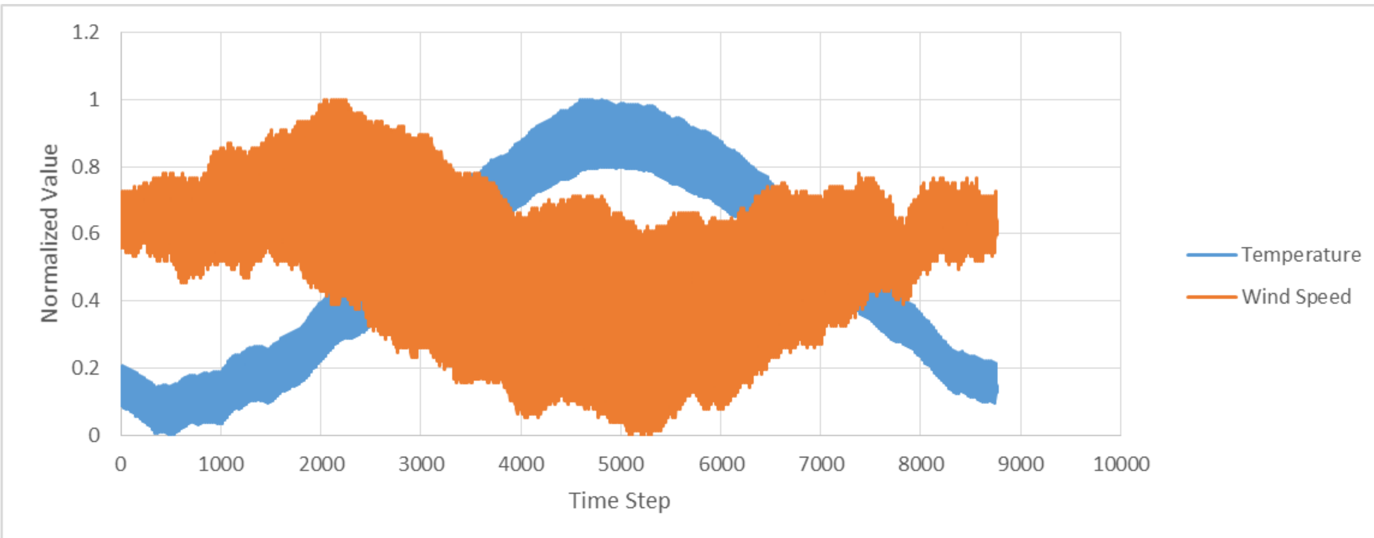


Figure 3.2 Temperature and wind speed time series over one hour time steps (normalized Boston dataset *(NCDC CDO 2010)*).

Using the neural network time series tool in MATLAB mean wind speed was predicted one step ahead, meaning the prediction occurred one hour ahead of the current time step. NAR and NARX methods were used in order to determine the most accurate method for forecasting wind speed.

Seventy percent of the data was used for training with the Levenberg Marquardt back propagation (LMBP) learning algorithm. Fifteen percent of the data was used for validation. Validation is used to measure network generalization, and to stop the training when generalization does not improve any more. Generalization stops improving as indicated by an increase in the mean square error of the validation samples. The remaining fifteen percent is used for testing. Testing has no effect on the training phase and is used to independently evaluate the network performance during and after training.

3.2 Openwind® Noise Propagation Simulation

3.2.1 Standards

ISO 9613-2 specifies an engineering method for calculating the attenuation of sound during propagation outdoors and is used by Openwind®® to predict the sound propagation of wind turbines in terms of simple A-weighted sound pressure levels (AWS TruePower 2014, International Organization for Standardization 1996). This is used to predict the levels of environmental noise at a distance from a source. The method contained in ISO 9613-2 predicts the equivalent continuous A-weighted sound pressure level under metrological conditions favorable to propagation.

Under this standard, several assumptions must be made. All noise sources are treated as point sources; all noise propagation is assumed to be in the same direction as the wind; atmospheric conditions are assumed favorable to noise propagation; and wind speeds that are between 3 and 11 meters above ground level are assumed to be between 1 and 5 m/s. ISO 9613-2 considers several types of attenuation including: atmospheric, geometric spreading and ground effect (porosity) (International Organization for Standardization, 1996).

ISO 9613-2 introduces basic equations used by Openwind®® to predict the attenuations of noise outdoors in community environments at a distance from a variety of sources of known sound emission. These conditions are for downwind sound propagation under moderate ground based temperature inversion (temperature rises as altitude increases) such as that occurs at night. Temperature inversion over water surfaces will not be considered accurate as it may result in higher sound pressure levels than predicted. This method is applicable in practice to a great variety of noise sources and environments.

$L_{fT}(DW)$ is the equivalent continuous downwind octave-band sound pressure level at a receiver location. This is calculated for each point source, and its image sources, and for the eight octave bands with nominal midband frequencies from 63 HZ to 8 kHz (International Organization for Standardization 1996).

$$L_{fT}(DW) = L_W + D_c - A \quad \text{… (3.2)}$$

Where

L_W is the octave-band sound power level, in decibels, produced by the point source relative to a reference sound power of one picowatt (1 pW)

D_c is the directivity correction, in decibel, that describes the extent by which the equivalent continuous sound pressure level from the point sound source deviated in a specific direction from the level of an omnidirectional point sound source producing sound power levels L_W; D_c equals the directivity index D_I of the point sound source plus an index D_Ω that accounts for sound propagations into solid angles less than 4π radians; for an omnidirectional point sound source radiating into free space, $D_c = 0\ dB$

A is the octave band attenuation, in decibels, that occurs during propagation form the point sound source to the receiver.

The attenuation term A in equation 1 is given by equation 3.3 (International Organization for Standardization 1996).

$$A = A_{div} + A_{atm} + A_{gr} + A_{bar} + A_{misc} \quad \text{... (3.3)}$$

Where

A_{div} is the attenuation due to geometrical divergence

A_{atm} is the attenuation due to atmospheric absorption

A_{gr} is the attenuation due to ground effect

A_{bar} is the attenuation due to a barrier

A_{misc} is the attenuation due to miscellaneous other effects (such as foliage, industrial noise, propagation through houses or buildings).

The equivalent continuous A-weighted downwind sound pressure level L_{AT} shall be obtained by summing the contributing time-mean-square sound pressures calculated according to equations 3.2 and 3.3 for each point sound source, for each of their image sources and for each octave band, as specified by equation 3.4 (International Organization for Standardization 1996).

$$L_{AT}(DW) = 10\lg\{\textstyle\sum_{i=1}^{n}[\sum_{j=1}^{8} 10^{0.1}[L_{fT}(ij) + A_f(j)]\}\, dB \qquad \ldots (3.4)$$

Where

n is the number of contributions i (sources and paths)

j is an index indicating the eight standard octave-band midband frequencies from 63 Hz to 8kHz

A_f denotes the standard A-weighting

To apply this method, parameters such as the geometry of the source and of the environment, the ground surface characteristics, and the source must be known. Accuracy limitations for this method can include the attenuation of sound propagation outdoors between a fixed source and receiver fluctuating due to variations in the metrological conditions along the propagation path (AWS Truepower 2014).

3.2.2 Noise Model

A noise model theoretically estimates noise levels within a region of interest under specific parameters. It is important to understand that the specific set of conditions for which the noise is being modeled will be only a ‘snapshot’ of a certain environment. Physical environments, particularly those found outdoors, will not be fixed but have constantly varying conditions leading to constantly varying sound fields. Recognizing that modeling is a means of estimating noise it is imperative to validate the model predictions with the measured data.

A noise propagation model for a potential wind energy generation site located in southern Georgia was created. The model consisted of four Bergey Excel 10 turbines, with a total sound power level of 90.18 dB(A) (Bergey Wind Power 2010) which is representative of L_W in equation 3.2 with a noise map generated at an observer height of 1.75m (5.7 ft.) above ground level. The noise analysis was conducted using the single A-weighted sound power level (ISO 9613-2) noise model in Openwind®. In this model, total sound power level at source (dB(A)), atmospheric attenuation, attenuation due to geometric spreading, ground effect attenuation, and site specific temperature and humidity information were all accounted for. A site specific relative humidity of 52.2%, a site specific temperature of 33.57° C, and a site specific air density of 1.139 kg/m3 were used as inputs for A_{atm} in equation 2. Additionally, an observer height of 1.75 m and a ground porosity of 0.75 were used and is representative of A_{gr}in equation 3.3. The total sound power level of 90.18 dB(A) (L_W) for the Bergey Excel 10 turbine was taken from manufacturer specifications. A correction factor of -10 dB(A) was also used to compensate for miscellaneous attenuation and is representative of a negative A_{misc} term in equation 3.3, this equates to some noise amplification from the environment due to echoing and hard packed asphalt surfaces (AWS Truepower 2015). This noise model was used to predict sound levels at residences that may be affected by wind turbine noise. These are considered receptors for noise calculation purposes (Kwong et al. 2012).

This model does come with several simplifying assumptions the first being reflections are ignored as wind turbines are aerial sources of noise. This sound model is only representative of an area of land with flat or constantly sloping terrain. The model does not take into effect terrain features such as hills that can act as barriers to the sound propagation. The model also does not take other obstacles into account such as vegetation or buildings. In the model ambient noise is also ignored.

3.3 Wind Speed Prediction

This body of research was established in three different stages: firstly, the data was collected as well as pre-processed, secondly, ANN modelling and, finally, the performance was analyzed and comparisons were drawn between the two different networks: NAR and NARX networks implemented within MATLAB neural network toolbox.

It has been shown in previous studies that a standard feedforward artificial neural network can yield very good results in time series prediction tasks (Haydari 2007, Doucoure et al. 2016, Macas et al. 2016, Kiartzis et al. 1995, Cao 2012, Mohanty et al. 2015). Two ANN architectures were selected that fit for the purpose of creating prediction models for wind speed. One model is called a non-linear autoregressive neural network (NAR). NAR networks are used to forecast data from a one –dimensional time series. The other model is called a non-linear autoregressive network with exogenous input (NARX). NARX networks forecast data from a multidimensional time series using external information to improve forecast performance in a time series.

NAR and NARX both have pros and cons: the NAR method are simpler than NARX and require less data. NARX methods allow the use of more information that corresponds to the data set to be predicted. In a wind power generation application meteorological towers at the site should generate wind speed data as well as other corresponding weather data such as pressure and temperature. The next subsections describe each model and how it can solve the issue of wind speed time series regression problem.

3.3.1 NAR Model

In most applications, time series problems have a high degree of transient periods as well as great variation or disparity. This is why most time series problems are difficult to approximate using a linear model; this is why a non-linear approach is recommended. A nonlinear autoregressive neural network (Nyanteh et al. 2013, Kisi 2007, Jursa 2007) that is used for a time series regression problem, describes a discrete, non-linear, autoregressive model that can be expressed in the equation below (Mathworks 2014).

$$y(t) = f\big(y(t-1), \dots y(t-d)\big) \tag{3.5}$$

Equation 3.5 defines how NAR methods are used to predict the value of a data series *y* at the time *t*, *y(t)* using the *d* past values of the time series. The function *f(·)* is not known in prior to training. During the training stage the NN tries to determine optimal weights and neuron biases in order to approximate the function.

The topology of a NAR network can be seen in figure 3.3 below. The *d* features $\big(y(t-1), \dots y(t-d)\big)$ are called feedback delays. The number of feedback delays as well as the number of neurons per hidden layer is adjustable. The number of feedback delays and neurons per hidden layer are optimized through trial-and-error testing in order to obtain the network architecture for the greatest performance. It must be noted that increasing the number of neurons in the hidden layer makes a system more complex and decreasing the amount of neurons in the hidden layer will lower the computing power and generalization of the ANN.

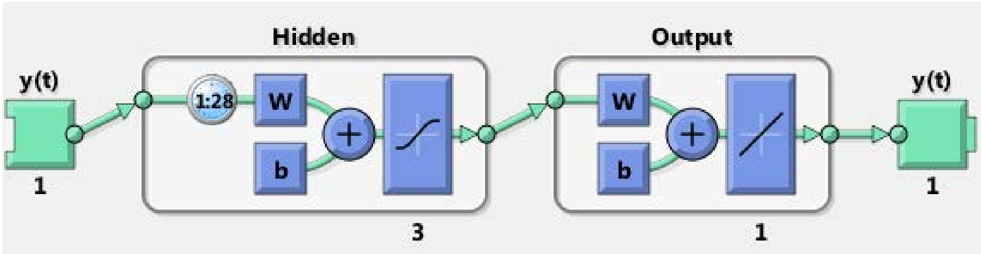


Figure 3.3 Neural Network setup for an open Nonlinear Autoregressive (NAR) time series problem

The learning rule used for NAR networks is the Levenberg-Marquardt backpropagation procedure (LMBP) (Marquardt 1963, Hagan et al. 1996, Cigizoglu and Kisi 2005). This training function is used the majority of the time because it is usually the fastest back propagation method. The LMBP is used in order to calculate the second-order derivative without having to calculate the Hessian matrix. This is why LMPB has the fastest training speed. The performance function is in the form of a sum of squares as is normal in feed forward network training. This performance function allows the Hessian matrix to be calculated (Eq 3.6) and the gradient can be approximated (Eq 3.7).

$$H = J^T J \tag{3.6}$$

$$g = J^T e \tag{3.7}$$

In equations (3.6) and (3.7), *J* is the Jacobian matrix. The Jacobian matrix has the first derivatives of the network error with respect the weights and biases. The variable *e* is a vector of the network errors in every training sample. In order to approximate the Jacobian matrix the study by (Hagan et al. 1996) uses the typical backpropagation method to estimate the Hessian matrix. The Levenberg-Marquardt method uses the following approach to approximate the Hessian matrix (Eq 3.8)

$$x_{k+1} = x_k - [J^T J + \mu I]^{-1} J^T e \tag{3.8}$$

This method used in this ANN assumes that the performance function is sum of squares such as mean square error (MSE) or error sum of squares (SSE) as stated in Equations 3.9 and 3.10 below. In these two equations y_i stands for the i-th data sample, $\hat{y}_i$ represents the data that was approximated by the network for y_i, and *n* represents the number of data samples for the network training.

$$SSE = \sum_{i=1}^{n} (\hat{y}_i - y_i)^2 \tag{3.9}$$

$$MSE = \frac{SSE}{n} \tag{3.10}$$

In this body of research, the NAR method is used to model a wind speed time series regression problem and is planned as such: the network architecture receives one input (corresponding to the wind speed at time *t-1*, *y(t-1)*), and one output (the following value of the series, *y(t)*, to be predicted). The amount of delays and hidden neurons to be used are determined experimentally after data is normalized and analyzed.

After an optimized NAR architecture has been established, the performance of one step ahead prediction (Fig 3.4) and multi-step ahead prediction using a closed loop network (Fig 3.5) are evaluated. A single step ahead prediction network is created by removing one delay tap so that its minimal delay tap is now 0 instead of 1. The new network returns the same outputs as the original network, but outputs are shifted one time step. A closed loop network is created by replacing the feedback input with a direct connection from the output layer. When using multistep prediction the network is simulated in open loop form for as long as there is known output data, then it is switched to closed loop form to perform multistep prediction while providing only the external input. In this study all but five time steps of the input series and target series (of hourly wind speed) are used to simulate the network in open loop form. This produces a forecast five hours ahead of the most recent data collection point.

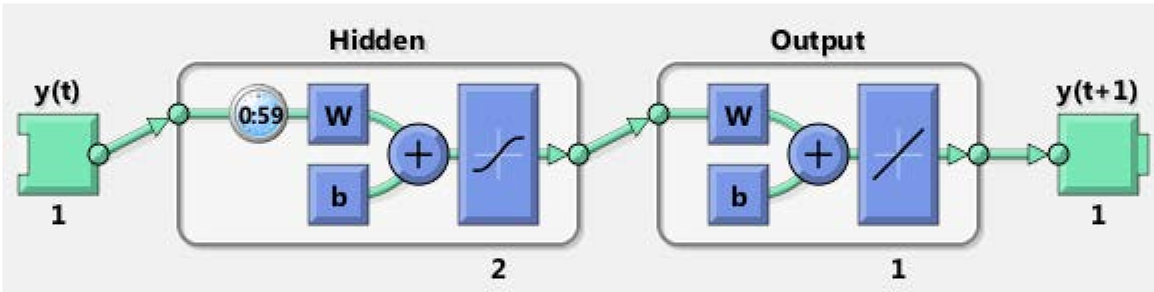


Figure 3.4 Neural Network setup for an open loop Nonlinear Autoregressive (NAR) time series problem for step ahead prediction

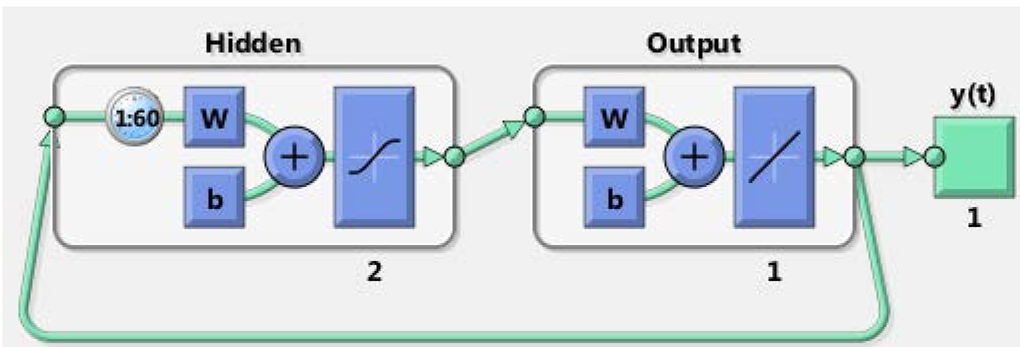


Figure 3.5 Neural Network setup for a closed loop Nonlinear Autoregressive (NAR) time series problem for multistep ahead prediction

3.3.2 NARX Model

In many applications time series have important correlations between the time series to be modeled as well as additional exogenous data. It is known that wind speeds are highly correlated with both temperature and pressure [Lei et al. 2009, Kaminsky et al. 1985]. The usage of these additional weather data sets could benefit the forecasting of wind speed in order to provide a more accurate prediction [Berge 2002, Benoit and Yu 2002].

Nonlinear autoregressive with exogenous (external) inputs, NARX, is the other model to be used in this study, as proposed in [Lin et al. 1996]. NARX methods predict the time series *y(t)* given past values *d* of series *y* and another external input series *x(t),* which can be single or multidimensional inputs. Equation 3.11 models the NARX methods behavior when conducting regressive time series forecasting (Mathworks 2014).

$$y(t) = f(x(t-1), \dots, x(t-d), y(t-1), \dots, y(t-d)) \tag{3.11}$$

The NARX method is a nonlinear model that approximates time step ahead values of a time series based on previous outputs and external data. This body of research uses one input for the wind speed time series at time *t-1*, *y(t-1)*, and an additional external input of dry bulb temperature at time *t-1*, *x(t-1)* to produce a single output *y(t)* that corresponds to the value of the wind speed at one time step (one hour) forward. Figure 3.6 below shows the topology for the NARX network. The learning rule used in training is still the LMBP as explained in the previous section.

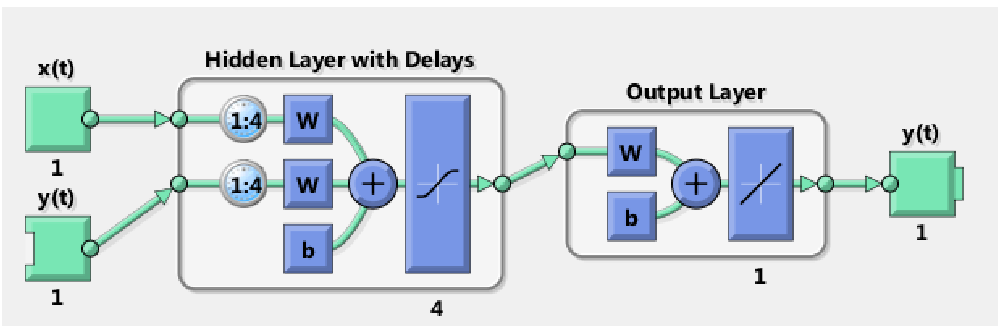


Figure 3.6 Neural Network setup for an open loop Nonlinear Autoregressive with exogenous inputs (NAR) time series problem

After an optimized NARX architecture is established the performance of one step ahead prediction (Fig 3.7) and multi-step ahead prediction using a closed loop network (Fig 3.8) will be evaluated. A single step ahead prediction network is created by removing one delay tap so that its minimal delay tap is now 0 instead of 1. The new network returns the same outputs as the original network, but outputs are shifted one time step. A closed loop network is created by replacing the feedback input with a direct connection from the output layer. When using multistep prediction the network is simulated in open loop form for as long as there is known output data, then it is switched to closed loop form to perform multistep prediction while providing only the external input. In this study all but five time steps of the input series and target series are used to simulate the network in open loop form. This produces a forecast five hours ahead of the most recent data collection point.

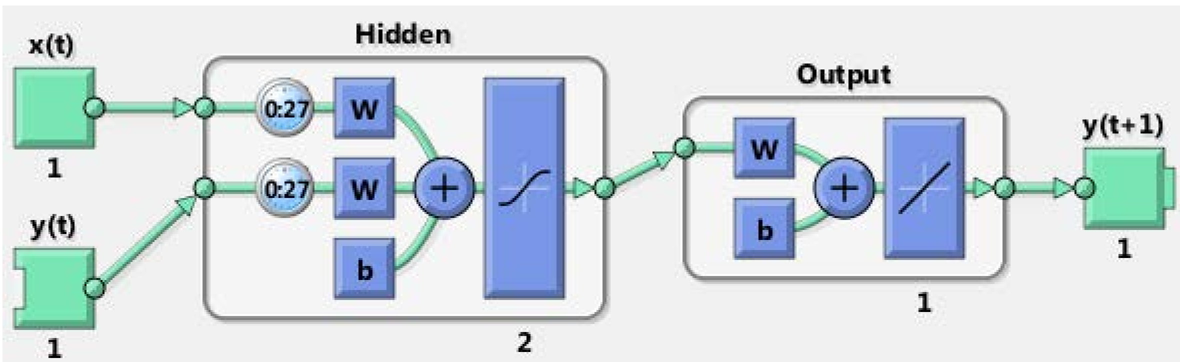


Figure 3.7 Neural Network setup for an open loop Nonlinear Autoregressive (NARX) time series problem for one step ahead prediction

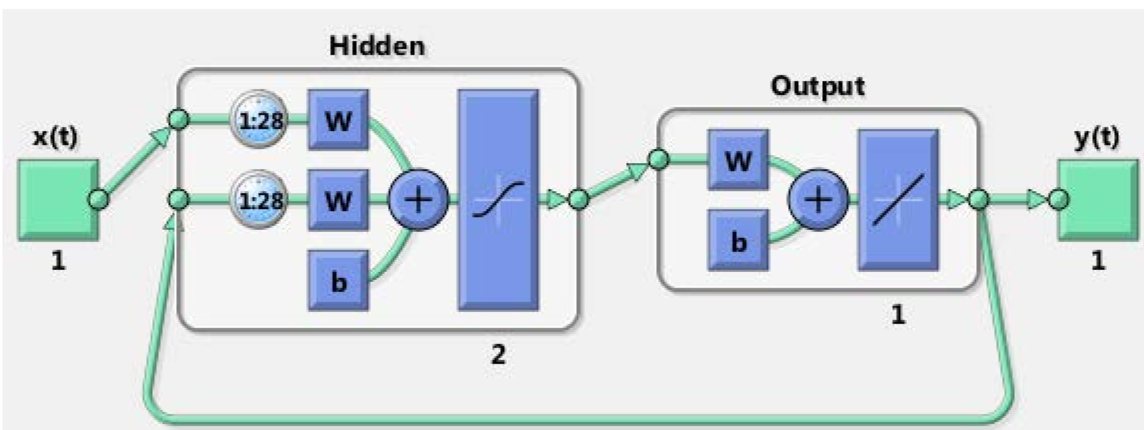


Figure 3.8 Neural Network setup for a closed loop Nonlinear Autoregressive (NARX) time series problem for multistep ahead prediction

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Wind Turbine Noise Prediction Results

4.1.1 Methods

Google Earth was used to retrieve satellite imagery for the potential wind energy demonstration site (Figure 4.1). Polygons of the proposed site were then taken to gain an accurate geometric model of the land area with correct coordinates and spatial data (Figure 4.2). Points were then used to index the location of the turbines, meteorological tower, and occupied residences (Figure 4.3). This spatial data was used to create a shape file (.shp) to represent the potential site geometrically (Figure 4.4).

Figure 4.1 Satellite imagery of the potential wind energy demonstration site

Figure 4.2 Polygon to represent the potential site in the model

Figure 4.3 Points to represent homes and turbines in the model

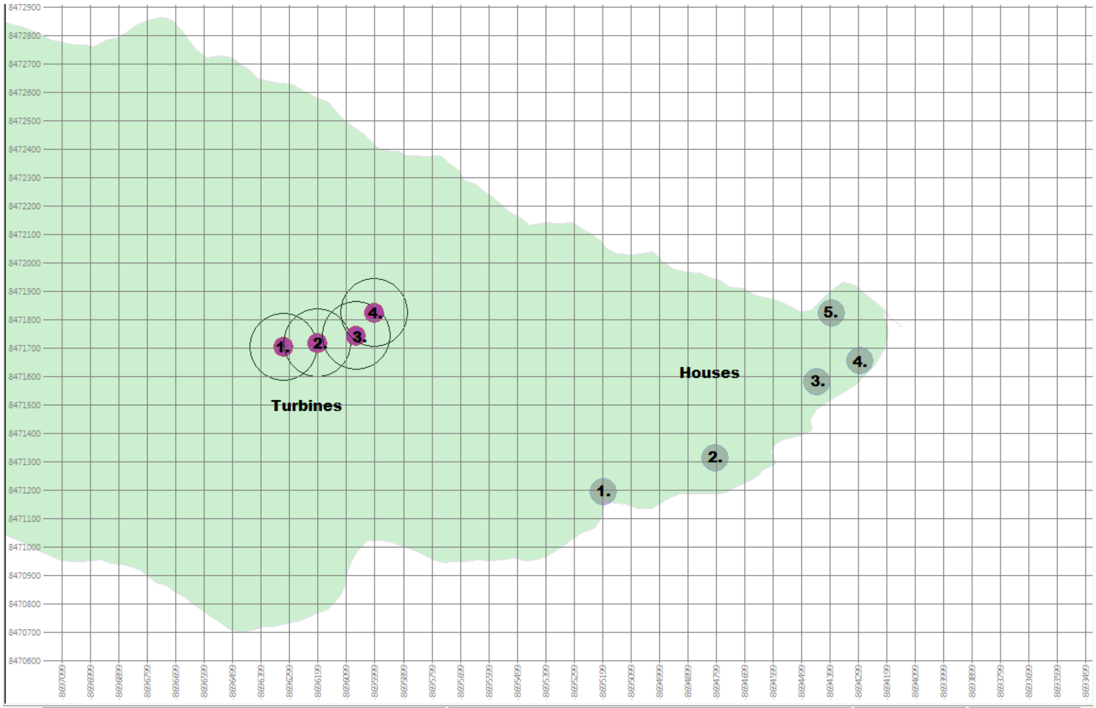


Figure 4.4 Completed shape file to represent the potential site's spatial data.

4.1.2 Predicted Noise Maps

Given the inputs specified above and in section 3.2.2, noise maps were predicted for the potential site with one turbine (Figure 4.5), two turbines (Figure 4.6), three turbines (Figure 4.7), and four turbines (Figure 4.8). The number of turbines used is representative of *n* in equation 4.3. The models depicted shows isolines of a 5 dB(A) gradation from the turbines in each case.

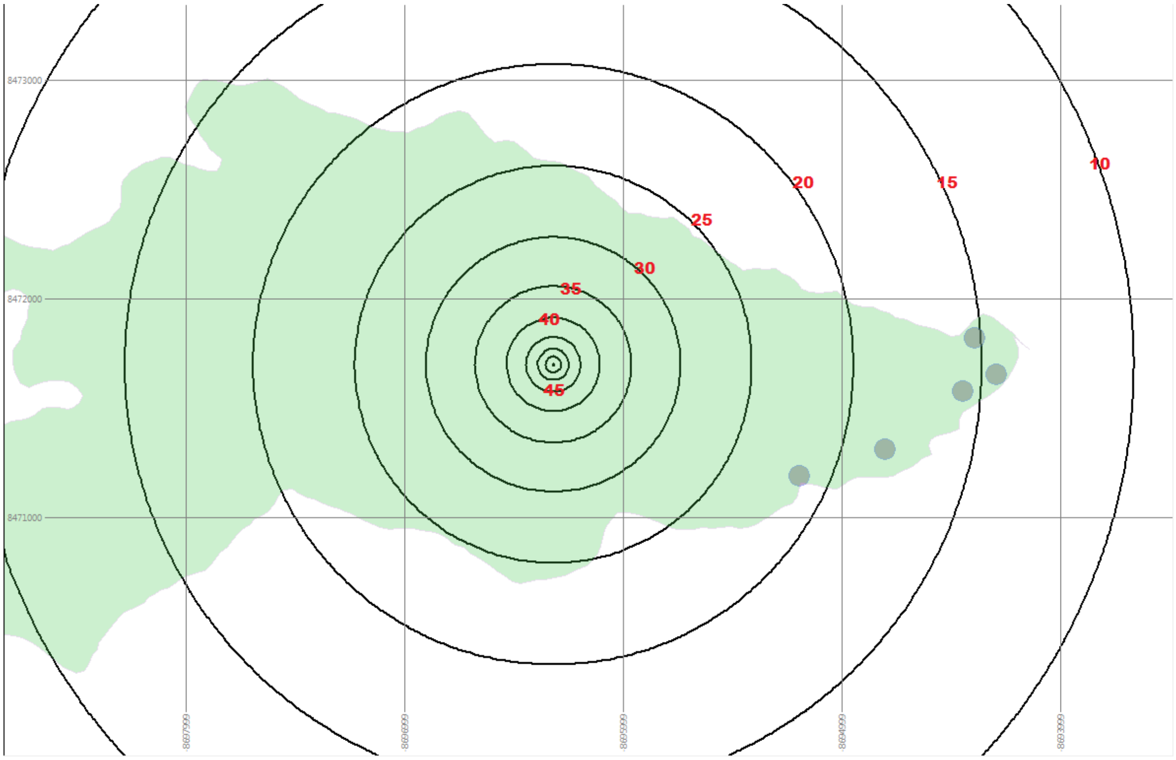


Figure 4.5 Noise map with one turbine

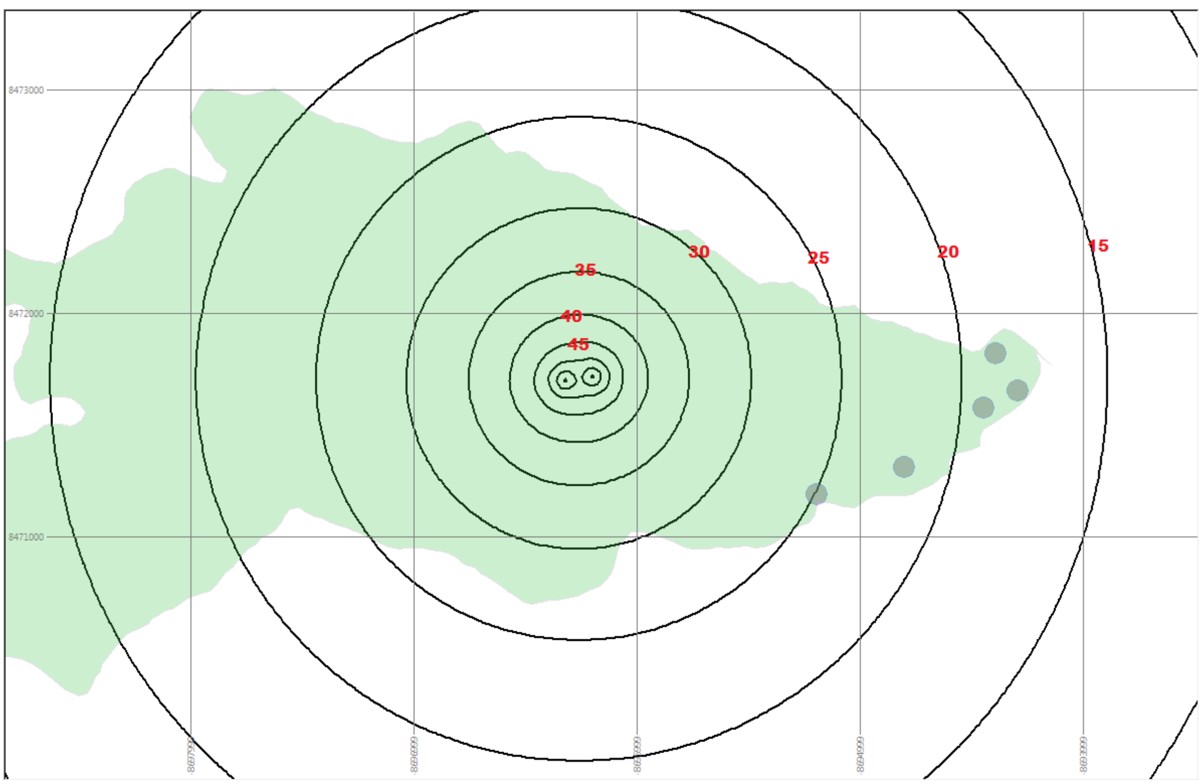


Figure 4.6 Noise map with two turbines

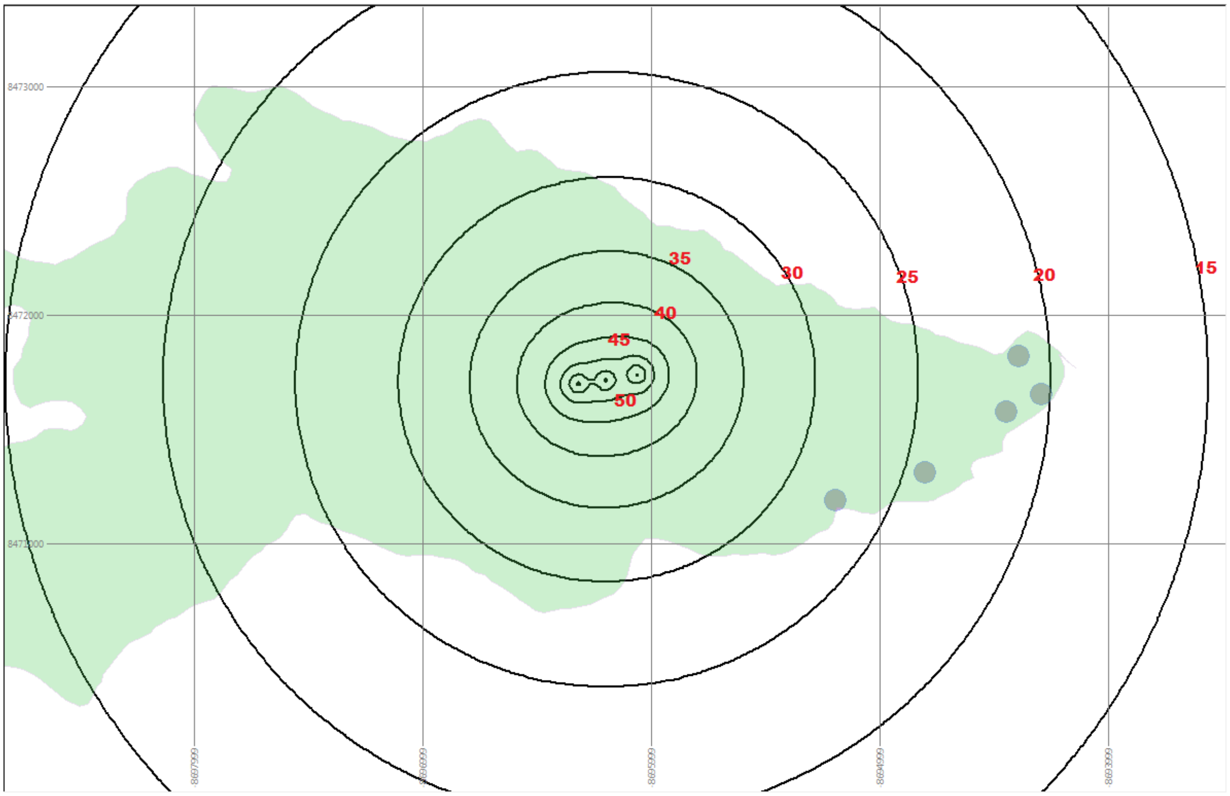


Figure 4.7 Noise map with three turbines

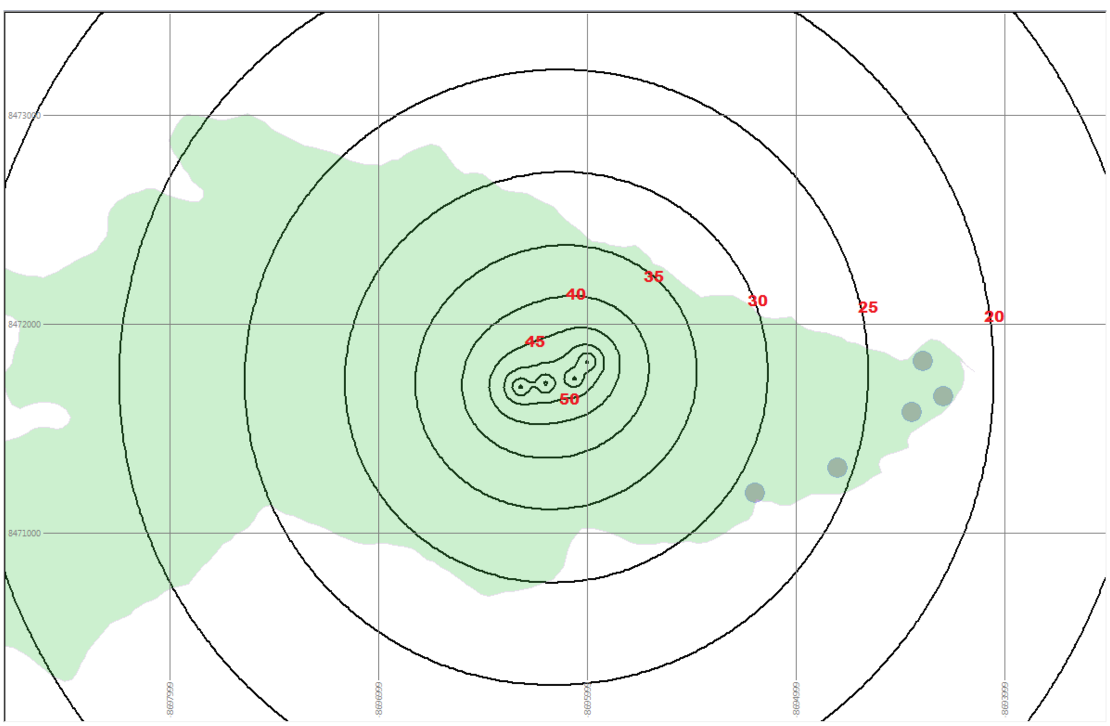


Figure 4.8 Noise map with four turbines

Table 4.1 depicts the various sound intensity levels at the homes on the map determined by observing the isolines at a resolution of one dB(A) per iteration. Table 4.1 also shows the max sound intensity level predicted 450 ft. from the turbine layout. These results could vary slightly from other future studies due to the lack of land parcel data that would accurately place the property lines of the homes in the geometry of the model. The results can also differ slightly due to several different site specific ambient conditions chosen including temperature, site specific air density, wind speed, and relative humidity.

Table 4.1 Predicted sound intensity at homes on the potential site

Predicted Sound Intesnsity in dB(a) from Openwind						
# turbines	Max (450 ft from turbine	Home 1	Home 2	Home 3	Home 4	Home 5
1	46.0	22.0	14.0	16.0	15.0	15.0
2	50.0	25.0	22.0	19.0	18.0	18.0
3	50.0	28.0	25.0	22.0	21.0	20.0
4	50.0	29.0	26.0	23.0	23.0	22.0

4.2 Analysis

The ambient sound measurements and the theoretical sound predictions for the turbines were taken into account to generate a resultant sound level prediction at the residential locations on the potential site.

4.2.1 Sample Calculation

The formula for the sum level of sound pressures of n incoherent radiating sources is (Sengpiel)

$$L_{\Sigma} = 10 * \log_{10}\left(\frac{p_1^2 + p_2^2 + \cdots + p_n^2}{p_0^2}\right) = 10 * \log_{10}(\left(\frac{p_1}{p_0}\right)^2 + \left(\frac{p_2}{p_0}\right)^2 + \cdots + \left(\frac{p_n}{p_0}\right)^2 \quad (4.1)$$

The reference sound pressure p_0 is 20 µPa = 0.00002 Pa = 2×10^{-5} Pa (RMS) ≡ 0 dB. From the formula of the sound pressure level we find

$$\left(\frac{p_i}{p_0}\right)^2 = 10^{\frac{L_i}{10}}, \quad i = 1, 2, \dots, n \qquad \dots (4.2)$$

This inserted in the formula for the sound pressure level to calculate the sum level shows (Sengpiel)

$$L_\Sigma = 10 * \log_{10}\left(10^{\frac{L_1}{10}} + 10^{\frac{L_2}{10}} + \cdots + 10^{\frac{L_n}{10}}\right) dB \qquad \dots (4.3)$$

L_Σ = Total level and L_1, L_2 ... L_n = sound pressure level of the separate sources in dBSPL. Incoherent means: lacking cohesion, connection, or harmony. It is not coherent.

For example, adding of three decibel values, that means levels 94.0 + 96.0 + 98.0:

$$L = 10 * \log_{10}(10^{9.4} + 10^{9.6} + 10^{9.8}) = 101.1\ dB \qquad \dots (4.4)$$

4.2.2 Predicted Level with Ambient Noise

Using the method described above the peak predicted sound intensity levels, at a distance of 450 ft. from the turbines, were considered along with the peak value measured at the turbine locations from Section 3.1.2 (52.10 dB(A)) for the max column. The predicted sound intensity levels at the homes were considered along with the average residence values measured (46.57 dB(A)) for the home columns (Table 4.2).

Table 4.2 Predicted sound intensity at residences on the potential site summed with measured values

Sound Intensity in dB(A): Combined noise level of turbines and ambient [46 dB(A)]						
# of turbines	Max (450 ft. from turbine	Home 1	Home 2	Home 3	Home 4	Home 5
1	53.1	46.6	46.6	46.6	46.6	46.6
2	54.2	46.6	46.6	46.6	46.6	46.6
3	54.2	46.6	46.6	46.6	46.6	46.6
4	54.2	46.6	46.6	46.6	46.6	46.6

It can be observed from Table 4.2 that the effect of sound propagation from the wind turbines in the residential areas is negligible when considered along with the average (46 dB(A)) ambient noise measured at the site. Even the worst case scenario of the max value 450 ft. from the turbine location would only increase the dB(A) value by 4.2 - 7.1 dB(A). One dB(A) is the smallest increment of sound detectable by the human ear so an increase of 4.2-7.1 dB(A) would be largely unnoticed by a human observer.

Table 4.3 Predicted sound intensity at residences on proposed site summed with an estimated 30 dB(A) night ambient noise

Sound Intensity in dB(A): Combined noise level of turbines and predicted ambient [30 dB(A)]						
# of turbines	Max (450 ft. from turbine	Home 1	Home 2	Home 3	Home 4	Home 5
1	46.1	30.6	30.1	30.2	30.1	30.1
2	50.0	31.2	30.6	30.3	30.3	30.3
3	50.0	32.1	31.2	30.6	30.5	30.4
4	50.0	32.5	31.5	30.8	30.8	30.6

It can be observed from Table 4.3 that the effect of sound propagation from the wind turbines in the residential areas is again negligible when summed with an estimated 30 dB(A) ambient noise at night. The sound intensity levels of the wind turbines are much lower than the 30 dB(A) ambient noise. The effect observed at the residences from the noise generated by the wind turbines would be an increase in the range of 0.1- 2.5 dB(A) barely noticeable to a human observer.

Table 4.4 Predicted sound intensity at residences on proposed site summed with an estimated 20 dB(A) night ambient noise

Sound Intensity in dB(A): Combined noise level of turbines and predicted ambient [20 dB(A)]						
# of turbines	Max (450 ft. from turbine	Home 1	Home 2	Home 3	Home 4	Home 5
1	46.0	24.1	21.0	21.5	21.2	21.2
2	50.0	26.2	24.1	22.5	22.1	22.1
3	50.0	28.6	26.2	24.1	23.5	23.0
4	50.0	29.5	27.0	24.8	24.8	24.1

It can be observed from Table 4.4 that the effect of sound propagation from the wind turbines in the residential areas could be noticeable under these conditions when considered along with an estimated 20 dB ambient noise at night. The sound intensity levels of the wind turbines are much closer to the 20 dB(A) ambient noise predicted than in other scenarios. The effect observed at the residences from the noise generated by the wind turbines would be an increase in the range of 1- 9.5 dB(A) which could be noticeable to a human observer.

Table 4.5 Predicted sound intensity at residences on proposed site summed with an estimated 10 dB night prediction

Sound Intensity in dB(A): Combined noise level of turbines and predicted ambient [10 dB(A)]						
# of turbines	Max (450 ft. from turbine	Home 1	Home 2	Home 3	Home 4	Home 5
1	46.0	22.3	15.5	17.0	16.2	16.2
2	50.0	25.1	22.3	19.5	18.6	18.6
3	50.0	28.1	25.1	22.3	21.3	20.4
4	50.0	29.1	26.1	23.2	23.2	22.3

It can be observed from Table 4.5 that the effect of sound propagation from the wind turbines in the residential areas would be noticeable under these conditions when summed with an estimated 10 dB(A) ambient noise at night. The sound intensity levels of the wind turbines are higher than the 10 dB(A) ambient noise than in other scenarios. The effect observed at the residences from the noise generated by the wind turbines would be an increase in the range of

6.2- 19.1 dB(A) which would be noticeable to a human observer. An increase of this magnitude would seem to be 6.2-19.1 times louder in perceived loudness (Brüel & Kjær 2000).

In conclusion the noise produced by the wind turbines is predicted to be negligible on all accounts when observed from the residences unless it is an extremely quiet night (10-20 dB(A) or equivalent to the sound of falling leaves to whispering) 1-19 dB(A) range of increase could be expected under these conditions which would be perceived as much louder than pure ambient noise by a human observer. This would be a very rare occasion for ambient noise to reach these very low levels. However the results presented are a conservative estimate of actual sound levels. The model does not take into account hills or mountains, vegetation, or other obstacles to sound propagation such as buildings and other constructions near the site.

4.3 Wind Speed Prediction Results

This section describes all the tests that were performed and the experimental setting used. Figure 4.9 contains all wind data (NCDC CDO 2010) used in the study but the time steps have been increased to every 48 hours to produce a plot with greater legibility. Noting the shape of the curves the Logan International Airport and JFK airport seem to have higher variability and overall wind speeds while the Bismarck Municipal Airport and Savannah International Airport seem to have less variable and overall lower wind speeds.

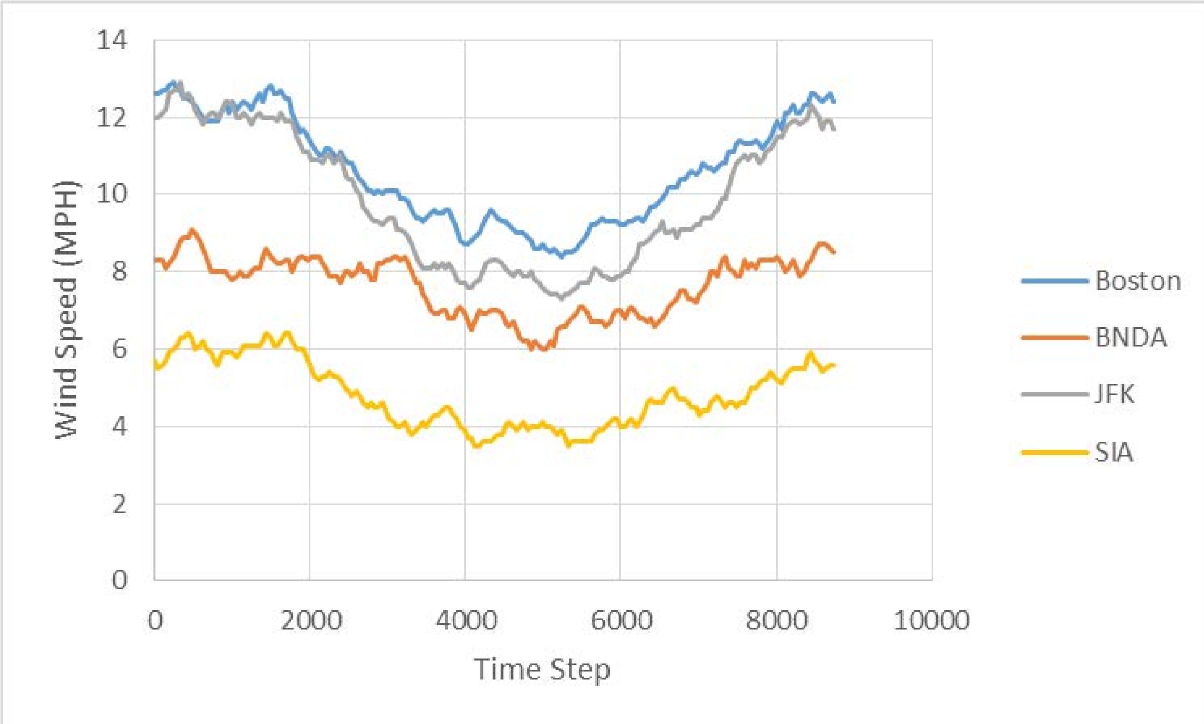


Figure 4.9 Average hourly wind speed data for the four different airports used in the study displayed as time steps of every 48 hours for legibility *(NCDC CDO 2010).*

4.4 Optimization of the network architecture

Each data set does not have a relationship between each other due to the large distances between each measuring location. This is why it was important to adjust the delay parameters for each data set individually. Delay parameters concern the number of hours the ANN will use to execute the prediction. Put simply, the model is trained with the last *d time*-steps as delays. Eighteen different tests were conducted on each data set in order to determine the optimal number of delays. The delay values included *d=2*, *d=4*, *d=8*, the last 12 hours *d=12*, *d=16*, *d=20*, the last day *d=24*, *d=36*, the last two days *d=48*, *d=60*, and the last three days *d=72*. To find the best delay, all the parameters were set to a fixed value (hidden neurons set at 10) and the delays were modified in a trial and error procedure in order to optimize performance.

Table 4.6 shows the average mean square error of 10 executions for each delay value and airport, using the NAR model. The best delay values are marked in bold, for all data sets 2 delays resulted in the highest error. The minimum number of delays in order to get the lowest error was 48 hours for the Bismarck Municipal data set while the highest was 72 hours for the JFK dataset, the other two data sets had the best error at the 60 delays. From this information it can be determined that a delay parameter between 48-72 previous hours is needed in order to obtain an accurate model.

Once the optimal amount of delays has been determined that value is used as a fixed value for that dataset in the neurons test. The neuron count needed to train the NAR was then adjusted. With the fixed delay value determined from each dataset's lowest MSE value, 10 different executions for each neuron count were conducted and the average MSE of each setting was calculated. Table 4.7 shows a different number of neurons ranging from two to twenty contained in the hidden layer. The purpose of this experiment was to determine which network architecture could provide the lowest error and therefore best performance. It was found that as the neuron count was increased the prediction was worse due to local optimum optimization results using the LMBP learning algorithm, as well as overtraining. From table 4.7, it can be concluded that the best average MSE values occur at either two or three neurons in the hidden layer, depending on which location was being tested.

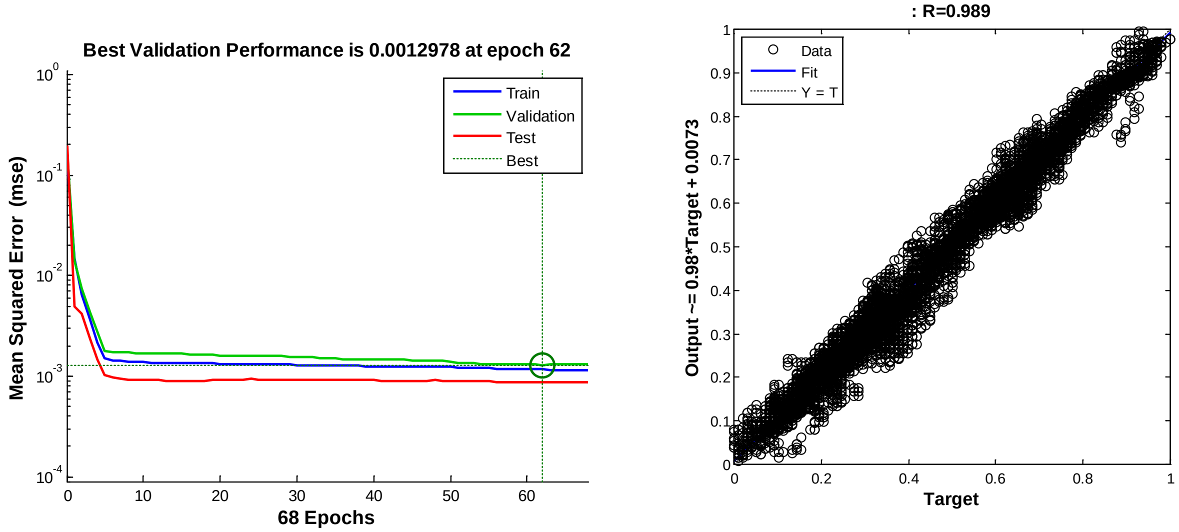
Table 4.6 MSE of the delay parameter obtained with the NAR model. (Bold values: best delay)

	NAR Delays								
Wind Speed	2	3	4	6	8	10	12	14	16
Boston	1.21E-03	1.09E-03	1.03E-03	1.08E-03	1.00E-03	8.83E-04	9.94E-04	9.14E-04	8.22E-04
BNDA	1.22E-03	1.13E-03	1.25E-03	1.33E-03	1.44E-03	1.54E-03	1.69E-03	1.60E-03	1.98E-03
JFK	1.33E-03	1.13E-03	1.08E-03	1.24E-03	1.21E-03	1.14E-03	1.28E-03	1.55E-03	9.86E-04
SIA	1.29E-03	1.20E-03	1.04E-03	1.18E-03	1.95E-03	2.04E-03	2.29E-03	1.50E-03	1.76E-03
MEAN	1.26E-03	1.14E-03	1.10E-03	1.21E-03	1.40E-03	1.40E-03	1.57E-03	1.39E-03	1.39E-03
Wind Speed	18	20	24	26	28	36	48	60	72
Boston	5.52E-04	5.46E-04	1.50E-04	7.46E-05	7.29E-05	7.10E-05	7.12E-05	**6.93E-05**	7.05E-05
BNDA	1.34E-03	1.03E-03	1.33E-04	6.36E-05	6.49E-05	6.52E-05	**6.33E-05**	6.68E-05	6.37E-05
JFK	8.97E-04	9.33E-04	1.29E-04	6.42E-05	6.39E-05	6.15E-05	6.44E-05	5.90E-05	**5.87E-05**
SIA	1.22E-03	7.75E-04	1.58E-04	6.26E-05	6.31E-05	6.35E-05	6.64E-05	**5.98E-05**	6.22E-05
MEAN	1.00E-03	8.20E-04	1.42E-04	6.63E-05	6.62E-05	6.53E-05	6.63E-05	**6.37E-05**	6.38E-05

Table 4.7 MSE of the hidden neurons parameter obtained with the NAR model. (Bold values: best neuron count)

	Number of Neurons in the Hidden Layer NAR									
Wind Speed	2	3	4	6	8	10	12	14	16	18	20
Boston	6.91E-05	**5.81E-05**	7.11E-05	7.22E-05	7.59E-05	7.37E-05	7.66E-05	7.48E-05	7.95E-05	7.94E-05	8.38E-05
BNDA	**5.88E-05**	6.01E-05	6.16E-05	6.43E-05	6.73E-05	6.76E-05	7.81E-05	7.36E-05	7.98E-05	8.02E-05	8.35E-05
JFK	**5.84E-05**	6.22E-05	5.85E-05	6.21E-05	6.06E-05	6.06E-05	6.82E-05	7.46E-05	6.61E-05	7.19E-05	6.14E-05
SIA	**5.07E-05**	5.19E-05	5.69E-05	6.22E-05	6.46E-05	7.36E-05	8.36E-05	9.15E-05	7.76E-05	9.87E-05	1.06E-04
MEAN	5.92E-05	**5.81E-05**	6.20E-05	6.52E-05	6.71E-05	6.89E-05	7.66E-05	7.86E-05	7.58E-05	8.26E-05	8.36E-05

After this experimentation was completed, the best NAR networks for predicted wind speed for the four different data sets were determined. Figure 4.10 below illustrates the best and worst trained networks for the 365 days for hourly wind speed data for the Bismarck Municipal Airport. Figure 4.10a shows validation performance of 0.0012978 and regression values of 0.989. Figure 4.10b shows the validation performance of 0.00005578 and a regression value of 0.9994.



(a)

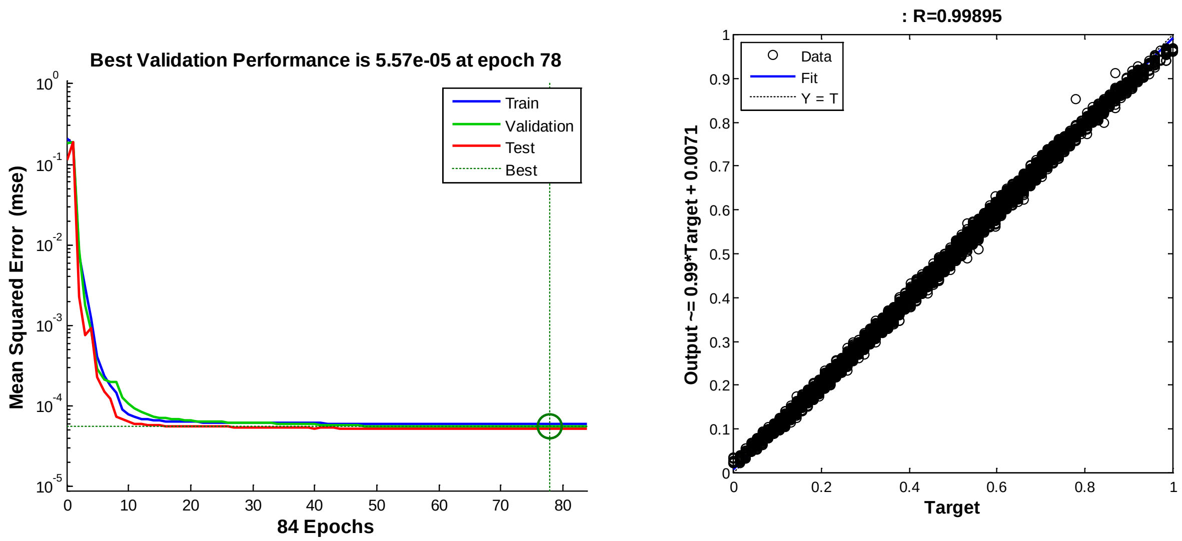


Figure 4.10 (a) Validation performance and Regression Values for the Worst MSE **(b)** Validation performance and Regression Values for the Best MSE for the normalized data of BNDA using the NAR model

The NARX model was tested in a similar fashion however extra information was used that could be useful to prediction. In this experiment dry bulb temperature in Fahrenheit was used as an exogenous input variable. Table 4.8 below illustrates the results pertaining to the amount of delays required to obtain the best model. With the NARX network the best delays were between 28-48 hours while the worst delays were generally under 18 hours.

Table 4.8 MSE of the delay parameter obtained with the NARX model. (bold values: best delay)

NARX Delays									
Wind Speed	2	3	4	6	8	10	12	14	16
Boston	7.66E-04	8.10E-04	7.38E-04	7.81E-04	9.95E-04	9.94E-04	1.11E-03	1.06E-03	1.08E-03
BNDA	1.05E-03	7.64E-04	8.65E-04	1.23E-03	1.32E-03	1.01E-03	8.13E-04	9.42E-04	9.73E-04
JFK	9.80E-04	8.34E-04	8.95E-04	8.92E-04	6.69E-04	7.83E-04	9.13E-04	8.01E-04	9.42E-04
SIA	9.80E-04	8.34E-04	8.95E-04	8.92E-04	6.69E-04	7.83E-04	9.13E-04	8.01E-04	9.42E-04
MEAN	9.45E-04	8.10E-04	8.48E-04	9.49E-04	9.13E-04	8.93E-04	9.37E-04	9.01E-04	9.84E-04
Wind Speed	18	20	24	26	28	36	48	60	72
Boston	6.72E-04	6.58E-04	1.54E-04	8.17E-05	**7.53E-05**	7.99E-05	7.88E-05	7.68E-05	8.13E-05
BNDA	1.11E-03	1.12E-03	1.77E-04	6.68E-05	**6.53E-05**	6.72E-05	7.58E-05	7.39E-05	7.50E-05
JFK	1.05E-03	8.70E-04	1.37E-04	6.98E-05	6.62E-05	6.78E-05	**6.28E-05**	6.76E-05	6.76E-05
SIA	1.05E-03	8.70E-04	1.37E-04	6.98E-05	6.62E-05	6.78E-05	**6.28E-05**	6.76E-05	6.76E-05
MEAN	9.73E-04	8.80E-04	1.51E-04	7.20E-05	**6.82E-05**	7.06E-05	7.01E-05	7.15E-05	7.29E-05

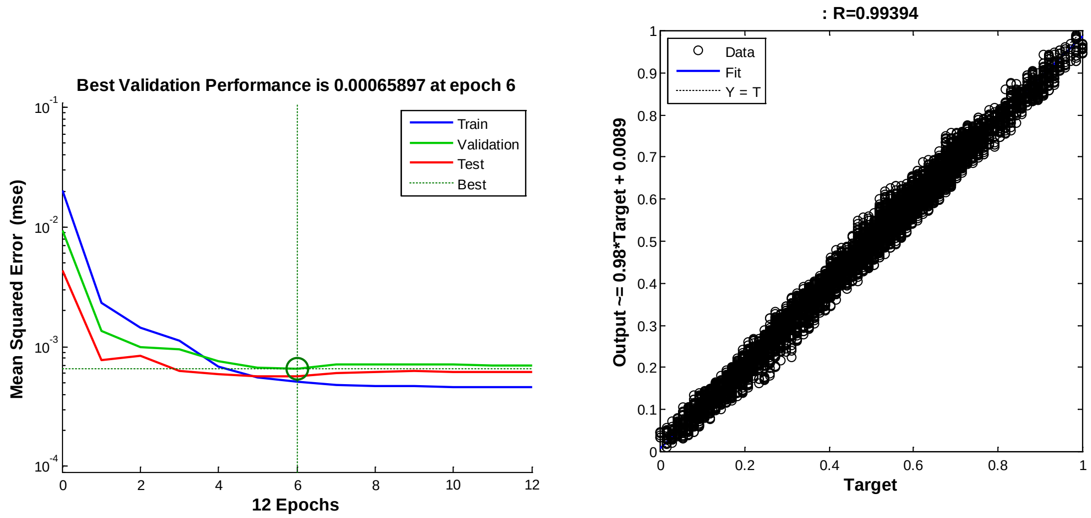
As was done for the NAR model table 4.9 illustrates the neuron values the corresponded to the best MSE for that dataset. The best results were achieved between three to six neurons with the average neuron count being three.

Table 4.9 MSE of the hidden neurons parameter obtained with the NARX model. (Bold values: best neuron count in the hidden layer)

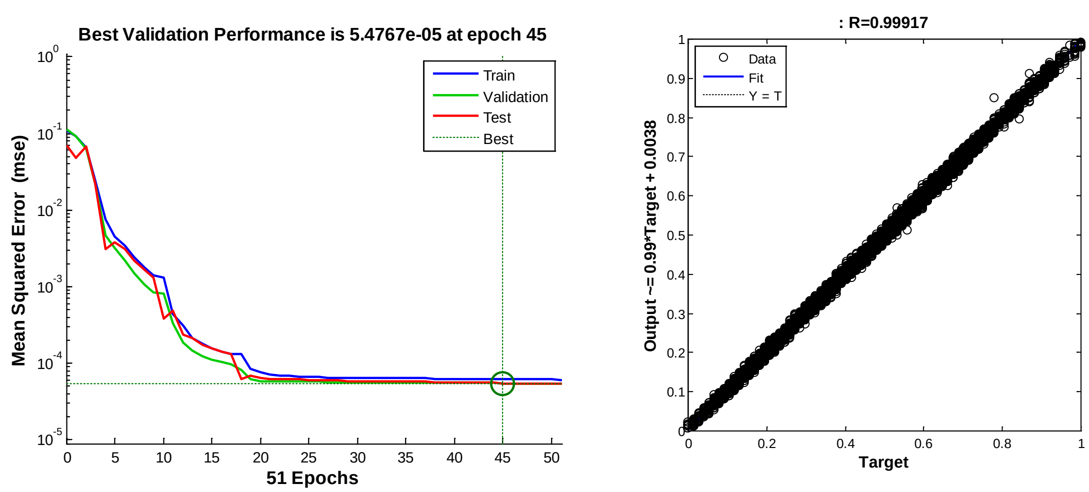
Number of Neurons in the Hidden Layer NARX										
Wind Speed	2	3	4	6	8	10	12	14	16	18	20
Boston	7.06E-05	**5.61E-05**	7.08E-05	7.07E-05	6.83E-05	7.44E-05	7.26E-05	7.23E-05	7.18E-05	7.23E-05	7.26E-05
BNDA	5.71E-05	**5.64E-05**	5.84E-05	6.29E-05	6.55E-05	6.47E-05	6.66E-05	6.48E-05	6.77E-05	6.95E-05	6.46E-05
JFK	6.04E-05	6.16E-05	**5.96E-05**	6.15E-05	6.11E-05	6.26E-05	6.38E-05	6.14E-05	6.33E-05	6.33E-05	6.49E-05
SIA	5.41E-05	5.04E-05	5.06E-05	**4.99E-05**	5.17E-05	5.57E-05	5.19E-05	5.15E-05	5.49E-05	5.16E-05	5.43E-05
MEAN	6.06E-05	**5.61E-05**	5.98E-05	6.13E-05	6.16E-05	6.44E-05	6.37E-05	6.25E-05	6.44E-05	6.42E-05	6.41E-05

After this experimentation was completed, the best NARX networks for predicted wind speed for the four different data sets were determined. Figure 4.11 below illustrates the best and worst trained networks for the 365 days for hourly wind speed data for the Bismarck Municipal

Airport. Figure 4.11a shows validation performance of 0.00065897 and regression values of 0.99394. Figure 4.11b shows the validation performance of 0.000054767 and a regression value of 0.99917.



(a)



(b)

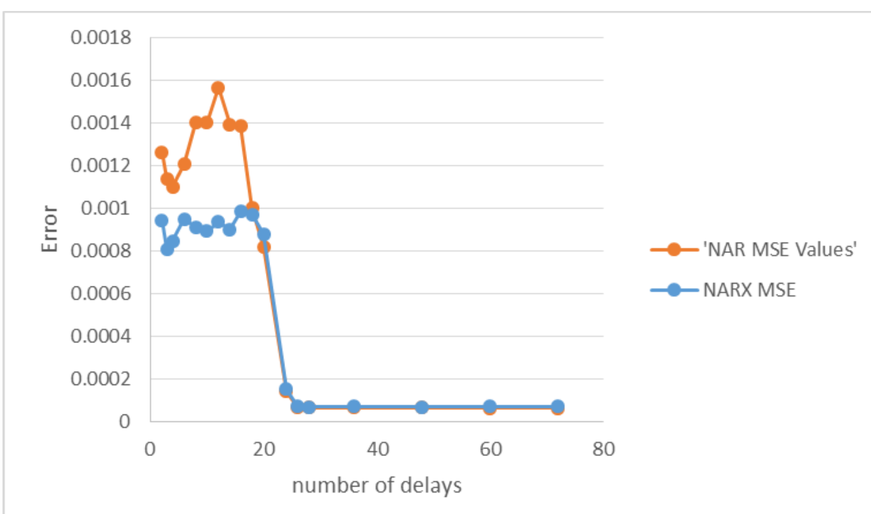
Figure 4.11 (a) Validation performance and Regression Values for the Worst MSE **(b)** Validation performance and Regression Values for the Best MSE for the normalized data of BNDA using the NARX model

According to this information it can be concluded that using a NARX network may reduce the amount of previous data points needed to get an accurate prediction however neuron counts must be increased yielding a more complex model. A summary of these results can be found in table 4.10 below

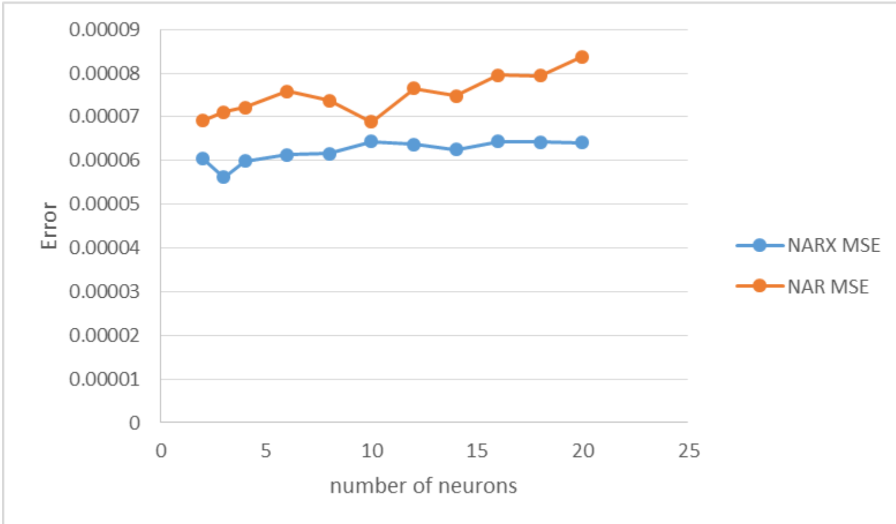
Table 4.10 Number of neurons and delays corresponding to the lowest MSE values for both NAR and NARX methods

Number of Neurons			Number of Delays		
Wind Speed	NAR	NARX	Wind Speed	NAR	NARX
Boston	3	3	Boston	60	28
BNDA	2	3	BNDA	48	28
JFK	2	4	JFK	72	48
SIA	2	6	SIA	60	48
MEAN	2	4	MEAN	60	38

Figure 4.12 below shows a comparison between the means obtained from the NAR and NARX methods in respect to the models number of delays and number neurons. Both of these plots show that the NARX's results have lower error and therefore better performance for predicting wind speed. The NARX's MSE curves are much lower than the NAR curves when the network complexity is simpler.



(a)



(b)

Figure 4.12 (a) Comparison of average MSE in respect to the delay parameter **(b)** Comparison of the average MSE in respect to the network complexity (number of neurons)

4.5 Implementation of optimized networks into step ahead and multi-step ahead prediction

After the network architecture had been optimized the NAR and NARX networks were subjected to one step ahead and multi-step ahead prediction. A single step ahead prediction network is created by removing one delay tap so that its minimal delay tap is now 0 instead of 1, see figure 4.13. The new network returns the same outputs as the original network, but outputs are shifted one time step thus each forecasted time step is occurring at. A closed loop network is created by replacing the feedback input with a direct connection from the output layer. When using multistep prediction the network is simulated in open loop form for as long as there is known output data, then it is switched to closed loop form to perform multistep prediction while providing only the external input this can be noted in figure. In this study all but 5 time steps of the input series and target series are used to simulate the network in open loop form. This produces a forecast five hours ahead of the most recent data collection point.

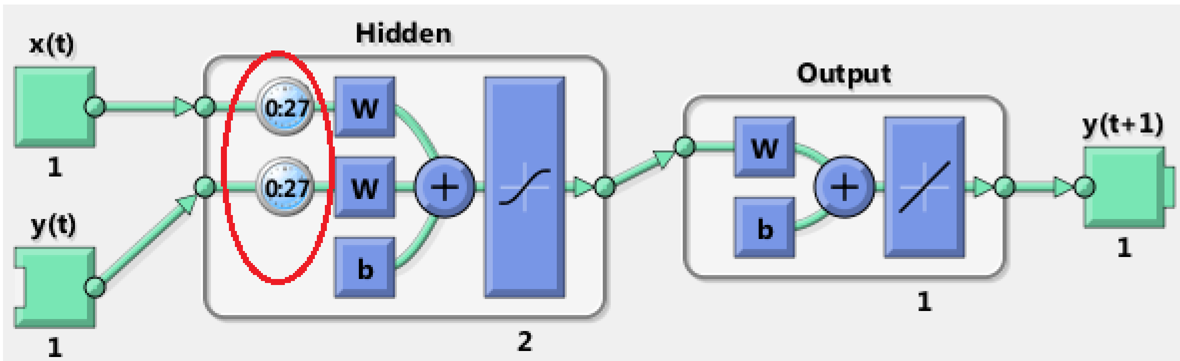


Figure 4.13 Single step ahead prediction using NARX Network with 27 delays and a delay tap of zero (0:27 as opposed to 1:28 in the normal configuration)

Table 4.11 below shows a comparison between the means obtained from the ten trials of the NAR and NARX methods for single step ahead and multi-step ahead prediction. The table displays information illustrating that the NARX's results have lower error and therefore better performance for predicting wind speed for single step ahead prediction. The multistep ahead prediction was much better with the NARX network with two orders of magnitude less error when predicting wind speed five hour in advance with the addition of the exogenous data. Tables 4.12 and 4.13 display the results of paired t test results for both single step ahead and multi-step ahead prediction. The paired t-test confirmed that the difference of prediction results of the NAR and NARX implementation for each case were statistically significant.

Table 4.11 Single step ahead and multi-step ahead MSE values for both NAR and NARX methods and all datasets

Single Step Ahead MSE		
Data Set	NAR	NARX
Boston	7.066E-04	7.306E-05
BNDA	5.871E-04	5.808E-05
JFK	5.815E-04	3.440E-05
SIA	5.853E-04	5.844E-05
MEAN	6.151E-04	5.600E-05

Multi-step Ahead MSE		
Data Set	NAR	NARX
Boston	5.599E-03	2.448E-04
BNDA	3.038E-02	9.524E-05
JFK	2.447E-02	6.460E-05
SIA	9.104E-02	7.645E-05
MEAN	3.787E-02	1.203E-04

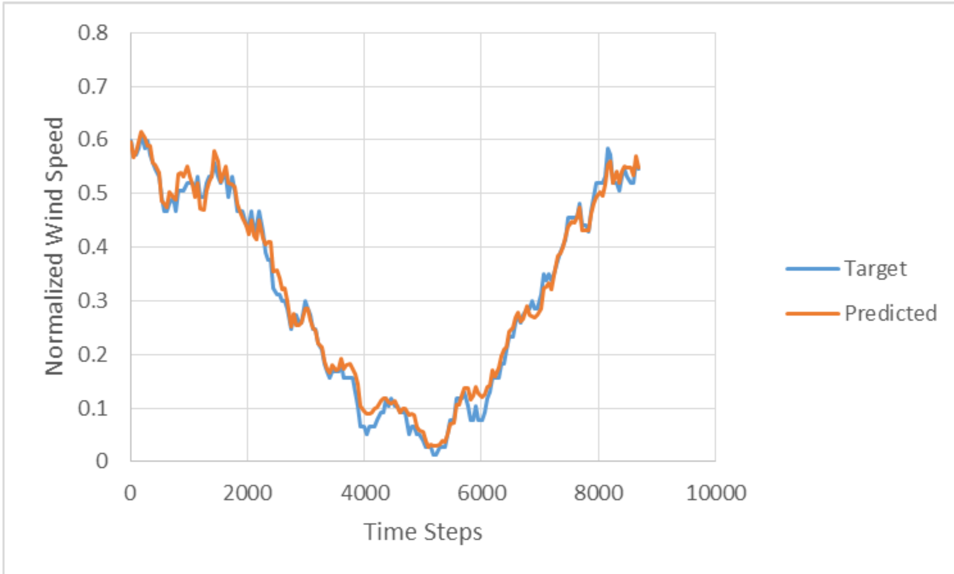
Table 4.12 Summary of paired t-test results for single-step-ahead prediction of wind speed

Dataset	Method	df	Mean	StDev	SE Mean	95% CI for mean difference	t-statistic (H0: $\mu_D = 0$, H1: $\mu_D \neq 0$)	P-Value
Boston	NAR	10	0.000707	0.000016	0.000005	(0.000621, 0.000646)	115.67	0.000
	NARX	10	0.000073	0.000002	0.000001			
	Difference	10	0.000634	0.000017	0.000005			
BNDA	NAR	10	0.000587	0.000023	0.000007	(0.000513, 0.000545)	72.65	0.000
	NARX	10	0.000058	0.000001	0.000000			
	Difference	10	0.000529	0.000023	0.000007			
JFK	NAR	10	0.000581	0.000036	0.000011	(0.000521, 0.000573)	47.63	0.000
	NARX	10	0.000034	0.000006	0.000002			
	Difference	10	0.000547	0.000036	0.000011			
SIA	NAR	10	0.000585	0.000021	0.000007	(0.000511, 0.000542)	77.18	0.000
	NARX	10	0.000058	0.000002	0.000001			
	Difference	10	0.000527	0.000022	0.000007			

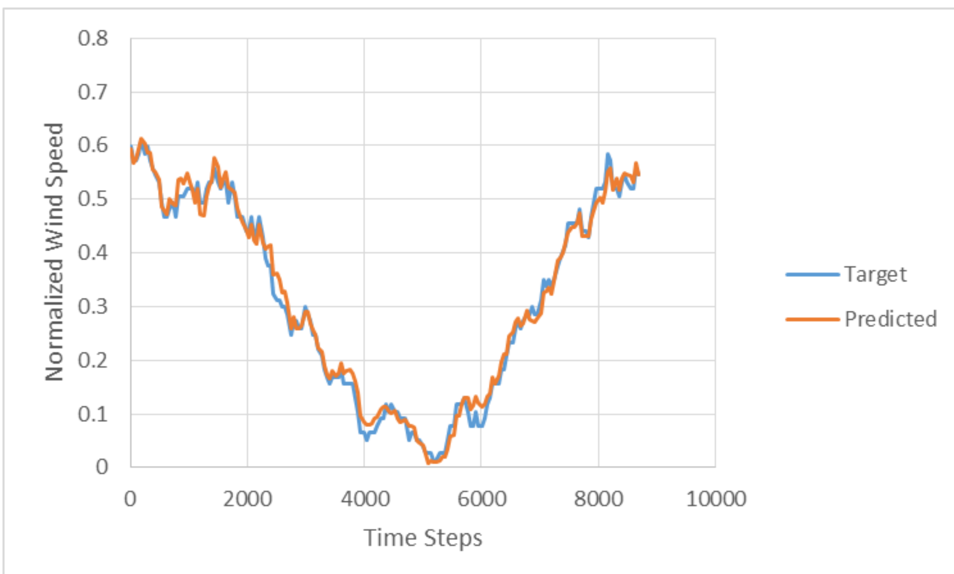
Table 4.13 Summary of paired t-test results for multi-step-ahead prediction of wind speed

Dataset	Method	df	Mean	StDev	SE Mean	95% CI for mean difference	t-statistic ($H0: \mu_D = 0$, $H1: \mu_D \neq 0$)	P-Value
Boston	NAR	10	0.005599	0.000271	0.000086	(0.005082, 0.005626)	44.53	0.000
	NARX	10	0.000245	0.000165	0.000052			
	Difference	10	0.005354	0.000380	0.000120			
BNDA	NAR	10	0.030380	0.002518	0.000796	(0.028481, 0.032089)	37.97	0.000
	NARX	10	0.000095	0.000013	0.000004			
	Difference	10	0.030285	0.002522	0.000798			
JFK	NAR	10	0.024470	0.001819	0.000575	(0.023101, 0.025709)	77.18	0.000
	NARX	10	0.000065	0.000007	0.000002			
	Difference	10	0.024405	0.001823	0.000577			
SIA	NAR	10	0.09104	0.00389	0.00123	(0.08818, 0.09374)	73.91	0.000
	NARX	10	0.00008	0.00000	0.00000			
	Difference	10	0.09096	0.00389	0.00123			

Figures 4.14-4.21 below displays plots of the normalized target values compared with the predicted values corresponding to the NAR and NARX networks for each data set. Figure 4.14 show the NAR and NARX network results for the Boston data set for single step ahead prediction. Figure 4.15 illustrates the NAR and NARX results for the Boston data set for multi-step ahead prediction. Figures 4.16-4.21 follow the same pattern for the Bismarck Municipal Airport, JFK Airport, and Savannah International Airport respectively. While hourly predictions were made, these plots are displayed in time steps of 48 hours to improve legibility of the plots.

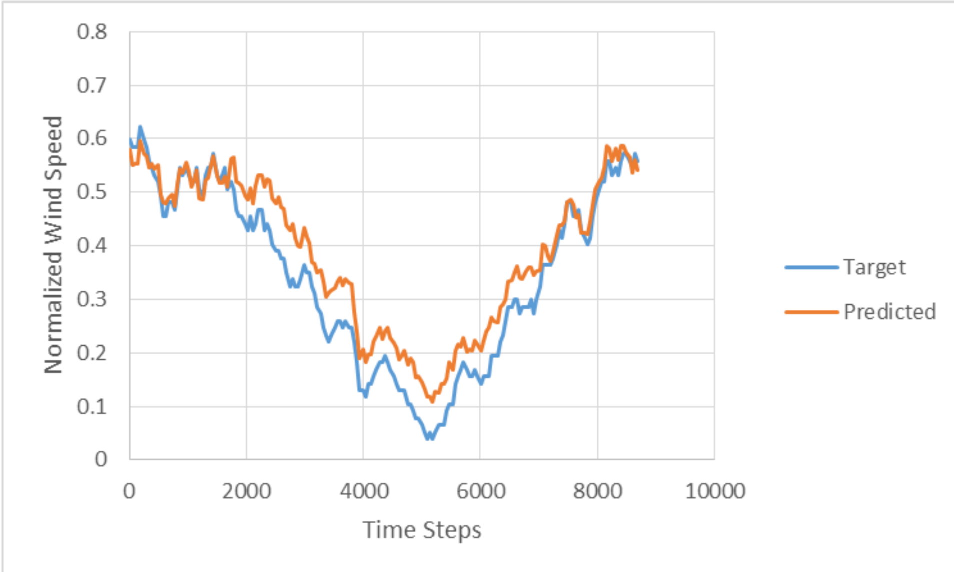


a

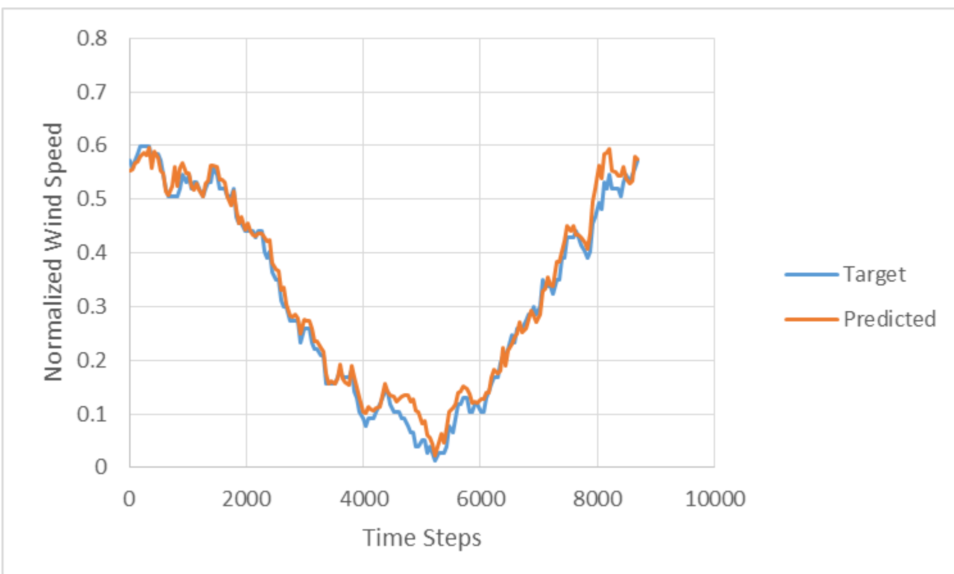


b

Figure 4.14 Single step ahead comparison of target values vs predicted in the Boston (Logan International) data set **(a)** NAR Network **(b)** NARX Network (48 hour time steps)

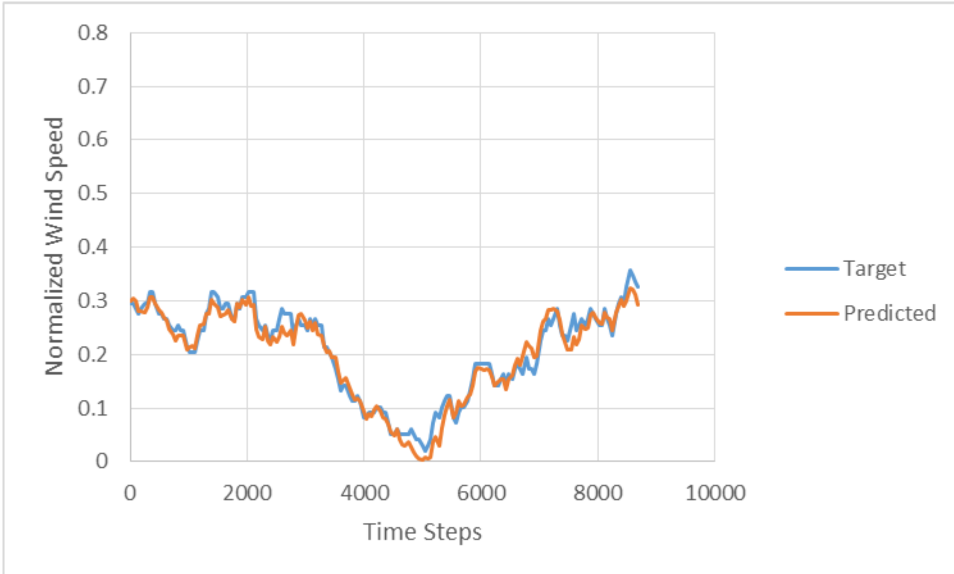


a

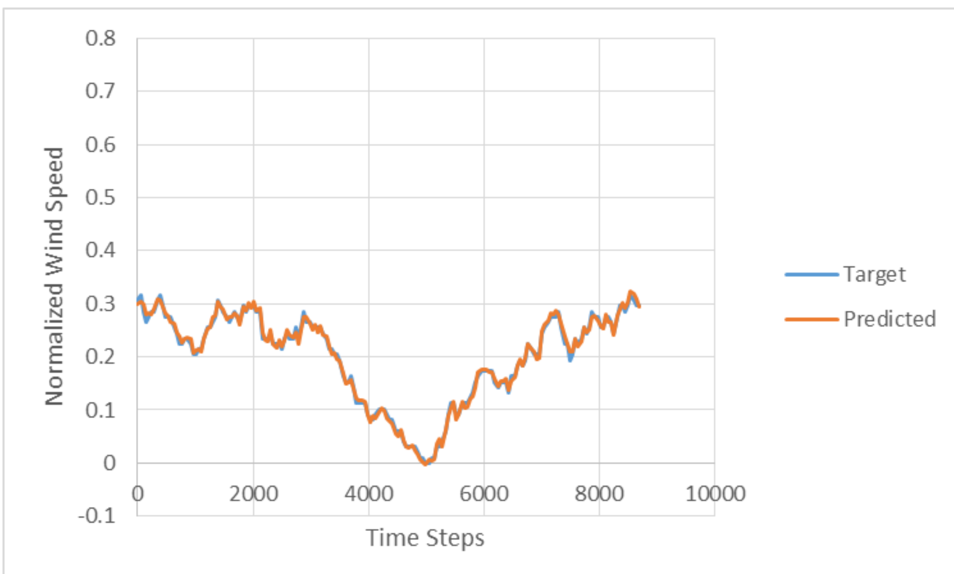


b

Figure 4.15 Multi-Step ahead prediction comparison of target values vs predicted in the Boston (Logan International) data set **(a)** NAR Network **(b)** NARX Network (48 hour time steps)

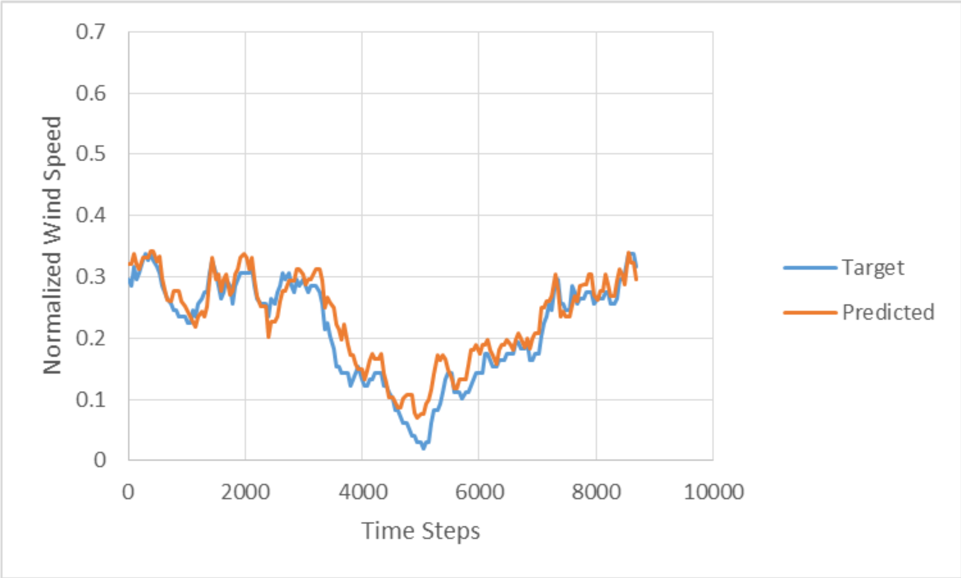


a

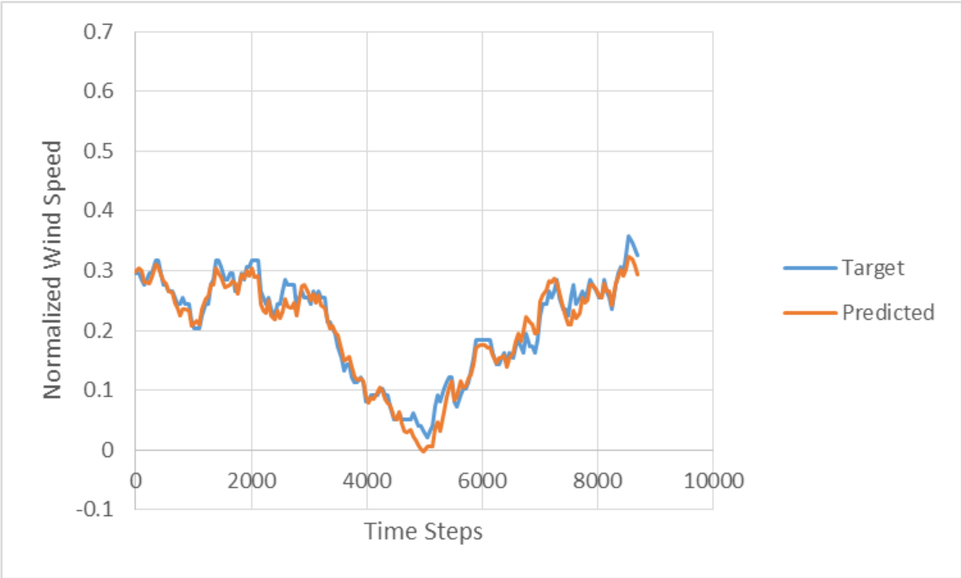


b

Figure 4.16 Single step ahead comparison of target values vs predicted in the Bismarck Municipal Airport data set **(a)** NAR Network **(b)** NARX Network (48 hour time steps)

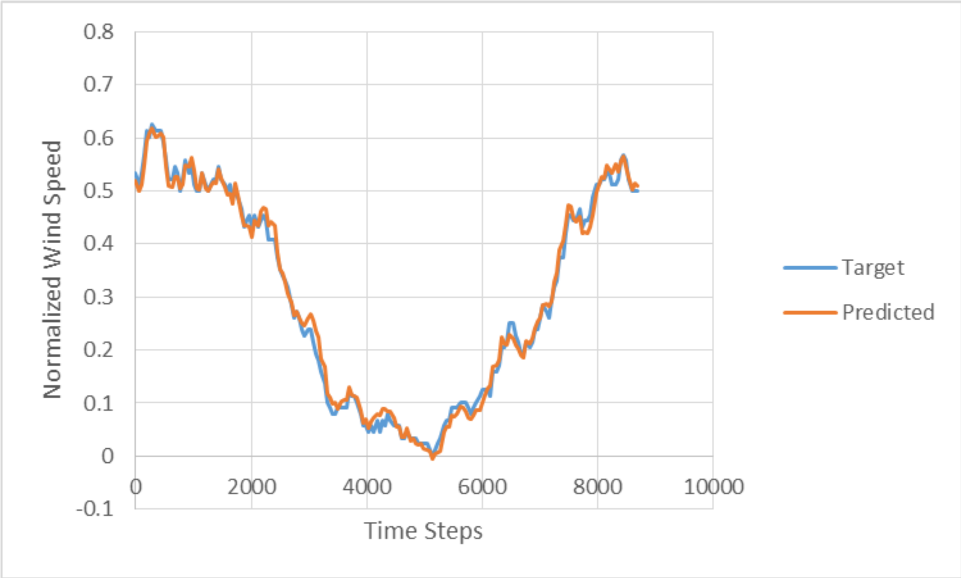


a

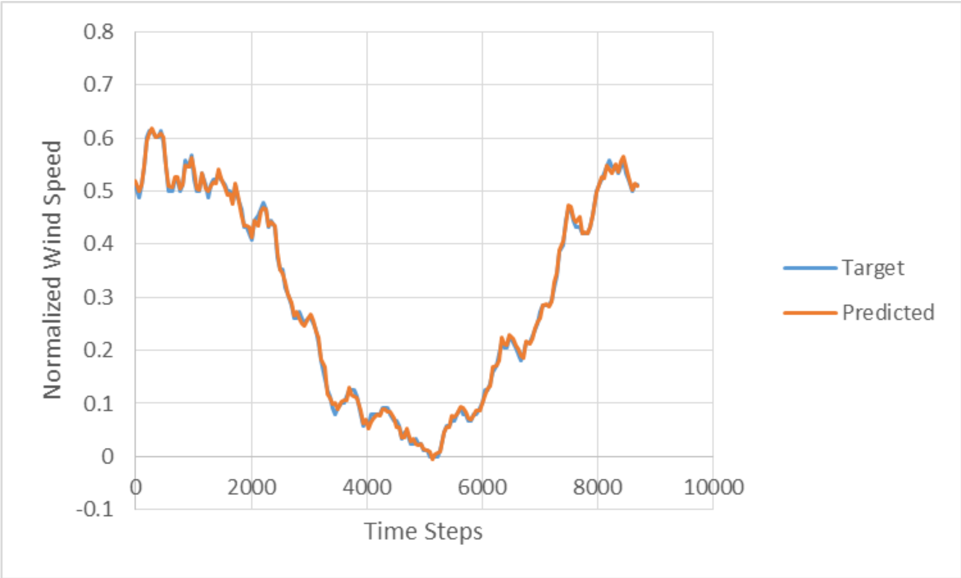


b

Figure 4.17 Multi-Step ahead prediction comparison of target values vs predicted in the Bismarck Municipal Airport data set **(a)** NAR Network **(b)** NARX Network (48 hour time steps)

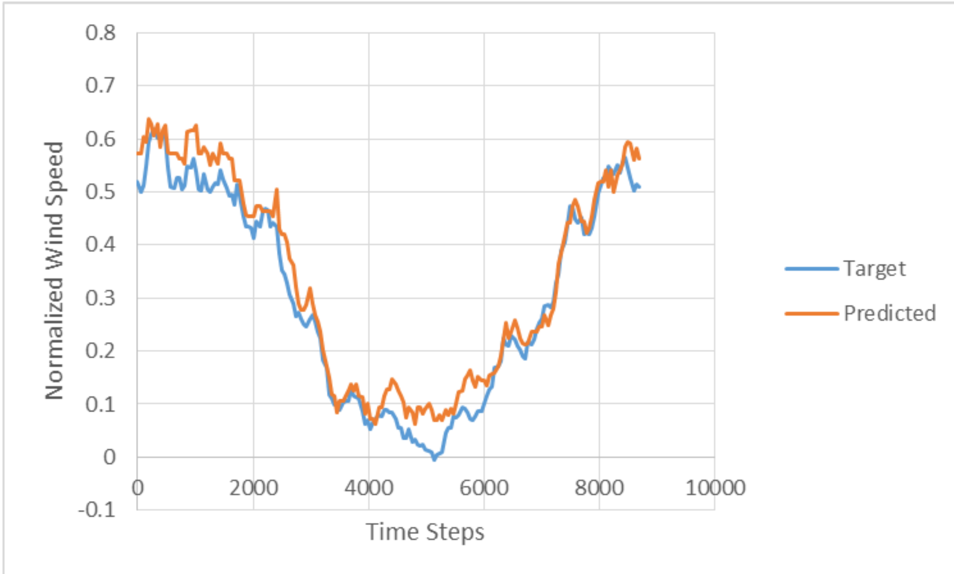


a

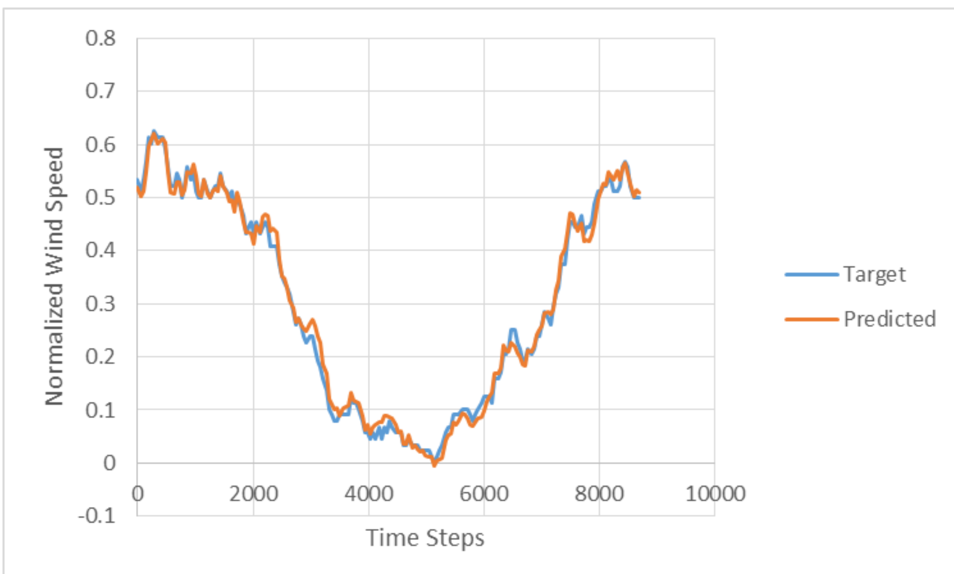


b

Figure 4.18 Single step ahead comparison of target values vs predicted in the JFK Airport data set **(a)** NAR Network **(b)** NARX Network (48 hour time steps)

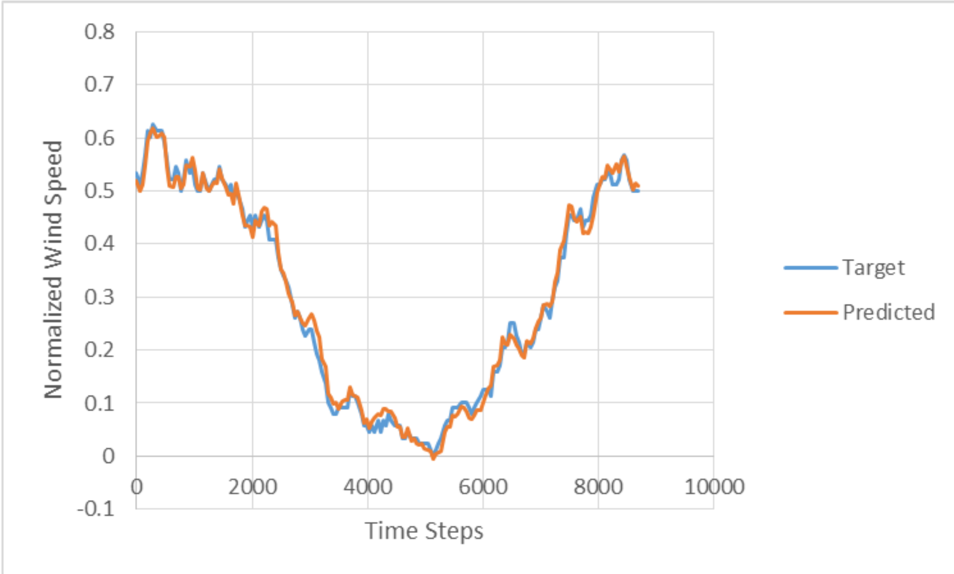


a

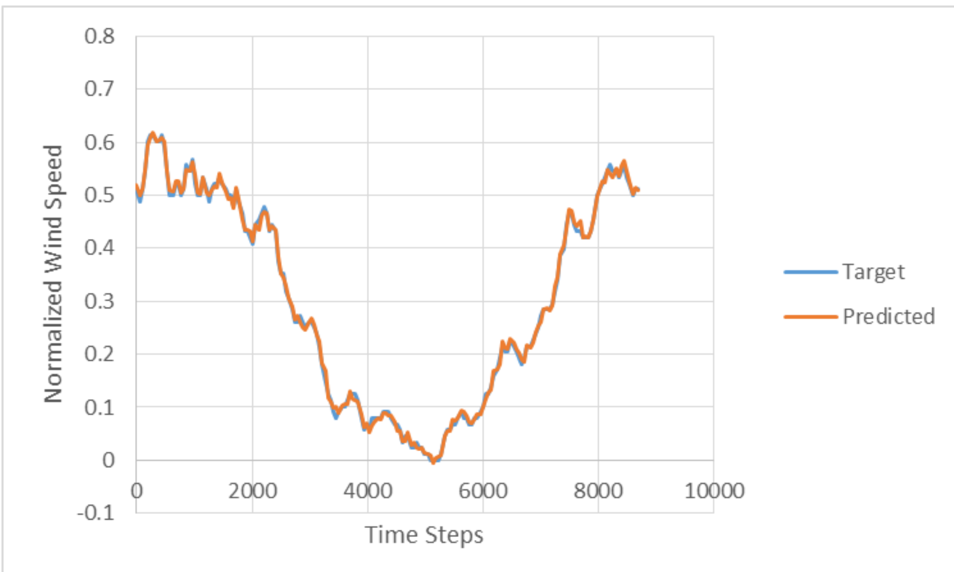


b

Figure 4.19 Multi-Step ahead prediction comparison of target values vs predicted in the JFK Airport data set **(a)** NAR Network **(b)** NARX Network (48 hour time steps)

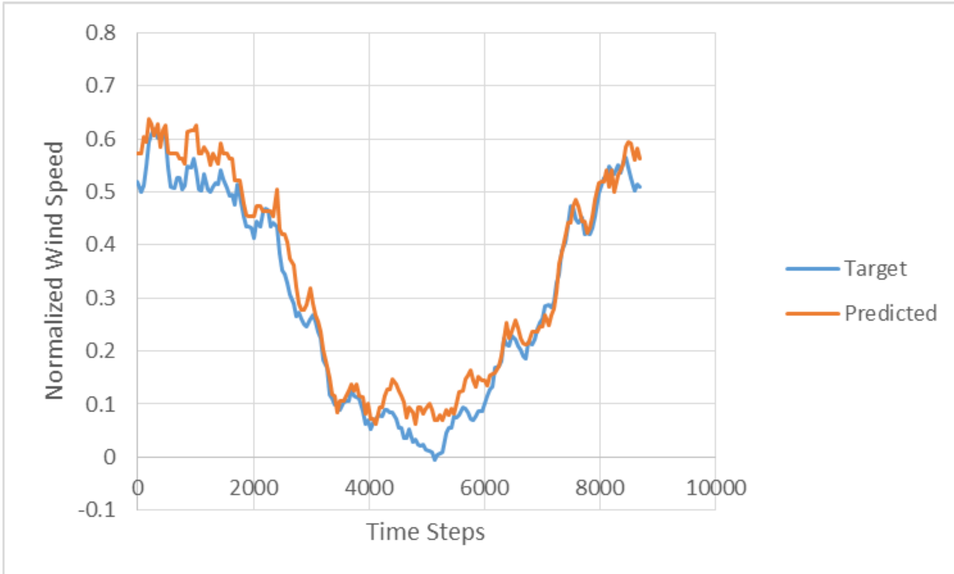


a

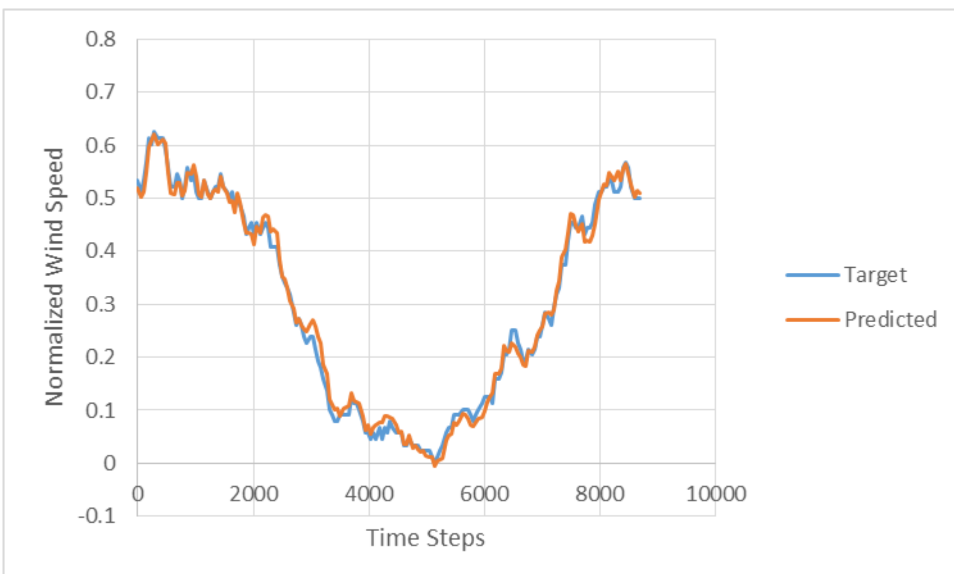


b

Figure 4.20 Single step ahead comparison of target values vs predicted in the Savannah International Airport data set **(a)** NAR Network **(b)** NARX Network (48 hour time steps)



a



b

Figure 4.21 Multi-Step ahead prediction comparison of target values vs predicted in the Savannah International Airport data set **(a)** NAR Network **(b)** NARX Network (48 hour time steps)

4.6 Discussion

4.6.1 Wind Turbine Noise Discussion

This experimentation began with ambient sound measurements at the potential site. Table 3.1 in section 3.1.2 summarizes the data obtained in the data collection stage. The data was comprised of ten two minute measurements of ambient noise data as well as wind speed acquired at the locations for potential wind turbines. The average wind speed varied in the range of 2.20-4.60 m/s, the minimum was in the range of 0.30-1.60 m/s and the maximum was in the range of 5.60-12.10 m/s. The ambient noise level was in the range of 42.5-52.1 dB(A) with peak values in the range of 47.1-65.7 dB(A). This data is significant because the noise data was during a time of acceptable wind levels for power generation.

The noise model was based off of four Bergey Excel 10 turbines with other inputs as specified in section 3.2.2. The noise model included simplifying assumptions as follows: does not account for terrain features and assume ground is flat or constantly sloping, does not take into account vegetation or buildings, and reflections of noise are ignored. Section 4.1.2 present the results of the noise model with table summarizing the results of the model. The model stated that the homes in the area would all receive noise propagation less than 30 dB(A). The max value of noise was calculated at 450 ft. from the turbine and was no greater than 50 dB(A) similar to the ambient noise levels recorded at the site with only ambient noise sources.

Section 4.2 implements a summation of the ambient noise values as well as the predicted noise values. It was noted that when the measured ambient noise levels were summed with the expected noise values of the turbine the increase in loudness was negligible. A worst case scenario 450 ft. away from the turbine with the highest peak ambient noise measurements resulted in an increase of 6.5 - 7.6 dB(A) which would be noticed by a human observer. It was further investigated that in the event of a quiet night in the range of 10-30 dB(A) how would that affect the noise levels. It was found that an extremely quiet night between 10-20 dB(A) (equivalent to the sound of falling leaves to whispering) would produce the right conditions for the wind turbines to be noticed by the observer at the location. Considering vegetation, hills, buildings and other obstacles, it is still unlikely that the noise created by the turbines would disturb residents close in the area of a potential wind farm.

The max wind turbine noise determined by the noise map generated by Openwind was about 50 dB(A) at a distance of 450 ft. (137 m). The GE global study discussed in the literature review presented the illustration seen in figure 2.1. Similar to this study, at a range between 100m-400m they determined an average wind turbine noise levels to be between 50-40 dB(a). After 400 m the wind turbine noise would have been unnoticed by a neighboring observer as the ambient noise recorded at the site was too great. This GE study helps give some validation to the wind turbine section of this study.

4.6.2 Wind Speed Prediction Discussion

This experimentation has concentrated on the comparison of prediction performances of NAR and NARX network models for wind speed forecasting, these networks consider cases where exogenous data is available and when there is not. The first issue to be confronted when selecting the model to be used is the availability of data when trying to perform prediction. Most wind generation efforts will include a meteorological tower that will collect wind speed data as well other categories of data useful for wind speed prediction.

When observing the architecture optimization section of the study it is important to note the importance of the number of delays chosen when optimizing this type of model. Tables 4.6 and 4.8 show that each data set had a different optimized delay parameter and its value cannot be generalized for all datasets. The delay parameter is important as each data set has its own characteristics and behavior as seen in figure 4.10 it was found that the use of external data such a temperature can lower the amount of delays needed to get accurate predictions in every case observed. NARX requires fewer past values and thus NARX prediction models will be simpler.

NAR methods were determined to be useful if only wind speed information was available, and can provide accurate midterm predictions. This simpler alternative to the NARX method can be used with a simple dataset only containing wind speed data. Table 4.6 shows that a NAR model with little information, such as only 2 hours of delay, is not an adequate amount of information to model the data and provides poor results when compared to higher delays. The highest accuracy for the NAR model was averaged at 60 hours of previous data. A higher amount of delays than 28 may be not necessary as the accuracy is basically stabilized at this point as seen in figure 4.10a.

It was determined that a less complex model in regards to the number of neurons in the hidden layer yielded a time series that modeled that target time series well, a large number of neurons in the hidden layer was determined to provide inaccurate results due to the local optima in the network parameters' optimization process during the training stage. Suitable prediction where obtained as seen in figure 4.10b.

NARX methods have illustrated that the inclusion of another simple time series, such as temperature, can help explain anomalies and sudden changes in wind speed and, thus, create a more accurate time series forecast. For example, it can be seen in figure 3.2 that temperature and wind speed have a somewhat inverse relationship. If the temperature were to fluctuate suddenly preemptively to a wind speed change the model would be able to take this into account and thus give a more accurate prediction.

Due to this extra time series the NARX methods tend to need less data in order to get a reasonable prediction. As seen in table 4.8 each dataset required a different delay parameter in order to get optimal results. The average delay for best results was 28 hours of previous data and this is where the data stabilized as seen in figure 4.11a it can also be observed from this figure that when less than 28 hours of data is given in the delays the NARX network performs much better.

The number of neurons present in the NARX network was on average best at three with the Savannah International Airport dataset having the lowest MSE values with six neurons as seen in table 4.9. Network complexity is about the same for both NARX and NAR networks. NARX networks require less historical data to get a more accurate prediction as displayed in figure 4.11b. This shows that the model was able to adjust itself very well to the curve of real data, a better fit than the NAR data.

To summarize the optimization portion of the study figure 4.12 provide a clear graphical representation of the models. NAR methods are good if only wind speed data is available. NAR methods work with a simpler dataset, however more of the historic data is needed to have a good forecast. Conversely, NARX models work with exogenous data, this allows the model to have simpler predictors by imputing additional data. Figure 4.12b illustrates the final results of neuron optimization in the NAR and NARX networks. This figure shows that the NARX network improves upon the NAR network, actually the worst result obtained by the NARX network are still better than the best NAR errors in most cases.

When the optimized networks were implemented in a single step and multi-step ahead prediction model the differences continued to grow. When observing the single step ahead MSE values (table 4.11) it can be seen that the NAR network had values averaging at 6.1994E-04 while the NARX networks had values averaging at 5.561E-05 a difference in another order of magnitude. The NARX network was able to outperform the NAR network when predicting one hour in advance. When observing the Multistep ahead predictions from the same table NAR averaged 3.850E-02 while NARX averaged at 1.043E-04 a difference of two orders of magnitude.

The NARX network significantly outperformed the NAR network when predicting five time steps in advance. This furthers the conclusion that NARX networks require less data in order to get a more accurate prediction that is able to adjust itself very well to the curve of real data, a much better fit than the NAR network. This comparison can be noted graphically in figures 4.6-4.13. The NARX networks are seen to outperform the NAR networks in single step ahead prediction in all locations. This effect is compounded when observing the multistep ahead prediction for all four locations. The NARX network greatly outperformed the NAR network in a five hour ahead prediction.

CHAPTER 5

CONCLUSION

5.1 Summary of Present Work

5.1.1 Wind Turbine Noise Prediction

This experiment collected ambient noise data from a potential wind energy generation site as well as created a noise propagation model for wind turbines at that location. It was found that during a period of time where wind levels were appropriate for power generation the noise generated by the turbines would be outweighed by the ambient noise present at residences close to the site. The noise produced by the wind turbines was predicted to be negligible on all accounts when observed from the residences unless it is an extremely quiet night (10-20 dB(A) or equivalent to the sound of falling leaves to whispering). A 1-19 dB(A) range of increase could be expected under these conditions which would be perceived as much louder than pure ambient noise by a human observer. This is would be a very rare occasion for ambient noise to reach such low levels. However the results presented are a conservative estimate of actual sound levels. The model does not take into account hills or mountains, vegetation, or other obstacles to sound propagation such as buildings and other constructions near the site.

5.1.2 Wind Speed Prediction

This body of work provided a methodology that could be used to forecast wind speed using artificial neural networks. This method collected one year’s worth of hourly wind data in MPH and dry bulb temperature data in degrees Fahrenheit from 4 different locations in the US. The data was then normalized to be in a range between zero and one in order to compare results between different data sets and prevent local maxima from skewing results. After this two different prediction models from the ANN area were chosen for study: NAR and NARX. Both time series prediction models were able to provide suitable results, however it was found that NARX networks where able to predict the time series regression problem more accurately due to the addition of external data. NARX also had the advantage of using less delays meaning less historical data were used in forecasting. When implementing the two networks into forecasting further in the future the NARX network was much more accurate and difference was found to be statistically significant. When forecasting one hour ahead the NARX network had error less than the NAR network in the scale of one magnitude. When forecasting five hours ahead the NARX network had error less than the NAR network in the scale of two magnitudes. This furthers the conclusion that NARX networks require less data in order to get a more accurate prediction that is able to adjust itself very well to the curve of real data, a much better fit than the NAR network.

5.2 Scope of Future Work

In regards to the noise propagation section of the study there are several areas of improvement. The primary improvement would be employing more comprehensive software, such as SPreAD-GIS, in order to produce a more comprehensive noise model, however this comes with increased costs. Secondarily more ambient noise data and weather data could be collected at various different times in order to get a more accurate data set to test. Another major improvement to the study would be to obtain wind turbine sound propagation data, using the sound level meter, in order to compare it to the predicted values. This would allow one to truly determine the accuracy of the model.

The wind speed prediction could potentially be improved by employing different NN techniques. Incorporating numerical weather prediction algorithms could potentially increase the time frame these models are able to predict accurately as well. It would be interesting to include more exogenous data into the model in order to determine if this would increase the accuracy of the current model.

REFERENCES

AWS Truepower. “Openwind® User Manual.” 2014

Aylor, D. “Noise Reduction by Vegetation and Ground.” *Journal of Acoustical Society of America*, (51), 197-205. 1972

Benoit R, Yu W. “Developing and testing of wind power forecasting techniques for Canada.” *Proceedings of the first joint action symposium on wind forecasting techniques. International Energy Agency (IEA).* 2002.

Berge E. “Experiences with wind forecasting techniques in Norway.” *Proceedings of the first joint action symposium on wind forecasting techniques International Energy Agency (IEA).* 2002.

Bergey WindPower. “Acoustic Characteristics of the Bergy Excel 10 kW Wind Turbine.” 2010

Brand A., Kok J. “Wind power by a quarter of the hour.” *Proceedings of the first joint action symposium on wind forecasting techniques.* International Energy Agency (IEA). 2002.

Brüel & Kjær. “Environmental Noise Booklet.” *Brüel & Kjær Sound & Vibration Measurement A/S*. 2000

Camara, A.; Feixing, W.; Xiuqin, L. “Energy Consumption Forecasting Using Seasonal ARIMA with Artificial Neural Networks Models.” *Int. J. Business Management.* 2016.

Cao, Q.; Ewing, B.T.; Thompson, M.A. “Forecasting wind speed with recurrent neural networks.” *Eur. J. Operator.* 221, 148–154. 2012.

Cigizoglu HK, Kisi O. “Flow prediction by three back propagation techniques using k-fold partitioning of neural network training data.” *Nord Hydrologic*. 36(1):1–16. 2005

Climate Data Online. “Normal Hourly Averages 2010, Bismarck Municipal Airport, John F. Kennedy International Airport, Logan International Airport, and Savannah International Airport.” *National Centers for Environmental Information.* 2010

Doucoure, B.; Agbossou, K.; Cardenas, A. “Time series prediction using artificial wavelet neural network and multi-resolution analysis: Application to wind speed data.” *Renew. Energy 2016.* 92, 202–211. 2016

Fadare DA. “The application of artificial neural networks to mapping of wind speed profile for energy application in Nigeria.” *Applied Energy*. 87(3): 934–42. 2010

Fadare, D.A.: “The application of artificial neural networks to mapping of wind speed profile for energy application in Nigeria.” *Applied Energy* 87, 934–942. 2010.

Fang, C., & Ling, D. “Investigation of the Noise Reduction Provided by Tree Belts.” *Landscape and Urban Planning*, (63), 187-195. 2003

Filippeli, M. “Wind Project Design and Optimization: Introduction, Validation, and Applications of Openwind®.” *2nd NREL Wind Energy Systems Workshop*, Jan 30, 2013, Broomfield, CO. 2013.

GE. “Global Research.” *National Institute of Deafness and Other Communication Disorders*. 2012

González, P.A.; Zamarreno, J.M. “Prediction of hourly energy consumption in buildings based on a feedback artificial neural network.” *Energy Build.* 37, 595–601. 2005

Hagan, M.T.; Demuth, H.B.; Beale, M.H. “Neural Network Design.” *PWS Pub. Co.* p. 3632. 1996

Haydari, Z.; Kavehnia, F.; Askari, M.; Ganbariyan, M. “Time-series load modelling and load forecasting using neuro-fuzzy techniques.” *Proceedings of the 9th International Conference on Electrical Power Quality and Utilization.* 2007.

Health Canada. “Wind Turbine Noise and Health Study: Summary of results.” *Government of Canada.* 2014

International Organization for Standardization. “ISO 9613-2: Attenuation of Sound during Propagation Outdoors- Part 2- General Method of Calculation.” *International Organization for Standardization.* 1996

Jursa R. “Wind power prediction with different artificial intelligence models.”+ “Proceedings of the European wind energy conference”. *EWEC2007, Milan, Italy*; 2007.

Kaminsky F, et al. “Time series models of average wind speed within synoptic weather categories.” *Proceedings of the fourth ASME wind energy symposium.* p. 215–9. 1985

Kariniotakis G, Stavrakakis GS, Nogaret EF. “Wind power forecasting using advanced neural network models.” *IEEE Trans Energy Conference.* 11(4): 762–7. 1996

Kiartzis, S.J.; Bakirtzis, A.G.; Petridis, V. “Short-term forecasting using NNs.” *Electric Power Syst.* 33, 1–6. 1995

Kisi O. “Comparison of different ANN techniques in river flow prediction.” *Civil Engineering Environmental Systems* (24):211–31. 2007

Kluijver, H., & Stoter, J. “Noise Mapping and GIS: Optimizing Quality and Efficiency of Noise Effect Studies.” *Computers, Environment and Urban Systems*, (27), 85-102. 2003.

Kwong, W., Zhang, P., Romero, D., Moran, J., Morgenroth, M., & Amon, C. “Wind Farm Layout Optimization Considering Energy Generation and Noise Propagation.” *Proceedings of the ASME 2012 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, Chicago, IL, Aug. 12-15, 2012.

Lamancusa, J. (2009). “Outdoor Noise Propagation.” *Penn State*, http://www.me.psu.edu/lamancusa/me458/10_osp.pdf. 2009

Larsson, C., & Ohlund, O. “Amplitude Modulation of Sound from Wind Turbines under Various Meteorological Conditions.” *Journal of Acoustical Society of America*, 67-73. 2014

Lei M, Shiyan L, Chuanwen J, Hongling L, Yan Z. “A review on the forecasting of wind speed and generated power.” *Renew Sustain Energy* Rev. 13(4): 915–20. 2009

Li, G., Shi J.: “On comparing three artificial neural networks for wind speed forecasting.” *Appl. Energy* 87, 2313–2320. 2010.

Mabel M.C., Fernandez E.: “Analysis of wind power generation and prediction using ANN: a case study.” *Renew. Energy* 33, 986–992. 2008.

Macas, M.; Moretti, F.; Fonti, A.; Giantomassi, A.; Comodi, G.; Annunziato, M.; Pizzuti, S.; Capra, C.A. “The role of data sample size and dimensionality in neural network based forecasting of building heating related variables.” *Energy Build.* 111, 299–310. 2016

MacKay, David, J.C. “Information Theory, Inference, and Learning Algorithms.” *Cambridge University Press*. 2003.

Marquardt, D.W. “An algorithm for least-squares estimation of nonlinear parameters.” *J. Society Ind. Appl. Math.* 11, 431–441. 1963

Mathworks. (2014). “Neural Network Toolbox: User's Guide (r2014a).” *Mathworks.* 2014.

Michaud “Wind Turbine Noise and Health Study.” *Noise News International.* 21(4) 14-23. 2013.

Mohandes, M., Rehman, S., Rahman, S.M.: “Estimation of wind speed profile using adaptive neuro-fuzzy inference system (ANFIS).” *Appl. Energy* 88, 4024–4032. 2011

Mohanty, S.; Patra, P.K.; Sahoo, S.S. “Prediction of global solar radiation using nonlinear auto regressive network with exogenous inputs (narx).” *In Proceedings of 2015 39th National Systems Conference (NSC).* 2015.

Monfared, M., Rastegar, H., Kojabadi, H.M.: “A new strategy for wind speed forecasting using artificial intelligent methods.” *Renew. Energy* 34, 845–848. 2009.

More A, Deo MC. “Forecasting wind with neural networks.” *March Struct*; 16(1):35–49. 2003

Nagy, G.I. “Probabilistic solar and wind power forecasting using a generalized additive tree ensemble approach.” *Int. J. Forecast*. 32, 1087–1093. 2016

National Physical Lab (NPL). “Guide to Predictive Modelling for Environmental Noise.” NPL Acoustic Technical Guides. 2006

Nyanteh, Y.D.; Srivastava, S.K.; Edrington, C.S.; Cartes, D.A. “Application of artificial intelligence to stator winding fault diagnosis in Permanent Magnet Synchronous Machines.” *Electric Power Syst.* 103,201–213. 2013

Pamanikabud, P., & Tansatcha, M. “Geographical Information System for Traffic Noise Analysis and Forecasting with the Appearance of Barriers.” *Environment Modelling and Software*, (18), 959-973. 2003

Reed, S., Boggs, J., & Mann, J. "A GIS Tool for Modeling Anthropogenic Noise Propagation in Natural Ecosystems." *Environmental Modelling & Software*, (37), 1-5. 2012.

Saavedra, Rodolfo & Samanta, Biswanath. "Noise and Vibration Issues of Wind Turbines and Their Impact-A Review." *Wind Engineering.* 39(6). 2015

Sanchez I. "Adaptive combination of forecasts with application to wind energy." *International Journal of Forecast*ing 24(4):679–93. 2008

Sengpiel, E. (n.d.). "Adding Acoustic Levels of Sound Sources." *Sengpiel Audio.*

Sfetsos A. "A comparison of various forecasting techniques applied to mean hourly wind speed time series." *Renewable Energy*; 21(1):21–35. 2000

Tande J, Landberg L. "A 10 sec forecast of wind turbine output with neural networks." *Proceedings of European community wind energy conference.* p. 774–7. 1993.

Tang, U., & Wang, Z. "Influences of Urban Forms on Traffic-Induced Noise and Air Pollution: Results from a Modelling System." *Environmental Modelling & Software*, (22), 1750-1764. 2007

Torres J, Garcia A, Deblas M, Defrancisco A. "Forecast of hourly average wind speed with ARIMA models in Navarre (Spain)." *Sol Energy.* 79(1):65–77. 2005

US Department of Energy. "Wind Vision: A New Era for Wind Power in the United States." *US Department of Energy*. 2015

Wang, X., Guoc, P., Huanga, X.. "A review of wind power forecasting models." *Energy Proceedings.* **12**, 770–778 2011.

Welch RL, Ruffing SM, Venayagamoorthy GK. "Comparison of feedforward and feedback neural network architectures for short-term wind speed prediction." *Proceedings of international joint conference on neural networks.* 2009.

Zhao, X., Wang, Sh, Li, T.: "Review of evaluation criteria and main methods of wind power forecasting." *Energy Procedia.* 12, 761–769. 2011

APPENDICES

Appendix A

Software Implementation of NAR Network in MATLAB 2014a

```
% Solve an Autoregression Time-Series Problem with a NAR Neural Network
% Script generated by Neural Time Series app
%
% This script assumes this variable is defined:
%
%   NormSIAWS - feedback time series.

T = tonndata(NormBNDAWS,false,false);

% Choose a Training Function
% For a list of all training functions type: help nntrain
% 'trainlm' is usually fastest.
% 'trainbr' takes longer but may be better for challenging problems.
% 'trainscg' uses less memory. NTSTOOL falls back to this in low memory
situations.
trainFcn = 'trainlm';  % Levenberg-Marquardt

% Create a Nonlinear Autoregressive Network
feedbackDelays = 1:2;
hiddenLayerSize = 2;
net = narnet(feedbackDelays,hiddenLayerSize,'open',trainFcn);

% Choose Feedback Pre/Post-Processing Functions
% Settings for feedback input are automatically applied to feedback output
% For a list of all processing functions type: help nnprocess
net.input.processFcns = {'removeconstantrows','mapminmax'};

% Prepare the Data for Training and Simulation
% The function PREPARETS prepares timeseries data for a particular network,
% shifting time by the minimum amount to fill input states and layer states.
% Using PREPARETS allows you to keep your original time series data
unchanged, while
% easily customizing it for networks with differing numbers of delays, with
% open loop or closed loop feedback modes.
[x,xi,ai,t] = preparets(net,{},{},T);

% Setup Division of Data for Training, Validation, Testing
% For a list of all data division functions type: help nndivide
net.divideFcn = 'divideblock';  % Divide data randomly
net.divideMode = 'time';  % Divide up every value
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;


% Choose a Performance Function
```

```
% For a list of all performance functions type: help nnperformance
net.performFcn = 'mse';  % Root Mean squared error

% Choose Plot Functions
% For a list of all plot functions type: help nnplot
net.plotFcns = {'plotperform','plottrainstate','plotresponse', ...
  'ploterrcorr', 'plotinerrcorr'};


% Train the Network
[net,tr] = train(net,x,t,xi,ai);

% Test the Network
y = net(x,xi,ai);
e = gsubtract(t,y);
performance = perform(net,t,y);

% Recalculate Training, Validation and Test Performance
trainTargets = gmultiply(t,tr.trainMask);
valTargets = gmultiply(t,tr.valMask);
testTargets = gmultiply(t,tr.testMask);
trainPerformance = perform(net,trainTargets,y);
valPerformance = perform(net,valTargets,y);
testPerformance = perform(net,testTargets,y);



% % % %
% % % % % View the Network
%  view(net)
% % % %
% % % % % Plots
% % % % % Uncomment these lines to enable various plots.
% figure, plotperform(tr)
% figure, plotregression(t,y)

% % % % %figure, plottrainstate(tr)
%figure, plotresponse(t,y)
%figure, ploterrcorr(e)
%figure, plotinerrcorr(x,e)
% % % %
% Closed Loop Network
% Use this network to do multi-step prediction.
% The function CLOSELOOP replaces the feedback input with a direct
% connection from the outout layer.
netc = closeloop(net);
[xc,xic,aic,tc] = preparets(netc,{},{},T);
yc = netc(xc,xic,aic);
perfc = perform(net,tc,yc);
% view(netc)
% Multi-step Prediction
% Sometimes it is useful to simulate a network in open-loop form for as
% long as there is known data T, and then switch to closed-loop to perform
% multistep prediction. Here The open-loop network is simulated on the known
% output series, then the network and its final delay states are converted
```

```
% to closed-loop form to produce predictions for 5 more timesteps.
[x1,xio,aio,t] = preparets(net,{},{},T);
[y1,xfo,afo] = net(x1,xio,aio);
[netc,xic,aic] = closeloop(net,xfo,afo);
[y2,xfc,afc] = netc(cell(0,5),xic,aic);
  multiStepPerformance = perform(net,T(1,predictOutputTimesteps),y2)
% Further predictions can be made by continuing simulation starting with
% the final input and layer delay states, xfc and afc.


% Step-Ahead Prediction Network
% For some applications it helps to get the prediction a timestep early.
% The original network returns predicted y(t+1) at the same time it is given
y(t+1).
% For some applications such as decision making, it would help to have
predicted
% y(t+1) once y(t) is available, but before the actual y(t+1) occurs.
% The network can be made to return its output a timestep early by removing
one delay
% so that its minimal tap delay is now 0 instead of 1.  The new network
returns the
% same outputs as the original network, but outputs are shifted left one
timestep.
nets = removedelay(net);
[xs,xis,ais,ts] = preparets(nets,{},{},T);
ys = nets(xs,xis,ais);
stepAheadPerformance = perform(net,ts,ys)
% view(nets)
% Deployment
% Change the (false) values to (true) to enable the following code blocks.
% See the help for each generation function for more information.
if (false)
  % Generate MATLAB function for neural network for application deployment
  % in MATLAB scripts or with MATLAB Compiler and Builder tools, or simply
  % to examine the calculations your trained neural network performs.
  genFunction(net,'myNeuralNetworkFunction');
  y = myNeuralNetworkFunction(x,xi,ai);
end
if (false)
  % Generate a matrix-only MATLAB function for neural network code
  % generation with MATLAB Coder tools.
  genFunction(net,'myNeuralNetworkFunction','MatrixOnly','yes');
  x1 = cell2mat(x(1,:));
  xi1 = cell2mat(xi(1,:));
  y = myNeuralNetworkFunction(x1,xi1);
end
if (false)
  % Generate a Simulink diagram for simulation or deployment with.
  % Simulink Coder tools.
  gensim(net);
end
```

Appendix B

Software Implementation of NARX Network in MATLAB 2014a

```
% Solve an Autoregression Problem with External Input with a NARX Neural
Network
% Script generated by Neural Time Series app
%
% This script assumes these variables are defined:
%
%   NormBNDATemp - input time series.
%   NormBNDAWS - feedback time series.

X = tonndata(NormJFKTemp,false,false);
T = tonndata(NormJFKWS,false,false);

% Choose a Training Function
% For a list of all training functions type: help nntrain
% 'trainlm' is usually fastest.
% 'trainbr' takes longer but may be better for challenging problems.
% 'trainscg' uses less memory. NTSTOOL falls back to this in low memory
situations.
trainFcn = 'trainlm';  % Levenberg-Marquardt

% Create a Nonlinear Autoregressive Network with External Input
inputDelays = 1:28;
feedbackDelays = 1:28;
hiddenLayerSize = 2;

net = narxnet(inputDelays,feedbackDelays,hiddenLayerSize,'open',trainFcn);

% Choose Input and Feedback Pre/Post-Processing Functions
% Settings for feedback input are automatically applied to feedback output
% For a list of all processing functions type: help nnprocess
% Customize input parameters at: net.inputs{i}.processParam
% Customize output parameters at: net.outputs{i}.processParam
net.inputs{1}.processFcns = {'removeconstantrows','mapminmax'};
net.inputs{2}.processFcns = {'removeconstantrows','mapminmax'};

% Prepare the Data for Training and Simulation
% The function PREPARETS prepares timeseries data for a particular network,
% shifting time by the minimum amount to fill input states and layer states.
% Using PREPARETS allows you to keep your original time series data
unchanged, while
% easily customizing it for networks with differing numbers of delays, with
% open loop or closed loop feedback modes.
[x,xi,ai,t] = preparets(net,X,{},T);

% Setup Division of Data for Training, Validation, Testing
% The function DIVIDERAND randomly assigns target values to training,
% validation and test sets during training.
```

```
% For a list of all data division functions type: help nndivide
net.divideFcn = 'divideblock';  % Divide data sequentially
% The property DIVIDEMODE set to TIMESTEP means that targets are divided
% into training, validation and test sets according to timesteps.
% For a list of data division modes type: help nntype_data_division_mode
net.divideMode = 'value';  % Divide up every value
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

% Choose a Performance Function
% For a list of all performance functions type: help nnperformance
% Customize performance parameters at: net.performParam
net.performFcn = 'mse';  % Mean squared error

% Choose Plot Functions
% For a list of all plot functions type: help nnplot
% Customize plot parameters at: net.plotParam
net.plotFcns = {'plotperform','plottrainstate','plotresponse', ...
  'ploterrcorr', 'plotinerrcorr'};

% Train the Network
[net,tr] = train(net,x,t,xi,ai);

% Test the Network
y = net(x,xi,ai);
e = gsubtract(t,y);
performance = perform(net,t,y);



% Recalculate Training, Validation and Test Performance
trainTargets = gmultiply(t,tr.trainMask);
valTargets = gmultiply(t,tr.valMask);
testTargets = gmultiply(t,tr.testMask);
trainPerformance = perform(net,trainTargets,y);
valPerformance = perform(net,valTargets,y);
testPerformance = perform(net,testTargets,y);
% % %
% % %         % View the Network
%      view(net)
% % %
% % %         % Plots
% % %         % Uncomment these lines to enable various plots.
% figure, plotperform(tr)
% % % %         %figure, plottrainstate(tr)
% figure, plotregression(t,y)
% % %         %figure, plotresponse(t,y)
% % %         %figure, ploterrcorr(e)
% % %         %figure, plotinerrcorr(x,e)
% % %
        % Closed Loop Network
        % Use this network to do multi-step prediction.
        % The function CLOSELOOP replaces the feedback input with a direct
        % connection from the outout layer.
        netc = closeloop(net);
        netc.name = [net.name ' - Closed Loop'];
```

```
%          view(netc)
        [xc,xic,aic,tc] = preparets(netc,X,{},T);
        yc = netc(xc,xic,aic);
        closedLoopPerformance = perform(netc,tc,yc)
        % Multi-step Prediction
        % Sometimes it is useful to simulate a network in open-loop form for
as
        % long as there is known output data, and then switch to closed-loop
form
        % to perform multistep prediction while providing only the external
input.
        % Here all but 5 timesteps of the input series and target series are
used to
        % simulate the network in open-loop form, taking advantage of the
higher
        % accuracy that providing the target series produces:
        numTimesteps = size(x,2);
        knownOutputTimesteps = 1:(numTimesteps-5);
        predictOutputTimesteps = (numTimesteps-4):numTimesteps;
        X1 = X(:,knownOutputTimesteps);
        T1 = T(:,knownOutputTimesteps);
        [x1,xio,aio] = preparets(net,X1,{},T1);
        [y1,xfo,afo] = net(x1,xio,aio);
        % Next the network and its final states will be converted to closed-
loop
        % form to make five predictions with only the five inputs provided.
        x2 = X(1,predictOutputTimesteps);
        [netc,xic,aic] = closeloop(net,xfo,afo);
        [y2,xfc,afc] = netc(x2,xic,aic);
        multiStepPerformance = perform(net,T(1,predictOutputTimesteps),y2)
        % Alternate predictions can be made for different values of x2, or
further
        % predictions can be made by continuing simulation with additional
external
        % inputs and the last closed-loop states xfc and afc.



        % Step-Ahead Prediction Network
        % For some applications it helps to get the prediction a timestep
early.
        % The original network returns predicted y(t+1) at the same time it
is given y(t+1).
        % For some applications such as decision making, it would help to
have predicted
        % y(t+1) once y(t) is available, but before the actual y(t+1) occurs.
        % The network can be made to return its output a timestep early by
removing one delay
        % so that its minimal tap delay is now 0 instead of 1.  The new
network returns the
        % same outputs as the original network, but outputs are shifted left
one timestep.
        nets = removedelay(net);
        nets.name = [net.name ' - Predict One Step Ahead'];
%          view(nets)
        [xs,xis,ais,ts] = preparets(nets,X,{},T);
        ys = nets(xs,xis,ais);
```

```
        stepAheadPerformance = perform(nets,ts,ys)

%         % Deployment
%         % Change the (false) values to (true) to enable the following code
blocks.
%         % See the help for each generation function for more information.
%         if (false)
%           % Generate MATLAB function for neural network for application
deployment
%           % in MATLAB scripts or with MATLAB Compiler and Builder tools, or
simply
%           % to examine the calculations your trained neural network
performs.
%           genFunction(net,'myNeuralNetworkFunction');
%           y = myNeuralNetworkFunction(x,xi,ai);
%         end
%         if (false)
%           % Generate a matrix-only MATLAB function for neural network code
%           % generation with MATLAB Coder tools.
%           genFunction(net,'myNeuralNetworkFunction','MatrixOnly','yes');
%           x1 = cell2mat(x(1,:));
%           x2 = cell2mat(x(2,:));
%           xi1 = cell2mat(xi(1,:));
%           xi2 = cell2mat(xi(2,:));
%           y = myNeuralNetworkFunction(x1,x2,xi1,xi2);
%         end
%         if (false)
%           % Generate a Simulink diagram for simulation or deployment with.
%           % Simulink Coder tools.
%           gensim(net);
%         end
```