Mu'tah University
College of Graduate Studies

Wind Energy Forecasting Using Artificial Neural Network

By
Osama Nizar Al-Sbou

Supervisor
Dr.Ziad Al-Majali

A Thesis Submitted to the College of Graduate Studies in Partial Fulfilment of the Requirements for the Degree of Master in Smart Grids in Electrical Power Systems

Mutah University, 2023

الآراء الواردة في الرسالة الجامعية لا تُعبر
بالضرورة عن وجهة نظر جامعة مؤتة

جامعة مؤتــة كلية الدراسات العليا		MUTAH UNIVERSITY College of Graduate Studies

قرار إجازة رسالة جامعية

تقرر اجازة الرسالة المقدمة من الطالب اسامه نزار احمد السبوع

والموسومة بـ: Wind Energy Forecasting Using Artificial Neural Networks

استكمالا لمتطلبات الحصول على درجة الماجستير في الشبكات الذكية في أنظمة القوى الكهربائية

التخصص: الشبكات الذكية في أنظمة القوى الكهربائية في تاريخ 2023/01/22

من الساعة 12 إلى الساعة 2 قرار رقم 5/2023

أعضاء اللجنــة:	التوقيــع	
د. زياد سلامه ذياب المجالي		مشرفا ومقررا
أ.د حسين ذياب سلامه المجالي		عضوا
د. ايمن عبدالله سليمان اللواما		عضوا
ا.د.عاطف صالح عثمان المشاقبه		عضو خارجي

عميد كلية الدراسات العليا

أ.د مخلد سليمان الطراونة

2023/1/12

جامعة مؤتة كلية الدراسات العليا

Dedication

This work is dedicated to my grandparents Ahmad and Rasmiah, my parents and siblings Islam, Zaid, Dana and Jood

Osama Al-Sbou

Acknowledgments

First, I would like to express my thanks, praise, and gratitude to Allah Almighty for his grace and for his blessings throughout this work to complete it successfully.

I would like to express my truthful appreciation to my supervisor, Dr. Zyad AL-Majali, for his patience and guidance during my study. His enthusiasm and support provided me with a lot of inspiration.

Osama AL-Sbou

Table of Content

List of Tables

List of Figures

List of Abbreviations

ANN	Artificial Neural Network
BR	Bayesian Regularization
CNN	Convolutional neural networks
DNN	Deep Neural Network
LSTM	Long Short-Term Memory
LM	Levenberg Marquardt
MIMO	Multi Inputs Multi Outputs
MISO	Multi Inputs Single Output
MNIST	Modified National Institute of Standards
NARX	Nonlinear Autoregressive with Exogenous Inputs
NORM	Normalized data
R	Regression Factor
RNN	Recurrent Nural Network
SCG	Scaled Conjugate Gradient
TDL	Tapped Delay Line
TDNN	Time Delay Neural Network
RMSE	Root Mean Square Error
MSE	Mean Square Error
NN	Neural Network
Stand	Standardization
WFPO	wind turbine
WT	

Abstract

Wind Energy Forecasting Using Artificial Neural Network

Osama Nizar AL-Sbou

Mutah University, 2023

Accurate modeling and simulation of the dynamic performance of wind turbines is essential to improve their operational performance and progress in the production of sustainable energy. This thesis presents models with accurate simulation of wind turbines using one of the best neural network techniques, which is the dynamic neural network. The network based model has been built Dynamic neural networks through an autonomous neural network provided with external inputs, or what is called NARX .MATLAB software was used to build the NARX model, This thesis is based on a group of readings or data extracted from the wind power plant in Jordan. which are data for several variables taken from the sensor systems at this station. Since the main objective of this thesis is to predict performance and describe system dynamics, Four variables directly related to the operation and performance of the wind turbine were taken into account the first of which is wind speed and direction, temperature and humidity, This data was collected over a period of 24 operating hours for the wind turbine, Accurately, 2042 data sets were taken for each of the previously mentioned variables. The sets of data were prepared using network training and validation methods: Actual data , standardization data and normalization data. This is in addition to conducting extensive experiments for this model by changing and experimenting with all variables that have a significant impact on network performance, which ultimately leads to satisfactory results.

Keywords: wind turbine, Dynamic Neural Network, ANN, NARX model, Dynamic modeling.

الملخص

إعداد: أسامة نزار السبوع

جامعة مؤتة، 2023

محطة طاقة الرياح في الأردن. وهي بيانات لعدة متغيرات مأخوذة من أنظمة الاستشعار في هذه المحطة. وبما أن الهدف الرئيسي من هذه الأطروحة هو التنبؤ بالأداء ووصف ديناميكيات النظام، فقد روعيت أربعة متغيرات مرتبطة مباشرة بتشغيل وأداء توربينات الرياح أولها سرعة الرياح واتجاهها ودرجة الحرارة والرطوبة، وقد تم جمع هذه البيانات على مدار 24 ساعة تشغيل لتوربينات الرياح، بدقة ، تم أخذ 2042 مجموعة بيانات لكل من المتغيرات المذكورة سابقا. تم إعداد مجموعات البيانات باستخدام أساليب التدريب والتحقق من صحة الشبكة: البيانات الفعلية وبيانات التقييس وبيانات التطبيع. هذا بالإضافة إلى إجراء تجارب مكثفة لهذا النموذج من خلال تغيير وتجريب جميع المتغيرات التي لها تأثير كبير على أداء الشبكة ، مما يؤدي في النهاية إلى نتائج مرضية.

الكلمات المفتاحية: توربينات الرياح ، الشبكة العصبية الديناميكية ، ANN ، نموذج NARX ، النمذجة الديناميكية.

Chapter One
Introduction

1.1 Wind Energy

In fact the requirement for energy in the world demand is increasing, so the related research gives a wide range of models for forecasting future energy requirements.

Many factors have pressured the world to use the green energies, such as climatic and environmental. In European, many states have demonstrated their interest in green energy topics. These states have enacted and changed their policies, rules, and laws to come up with the green energy models (Sun *et al.*, 2019).

Some cities are designated as green cities because they have a low level of greenhouse gas emissions(Abdmouleh, Alammari and Gastli, 2015)(Newton and Rogers, 2020).

Some regions have already proposed annual targets to exploit the renewable energy sources to cover a certain percentage of their energy demand. Consequently, renewable energy is considered as a popular issue in this century. The most popular renewable energy sources in use are solar, wind, biomass, geothermal, hydropower, and ocean waves. Approximately, renewable energies provided 27.2% of the world's energy demand on average in 2021, with the percentage expected to rise to 45% by 2040 (Peiris, Jayasinghe and Rathnayake, 2021).

Wind energy can be considered as one of the best solutions for global warming, as it does not produces pollution and does not have greenhouse gas emissions (Peiris, Jayasinghe and Rathnayake, 2021).

Wind is considered as the most favorable renewable energy source because it is the only natural source of energy that is available every where.Its contribution to the increase in global energy demand is expected to increased in the future as the new age of fossil fuels comes to an end(Rahman *et al.*, 2021).

According to the global energy report, wind power generation reached 790 GW of installed capacity in 2020. According to installed capacity, China and the USA are in the leading. Fig.(1.1) shows the amount of wind energy installed globally between 2001 and 2020.

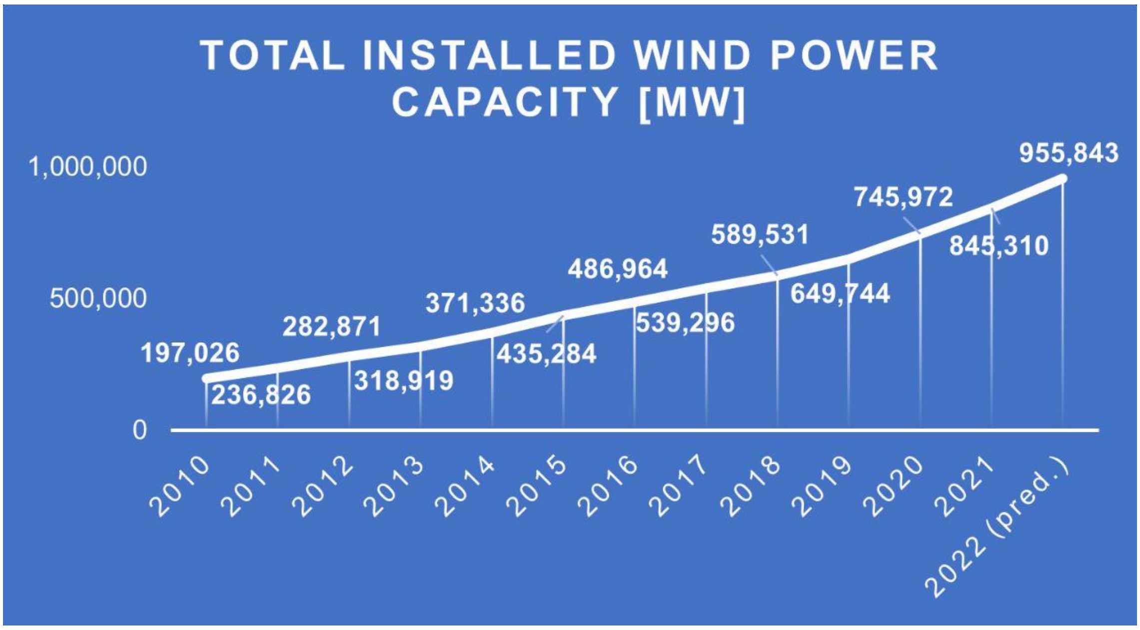


Fig. (1.1) shows the amount of wind energy installed globally between 2001 and 2022 (World Wind Energy Association 2022)

1.2 Wind Energy Forecasting

Wind energy forecasting technology plays an important role; it provides information about the amount of the expected wind power at any point and during the next few days. Wind energy forecasting incorporates in facilitating the mixing of wind energy into the main electricity supply systems. Some difficulties could face the energy mixing process; these difficulties are due the irregularity of wind speed. This causes difficulties with planning and regulation due to sudden changes in wind speed, which affects the accuracy of predictions for the power system. For planning wind power generation and reliability, it is essential to have fast and strong wind speed predictions and responses to system dynamics (Awan and Khan, 2014).

Due to the ability to accurately predict wind speed, nations such as Germany and India are demonstrating a strong desire for the production of wind energy (Iessa *et al.*, 2017).

Wind energy forecasting methods can be categorized as: firstly physical methods and secondly statistical methods. Typically, the former is better at making short- and long-term predictions, while the latter exhibits advantages when making quick predictions.

Artificial intelligence techniques play a crucial role in the prediction process, it involves artificial neural networks (ANN), fuzzy logic, neuro-fuzzy, evolutionary algorithms, and some hybrid techniques, which are more sophisticated methods based on artificial intelligence.

1.3 Problem Statement

Renewable energy resources now play a major role in global energy generation, with wind energy being one of the fastest-growing energy sources due to its environmental and financial benefits. However, as wind turbines improve in terms of flexibility and dependability, it becomes more difficult to monitor turbine-generator power using verified mathematical equations without reliable modeling. . Thus, modeling tools can be investigated in order to simplify turbine monitoring, prediction, and control. Accurate wind power estimate is one of the most difficult challenges in wind power. The variability of prevailing wind speeds over different seasons affects the feasibility of wind availability regions, making predicting total wind availability difficult.

Estimating and predicting wind power is a function approximation challenge. Artificial Neural Networks (ANN).and some other nonlinear programming techniques can be used to solve this problem. An sophisticated computing approach known as artificial neural networks (ANN) may be utilized to accurately anticipate wind power generation. ANN models are created for various situations of wind power estimate and forecasting in this research. These models are built using data from wind monitoring stations and wind turbines that is both global and temporal.

1.4 Research Importance

Wind energy estimate and forecasting is an approach for dealing with wind intermittence. System operators will be able to recognize shifting wind patterns and corresponding wind energy output, allowing them to avoid grid integration problems.

We may use modeling techniques to apply any experiment that would be too expensive or dangerous to undertake in real life and minimize the time it takes to do so; we can even test the design to correct any problems that may arise before the turbine is built.

In this thesis machine learning approaches are employed for time wind power prediction. Machine learning is one of the major branches of computer science, it uses statistical approaches to enable computer programs to learn from data and apply it to some tasks such as prediction. The usage of machine learning improves the performance measure of a task through a training procedure.

Realizing the importance of forecasting wind power output, engineers have developed prediction models using a numerous of statistical, data mining, and machine learning approaches. The research in the field of wind power prediction using artificial neural networks (ANN) is quite popular.

1.5 Research Aims and Objectives

The main goal of this thesis is to use statistics and machine learning approaches to improve forecasting strategies for wind power prediction in complicated terrain. In particular, artificial neural network (ANN) model is employed to determine the most efficient parameters for estimating produced power from projected wind speed. To achieve this, historical data from both observations and projections of climatic factors, as well as turbine availability and associated power outputs, are employed. So the wind power is forecasted for each overlapping pair of months of the year, as well as for the whole year. The overlapping months are used to provide a better transition between predictions made from various training datasets. Furthermore, the data are separated into twelve groups, where each group of two months interval rather than single month, this will increase each sample size and at the same time gives a greater variable range for each training interval. In this thesis each group of months was trained independently. So the experiments are started with observations and then with forecasts as predictors as an input into the ANN. This gives the appropriate model with accurate forecast. The validation of the result of the appropriate model results are based on the comparison to linear-model fits which can prove the capacity of artificial neural network models and to capture nonlinearity effects.

Chapter Two
Literature Review

Modeling and simulation techniques are considered as the successful strategies that can be used to develop wind turbines. These strategies provide the wind turbine with higher efficiency, durability, and reliability. Modeling and simulation techniques can also be employed for condition monitoring, optimization and forecasting, fault detection, sensor validation, and maintenance scheduling. According to these benefits of modeling strategies, the researchers have been motivated to make more study and research in this topic.

In the literature there are many research and studies that concerns in the field of data-driven modeling and simulation of wind turbines. This chapter explores the most significant aspects related to the research field over the last several decades.

Wind power forecasting began in the late 1980s, corresponding with the rising penetration of wind power in electrical power generation. The number of research articles published in this topic has grown since then. This section will explore some of the most related research for this thesis.

Research objectives are prepared after correlating the various works done by contemporary researchers. The majority of the researchers have developed methods for predicting based on wind speed. Many additional criteria necessary to determine the wind energy potential are investigated in addition to wind speed forecasts. For wind power estimation at a specific location, meteorological and climatological data, as well as topographical data, must be used. The power curves of wind turbines must then be plotted against the wind parameters.

Statistical models like the autoregressive (AR) and the ARMA dominated the research on short-term forecasting until the 2000s. Standard statistical time series models in to predict wind power output up to 6 hours ahead of time using an ARMA model.

Their research intends to see if statistical forecasting approaches that do not require any data other than historical wind power generating data are feasible.

Recently, more attention have been concerned to machine learning and hybrid methods, where ANN was used to forecast daily wind power in Germany (Lu *et al.*, 2013).

For training of the ANN, The physical coherence of wind speed and wind power production is learned using previous forecasted meteorological factors and recently observed power data. In a deep LSTM network, the primary components of NWP data such as wind speed, air density, temperature, pressure, and wind direction is used as input data (Qu *et al.*, 2016).

Based on a LSTM method, a deep neural network is used to predict wind turbine power, while the error of short-term power forecasting is evaluated based on a Gaussian mixture model (GMM) (Zhang *et al.*, 2019).

The neural network were used to predict short-term wind power for a wind power plant, where the neural network was trained on historical wind speed and wind direction data. The process of predicting wind power consists of two steps. The first step involves the collection of the raw data which can be filtered by using a probabilistic neural network. This valid data is used in the construction of a prediction model. Secondly, a complex-valued recurrent neural network is used to create a wind power prediction model. This paper describes a method for using neural networks to predict the total output of a wind power plant (WPP). The raw wind data was classified and screened using a probabilistic neural network (PNN) to train neural network prediction models. Then, to simplify the model's input signals, certain representative wind turbines were chosen as an input data source for modeling. Finally, the total output of WPP with high accuracy was predicted based on a complex valued recurrent neural network (CRNN) model which was chosen based on previous wind power prediction experience (Liu *et al.*, 2012).

In (Chinedu, 2019) the author illustrated in the scientists, investors, and policymakers have realized the value of providing near-perfect predictions of green sources. As a result, current research demonstrates advances in methodologies for providing more precise energy forecasting. In this paper the wind energy was linked to variations in wind speed, where the wind speed is irregular parameter in erratic weather. Various model technologies were employed to predict wind power output for different period such as short, ultra-short, medium, and long term predictions, these model technologies involves autoregressive integrated moving average (ARIMA), variants of ARIMA, hybrid models that included ARIMA and artificial neural networks (ANN), Kalman filters, and support vector regressions (SVR).For short and ultra-short terms (two to three hours) it is better to integrate between ARIMA and ANN. On the other hand, for medium-term wind speed predictions SVR, Kalman filters, and their ensembles have demonstrated good performance. For time series predictions, especially for the medium and long term recurrent neural networks (RNN) have achieved enormous success, due to its retentive memory-mapping capabilities in fitting sequence in series. As a result, RNNs are used to improve wind-farm power output prediction(Chinedu, 2019).

Most countries have significant environmental impacts; renewable energy such as wind energy can be employed to reduce these impacts as it can be considered as the most promising solution for reducing these impacts. However, the use of offshore wind energy is growing rapidly to cover the rising in electricity requirements.

The random forest regress or algorithm is used to predict the output power from the wind turbine. The dataset for this experiment was collected from a wind farm in France for two years. In this study, various parameters such as wind direction, wind speed, and outdoor temperature were used as inputs to predict output power. The model used two different capacity factors. The result of the study presented that the estimated mean absolute errors for the proposed model were 3.6% and 7.3% for different capacity factors. The proposed model provided an efficient method for predicting wind turbine output power with a low error(Rashid, Haider and Batunlu, 2020).

Wind power is primarily affected by wind speed. Many approaches such as artificial neural network (ANN), fuzzy logic (FL), and Neuro-Fuzzy, have been proposed to obtain the maximum power point (MPPT) of the wind. G.Q.BAO and Y.F. REN G.Q.BAO and Y.F. REN introduced a variable speed wind generator MPPT depending on ANN. The proposed model combined the generator speed forecasting model with a neural network, where ANN was employed to predict the optimal rotation speed by varying the wind speed and generator speed.

A wind energy control system was proposed; it used a permanent magnet synchronous generator connected to a DC bus via a power converter, also the performance of the control system was evaluated for variant wind speed. The functionality and performance of this method have been confirmed by system simulation results. For small wind turbines, the ANN is presented. System of directly driven permanent magnet synchronous generators The new method has the following advantages over traditional control strategies:(1) The most The mechanical power of a wind turbine can be accurately measured; (2) A neural network based on both dynamic and steady states The wind velocity estimator was created to provide quick and accurate results. (3) This method incorporates generator speed forecasting based on the algorithm and the wind speed measurement model The neural network Only the output voltage is used in the entire system. The rectifier's voltage and current values were tested. This method reduced system failure rates while lowering design costs. The simulation study of the PMSG system validated the theoretical concept of the control system (Ren and Bao, 2010).

Nicolus Kibet Rotich Nicholas illustrated in his thesis wind speeds and directions were designed in order to create models suitable for hourly, daily, weekly, and monthly forecasting. The forecasts were made using Artificial Neural Networks implemented in MATLAB software. Various main types of artificial neural network were built, which are: feed forward neural networks, Jordan Elman neural networks and cascade forward neural networks. For both wind speeds and directions, four sub models of each of these neural networks were created, one for each of the four forecast horizons. Regardless of model type, a single neural network topology was used for all forecast horizons. All of the models were then trained using real-world wind speed and direction data collected over a two-year period in the Finnish municipality of Puumala in Finland. Only 70% of the data was used for model training, validation, and testing, with the remaining 15% being presented to the trained models for verification. The model outputs were then compared to the final 15% of the original data by calculating the mean square and sum square errors. According to the results, feed forward networks produced the lowest generalization errors for hourly, weekly, and monthly wind speed forecasts, while Jordan Elman networks produced the lowest errors when used for daily wind speed forecasting. Cascade forward networks produced the lowest errors when forecasting daily, weekly, and monthly wind directions; Jordan Elman networks produced the lowest errors when forecasting hourly wind directions. The errors were significantly small during model training but skyrocketed during simulation with new inputs. Furthermore, a combination of hyperbolic tangent transfer functions for both the hidden and output layers produced better results than other transfer function combinations. In general, wind speeds were more predictable than wind directions, allowing for more research into improving existing models for wind direction forecasting (Kibet and Nicholas, no date).

Wind energy is considered as a free and easily accessible source of energy. Different states such as Canada integrated the wind power into their power generation systems.

The forecasting of wind power production has many difficulties. These difficulties are due the variation of wind speed as well weather conditions, terrain factors, and turbine height. There are traditional physical and statistical methods for prediction; also there are some advanced artificial intelligence-based methods which have been recently examined to achieve more reliable wind-power forecasts. In Banafsheh Bolouri Afshar's study, (ANN) models were used to determine the most important parameters that affected the wind output generated power in mountainous Canada .In the study, historical data from both observations and forecasts of weather characteristics, as well as turbine availability and reported power production were employed. Experiments were conducted in order to determine the best structure for the artificial neural network. The results of ANN models were compared to linear-model fits to demonstrate ANN models' ability to capture nonlinearity impacts. Another comparison is made between the results of artificial neural network models and the current operational strategy used by a utility company. A three-layered feed-forward back-propagation ANN model with eight hidden neurons was chosen as the architecture. The ANN method can improve day-ahead wind-power forecasts by up to 56% compared to the current operational approach, according to verification results using an independent dataset (Bolouri Afshar, 2016).

An artificial neural network (ANN) is a type of information-processing system that uses the neural structure of the human brain to analyze data, detect patterns, classify, and forecast using a sequence of mathematical equations. An artificial neural network is made up of numerous layers of neurons (nodes). Each layer receives multiple inputs to its nodes and then sends the information to the next layer after performing certain mathematical computations. As a result, each layer feeds information to the next layer — this is the fundamental principle of a feed-forward neural network. In an artificial neural network, there are three types of layers: input, hidden, and output. The capacity to tackle nonlinear problems is the primary benefit of an artificial neural network with numerous layers over a single-layer model (Svensson, no date)(Tu, 1996).

A hybrid neural network-based approach was suggested for simulation that combines ARIMA and ANN. The neural networks based on ARIMA and ANN are run on a month's worth of wind speed time series data. Statistical errors were computed, showing that the model can properly estimate the wind speed of a new site (Cadenas and Rivera, 2010) .

ANNs have been effectively used in a variety of application engineering areas, including function approximation, pattern association, and associative memory (Abiodun *et al.*, 2018). ANNs have been effectively used in a variety of application engineering areas, including function approximation, pattern association, and associative memory. They also show fault tolerance and robustness (Kibet and Nicholas, no date). The mapping of numerous inputs to a single input is referred to as function approximation. Statistical approaches estimate parameter values to tackle these difficulties. When ANNs are used as function approximation solvers, the parameters of the output function are approximated using a model. Activation functions are used in such models to connect inputs and outputs. Various activation functions will be selected and tried by the programmer or designer of ANN.

Chapter Three
Artificial Neural Networks

Since the 1950s, when Bernard Widrow of Stanford University presented the first artificial neural network, researchers have been encouraged to find optimal ANN-based solutions for designing, manufacturing, developing, and operating new generations of computers industrial systems as efficiently, dependably, and durably as possible (Ahmad *et al.*, 2022).

The first stage in the system identification and modeling process is obtaining enough information about the system to be modeled. A clear definition of the modeling aims is also required for the development of an efficient model. Modeling and simulation of industrial systems can be used for condition monitoring, fault detection and diagnosis, sensor validation, system identification, and the optimization and design of control systems (Ahmad *et al.*, 2022).

So far, a number of analytical and experimental approaches to industrial system modeling have been proposed. One of the novel approaches for system identification and modeling of wind and gas turbines is the use of ANN-based techniques. ANN is able to solve a wide range of complex problems. It is possible to do function fitting, approximation, pattern recognition, clustering, image matching, classification, feature extraction, noise reduction, extrapolation (based on the historical data), dynamic modeling, and prediction (Asgari, 2014). This chapter gives a brief overview of artificial neural networks, focusing on the two proposed neural network methodologies, NARX and CNN techniques, as well as the main idea behind them and the main elements, structures, and training and validation algorithms that are used to fit their architectures.

3.1 Artificial Neural Network (ANN)

The primary goal of developing artificial neural networks was always to mimic the human brain in order to solve challenging problems in a wide range of scientific fields such as engineering, psychology, linguistics, philosophy, economics, neurology, and more (Asgari, 2014). (ANN) is a computational system made up of simple, highly interconnected processing elements (neurons) with linear or nonlinear transfer functions [104].These neurons process information by adapting their dynamic state in response to external 31 inputs. Nervous system is divided into layers, that include an input layer, a hidden layer or layers, and an output layer.

The number of neurons and layers in an ANN model is determined by the complexity of the system dynamics. An ANN learns the relationship between the system's inputs and outputs through an iterative process known as training. Each neuron input has its own associated weight (Asgari, 2014).These weights are adjustable values which are computed during the training step to achieve the best performance of the network. In order to create a reliable and accurate design, it is critical to select the appropriate parameters for ANN inputs and outputs. The availability of information and data for the selected parameters, system knowledge for identifying interconnections between different parameters, and the goals for creating a model are all important variables in determining acceptable inputs and outputs. Sensitivity analysis can be used to evaluate the accuracy of the output parameters that have been chosen.Figure 3.1 shows the typical ANN's general structure, which includes three inputs, two outputs, and four neurons in two hidden layers.

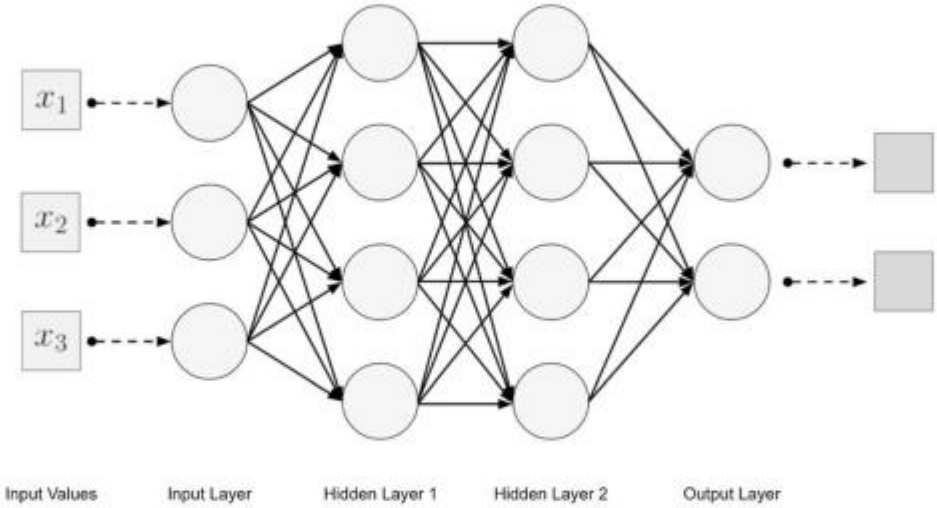


Figure 3.1: the typical ANN's general structure

This study, on the other hand, aims to model the dynamics behavior of a practical wind turbine using the two aforementioned ANN methodologies. As a result, it is more informative to focus on the NARX ANN and deep CNN-based modeling methods, and the following two sections provide a thorough overview of these two techniques.

3.2 Nonlinear Autoregressive with Exogenous Inputs (NARX) Dynamic Neural Network

NARX, or nonlinear autoregressive model with exogenous inputs, is a type of dynamic recurrent neural network used mostly for complex nonlinear modeling of different systems. Autoregressive indicates that the network output parameters are computed from the network's previous inputs and outputs as well as the network's current input and output variables. In other words, each layer of a dynamic neural network contains a recurrent connection with a time delay. This enables the NARX model to propagate data forward and backward, from later stages of processing to earlier ones, resulting in an infinitely dynamic response to the input data.

Feed forward neural networks, in contrast to static neural networks, lack feedback components and delays; the output is determined solely by the current value of the network's input.

NARX networks' dynamic topology create them excellent for mapping inputs to outputs of nonlinear dynamic systems like as wind turbine power plants. The NARX model's mathematical expression is shown in Eq (3.1)(Liu *et al.*, 2020).

$$\hat{y}(t) = f\begin{bmatrix} u(t), u(t-1), \ldots., u(t-n_u), \ldots, \\ y(t-1), \ldots., y(t-n_y) \end{bmatrix} + e(t) \qquad (3.1)$$

Where ŷ(t) and y(t) are the actual and estimated output variables, respectively; $u(t)$ is the input variable of the network; u_n is the time delays of the input variables and n_y is the tapped delay time of the output variables; and $e(t)$ is the model error between the target and prediction. In other words, y and u are the equation's output and externally selected parameters, respectively. y(t) is the dependent output signal's next value, that is regressed on last values of the output signal and an independent (exogenous) input signal. In Eq. (3.1), the function f(..) should have been parameterized. It demonstrates that a search should have been performed across the GT parameters specification to find a function that matches the wind turbine data .A NARX model can be implemented by using a feedforward neural network to approximate the function f(.) , Since the machine learning function classes are able to adapt enough to mimic a variety of functions. A two-layer feedforward network is used for the approximation, as shown in the network's diagram in Fig. 3.2.A vector ARX model, with multidimensional input and output, is also supported by this implementation (Howard and Mark, 2004).

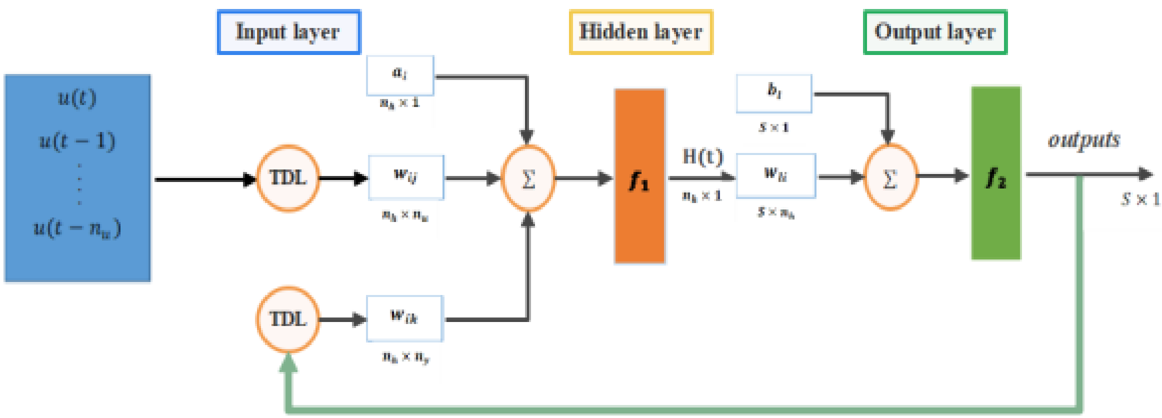


Figure 3.2: The structure of dynamic NARX ANN with mathematical functions

The network is fed with the input and output values from the past using the tapped delay line (TDL). The output from the hidden layer at time t is calculated using the input variable u(t) in Equation (3.1), the typical formation, and the figure above as well as the following(Liu *et al.*, 2020) :

$$H(t)_i = f_1\left[\sum_{j=0}^{n_u} w_{ij}\, u(t-j) + \sum_{k=1}^{n_y} w_{ik} y(t-k) + a_i\right] \qquad (3.2)$$

Where wij is the weight of the connection between the ith hidden neuron and the input neuron u(t - j); Wij is the strength of the connection weight between the ith hidden neuron and the delayed output feedback loop;ai is the bias of the neurons in the hidden layer. The hidden layer transfer function is f1(.),that is, an activation function that can be used to determine the output value of a hidden layer (Liu *et al.*, 2020).where, in the proposed code, a tangent sigmoid function has been used as a hidden layer activation function to conduct the non-linear transformation to the input and thus allowing it capable of learning and performing in a more effective manner, i.e. to avoid any complexity while implementing any back-propagation techniques for weights and biases adjusting. The mathematical expression of the tangent sigmoid function (Liu *et al.*, 2020) is shown in Eq. (3.3):

$$f_1 = \frac{1}{1+e^{-2s}} - 1 \qquad (3.3)$$

By integrating the hidden layer outputs as shown in (Liu *et al.*, 2020), the final NARX prediction value network can be obtained. Equ(3.4)

$$\hat{y}_1(t) = f_2\left[\sum_{i=1}^{n_h} w_{li}\, H(t)_i + b_l\right] \qquad (3.4)$$

Where w_{li} is the weight of the connection between the i^{th} hidden neuron and the l^{th} estimated output n_h; b_l is the bias i^{th} forecasted output; The number of hidden neurons is denoted by nh; and f2(.) is the output layer activation function which will parameterize the forecasted value of the output. The mathematical representation of the linear activation function f2 is shown in Eq. (3.5) (Liu *et al.*, 2020):

$$f_2 = X \qquad (3.5)$$

Where X is the variable which will be replaced by the output of the hidden layer, implying that the output of the hidden layer will only be multiplied by one in the output layer.

Nonlinear Autoregressive models with exogenous input (NARX model) and Recurrent Neural Network (RNN model) were also two models that are widely used in system identification, time series forecasting, and system control (Khan and Ahmad, 2021), but the Recurrent Neural Network does not have feedback connections from the output to the input

Only neurons in the hidden layer have a feedback connection (Khan and Ahmad, 2021).However, using the hidden layer transfer function (Khan and Ahmad, 2021), any NARX model can be transformed into an RNN model.

Before demonstrating NARX network training, a critical training configuration must be explained. The output of the NARX network can be assumed of as an estimate of the output of some non-linear dynamic system, such as Wind Turbine system that this study is trying to model.As shown in Fig 3.3 (Howard and Mark, 2004), the estimated outputs are fed back to the feedforward neural network's input data as part of the typical NARX structure. However, because the actual output values have become available during the network's training phase, it will be more accurate and reliable to design the NARX network in series-parallel structure, i.e. open loop mode, so that the true outputs are used instead of feeding back the feedforward neural network with the estimated ones, as shown in Fig (3.3). This has two advantages. The first advantage is that the feedforward network's inputs are more accurate. The second benefit is that the resulting network is purely feedforward and can be trained using static back-propagation.

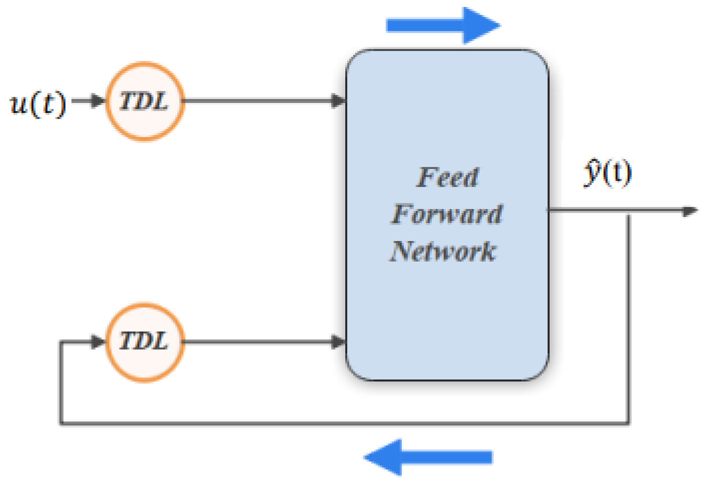


Figure (3.3):closed-loop mode NARX model i.e. with parallel structure

Finally [illegible] th independent (exogenous) inputs, making it an excellent choice for forecasting. Furthermore, feedback connections encircle several layers of the network in a NARX model. When handling with time series forecasting, a NARX network can be organized as a feedforward time delay neural network (TDNN) without the feedback loop from the expected delayed outputs. In this case, it can significantly improve system performance. More information on the dynamic NARX ANN will be provided in the following chapter.

3.3 Deep Convolutional Neural Network (CNN)

Convolutional neural networks (CNNs) have become state-of-the-art in computer vision issues in the last decade (e.g., see (Lecun, Bengio and Hinton, 2015) for many examples of successes and a discussion of the history of deep learning).They were proposed in 1989 [109] and are based on biological brain structure for visual processing, in which a single neuron is only engaged by a small region of input (a "local receptive field").Handwritten digit recognition is now one of their first real-world applications (Edelen *et al.*, 2016), but is still taught in introductory courses today using the Modified National Institute of Standards (MNIST) handwritten digit dataset (Edelen *et al.*, 2016).Despite their early debut, CNNs did not achieve contemporary state-of-the-art achievement in computer vision until the early to mid-2010s. CNNs learn to process images through the use of hierarchical features.

Early layers, for example, learn to recognize basic lines and curves, whereas later layers learn more intricate compound structures (e.g., eyes and noses in the case of facial recognition). [110] contains a few examples.

Among the various deep learning approaches, CNN is one of the most widely used. CNN has shown significant superiority in fault feature extraction and is thus widely used in industrial fault diagnosis. Liu et al. (Liu *et al.*, 2018) used a CNN model to detect faults in a wind turbine and gas turbine combustor and had good detection performance. However, there is no evidence that it cannot be used to simulate the dynamic behavior of a WT& GT; thus, this study presents a novel methodology based on the CNN technique for time-based modeling purposes in order to study the dynamic behavior of a practical wind turbine generation unit and evaluate the capability of this technique for such applications. Figure 3.4 illustrates a simplified representation of the basic CNN concept, while Figure 3.5 illustrates the basic convolution process. Individually, in the CNN layer, the learned filters (or "kernels") are convolved across the image (e.g., left-to-right, top-to-bottom), and the dot product (element-wise multiplication and summation) of the filter matrix and image patch for each yields the actual output matrix value (Edelen *et al.*, 2016).Each layer in a CNN network does have some number of unique filters (e.g., 16, 32 are the most common options), these filters stride through to the image in two main ways, either scanning the image pixel-by-pixel and otherwise skipping a range of pixels each time, and the most common filter size choices are 33 and 55.The output of a specific intermediate layer is then converted into another set of matrices (also known as "feature maps") using a non-linear activation function (the most common choice is the ReLU activation function). Padding the datasets (adding null pixels around the outside) may also be used to increase the output size of the dataset after the convolution process, which improves their ability to extract significant features. CNN hyper-parameters (number of layers, filters, filter sizes, pooling procedures, and so on) are typically computed using the image's predicted features (e.g., estimated feature size, sparsity, and so on) and empirical tuning (Edelen *et al.*, 2016). The main idea behind hyper parameter tweaking is to take a publicly-acceptable and reliable CNN-based structure and set of weights which have been trained on large datasets (many of which are now freely available) and adapt it to a specific issue by simply changing the last few layers of weights on new data (Edelen *et al.*, 2016). (e.g., see (Edelen *et al.*, 2016)). This enables the application of smaller data sets by utilizing previously learned basic filters.

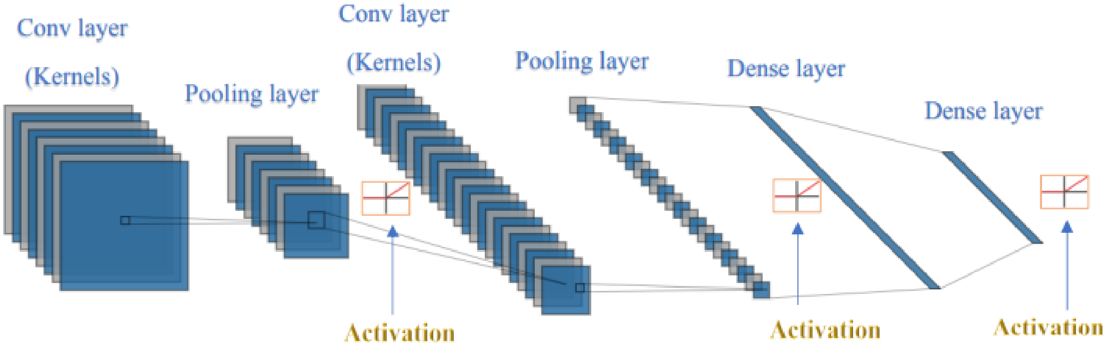


Figure (3.4): CNN Structure

Cascaded filters are convolved over input datasets to produce output features. The output size can then be minimized by using pooling layers (for example, trying to take the maximum of a set of adjacent features or averaging a set of adjacent features).

After several iterations of convolution and pooling processes, a few densely connected layers with decreasing numbers of nodes are typically formed. Finally, an output layer provides the network's final output, which could be a series of numbers coding for specific objects detected in the image by the neural network. However, as this study intends, the output layer can also be used for regression and time-series prediction. More information on the regression using CNN will be provided in the following chapter.

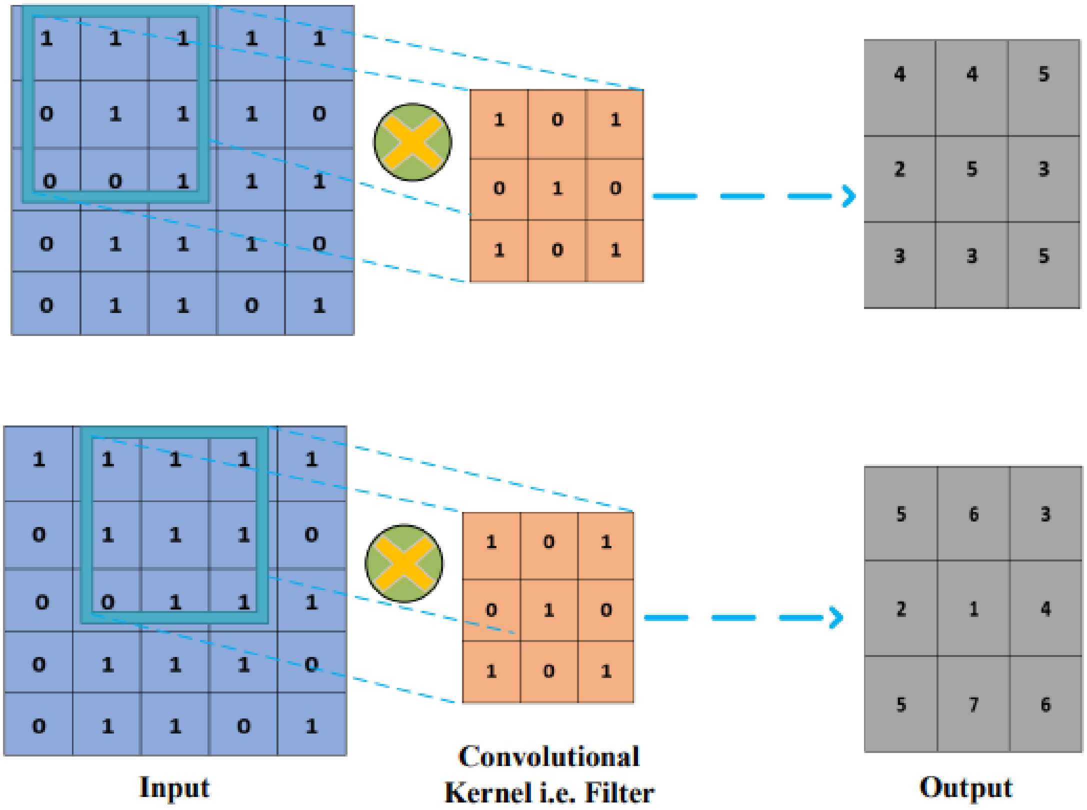


Figure (3.5):convolution operation

Figure 3.6 depicts the first two steps of a convolution process in an input dataset. The dataset features are convolved with a filter, and the dot product of the filter and dataset patch (element-wise multiplication and summation) yields a processed feature output. This example has a 55 input, a 33 filter, and a stride of 1. The stride value specifies that the filter moves one step horizontally between rows and one step downward at the beginning of each subsequent row. Pooling operations work similarly, but they take the maximum or average value in a batch.Max pooling (see Fig. 3.6) can obtain the most significant features. In this study, the max pooling operation is used.

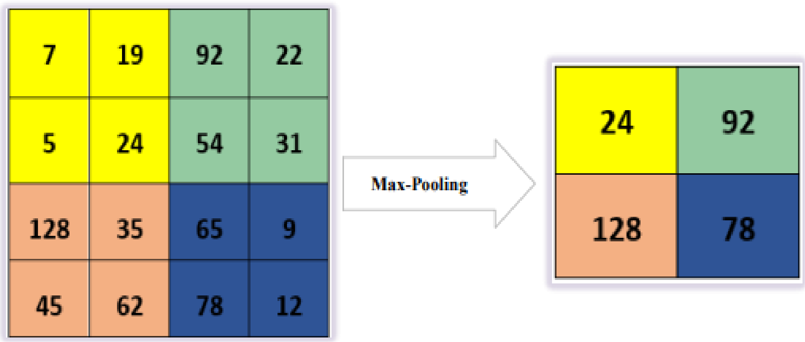


Figure (3.6): Max-pooling operation.

More information about the CNN activation function, CNN training algorithm, and fully connected layer will be covered in the following chapter.

Chapter Four
Methodology

The wind turbines are divided into four inputs and three outputs. The goal is to create the best model by which the inputs are mapped to its corresponding outputs precisely. The reliability of the obtained must be taken into consideration during the training process and when selecting the best model. In order to reach this goal, NARX and CNN constructions have been studied across a wide range of trials as well as the resulting models have been compared in terms of high precision, reliability, and generalization. For this objective, several CNN and NARX designs with hyper-parameter adjustment were used and analyzed across a large variety of trials, i.e. trial and error criteria were used. Given the huge amount of gas turbine data, four out of five records are utilized for training, the other record is employed to validate and test the model. To accomplish this, Matlab software used was used to construct the required NARX model to implement experiments.

4.1 ANN Model Set up

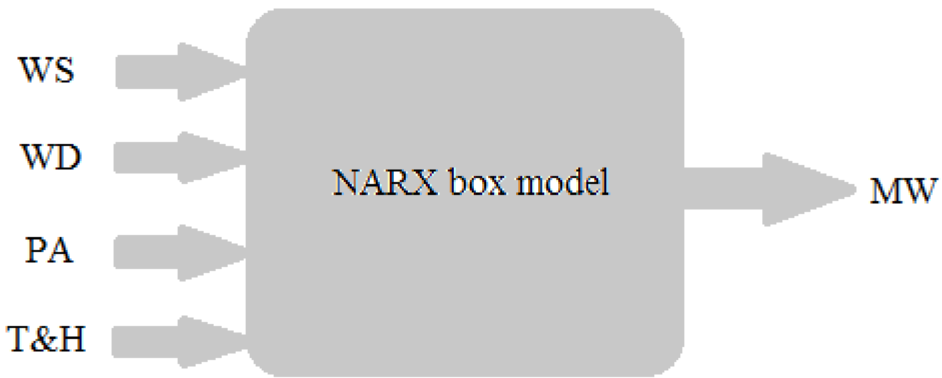
ANN has often been considered as one of the potential option techniques as a data-driven model. Because of the many network designs, training procedures, types of activation functions, neurons number, hidden layers, biases and weights, ANN models for wind turbines can be developed utilizing a variety of ways(Asgari, 2014).

4.2 Data Acquisition and preparation

Data points and information were collected as discrete time signals from the Ma'an wind station in southern Jordan, this data was taken for one wind turbine every ten minutes and the power generated from this turbine is approximately 2 MW which were collected from 1st December in 2020 to 1stJanuary in 2021. One of these data set was used for a training phase and the other two sets was used for verification and testing the model. The inputs to the system were also identified from a control point of view, i.e., wind speed, direction, pitch angle, finally temperature (℃) and humidity. The output is output power in MW of the turbine.

After the definition of both input and output variables based on the collected datasets, related data was separated into 2 groups. This includes training and testing groups which will facilitate the assessment of the generated model and avoid over-fitting while training phase.

The first set of the collected data was employed to train and prepare the model. The second set was used to assess the models' performance by which these data was not included in the training phase. The architecture of the system is illustrated in figure 4.1.



Fig(4.1): input and output parameter of the model in this study

Scale variations between model parameters may enhance the complexity of the modeled task (Ahmad *et al.*, 2022). Because of the enormous input and output numbers, certain models may learn high weights which will typically create unstable models. This means that these models, and during training, may perform badly which will make them sensitive to any change in inputs. As a result, models with high MSE will be created. As a result, in the data pre-processing stage, a features-rescaling approach must be used to the WT's variables.

The most often used pre-scaling approaches are standardization and normalization. Both techniques preprocess data properties in a way that enables the model to accurately map every input to its corresponding output. Nevertheless, the way each approach works differs, and each has its own set of applications. In this study, the gathered data from the WT station were resampled to obtain more data points, and then they were preprocessed into two groups. After that, standardized and preprocessed data (min-max normalization) as well as normalized data are used for training and validation of the proposed model. In the followings, brief discussion of the resampling and pre-processing approaches so that the reader may understand what happens and why the given data is resampled, standardized, and normalized.

4.2.1 Data Resampling

The WT datasets obtained are considered long-term data, to every sample lasting 10 minutes in the actual world. In this research, the data was rescaled in order to gain more datasets for training the NARX model. The resampled data was created by recalculating the time frame period of ten min in thirty second increments. This is accomplished by sending the data to the Simulink working space with a sample frequency of 1/120. This will be divided into 20 samples per 10 minutes, with each sample representing thirty seconds.

4.2.2 Data Normalization

Normalization is a process of scaling data to be in the range of either 0 or 1 or the range of -1 and +1. Normalization is used if a big discrepancy in the given data set. In addition, this normalization is necessary if the collected data do not follow a specific distribution, such as a Gaussian distribution. As a result, because it does not require any data distribution, this method may be greatly effective in building the ANN algorithm. This technique is usually called min-max scaling. The mathematical formula used to normalize given data is based on Equation (4.1) (Ahmad *et al.*, 2022).

$$x_N = \frac{x - x_{minimum}}{x_{maximum} - x_{minimum}} \qquad (4.1)$$

Where x_{min} and $x_{max}$are the largest and lowest values in the given model's data features, respectively. Based on the aforementioned equation, it is clear that the range of data for every variable will be in the range of 0 and 1. In addition, using the equation (4.1), there are3 main conditions:

1. If x is the same as the lowest value, x_N will be zero
2. when x equals the maximum value , then x_N equal 1
3. if x is among(max and min) , the x_N will be among zero and one

4.2.3 Data Standardization

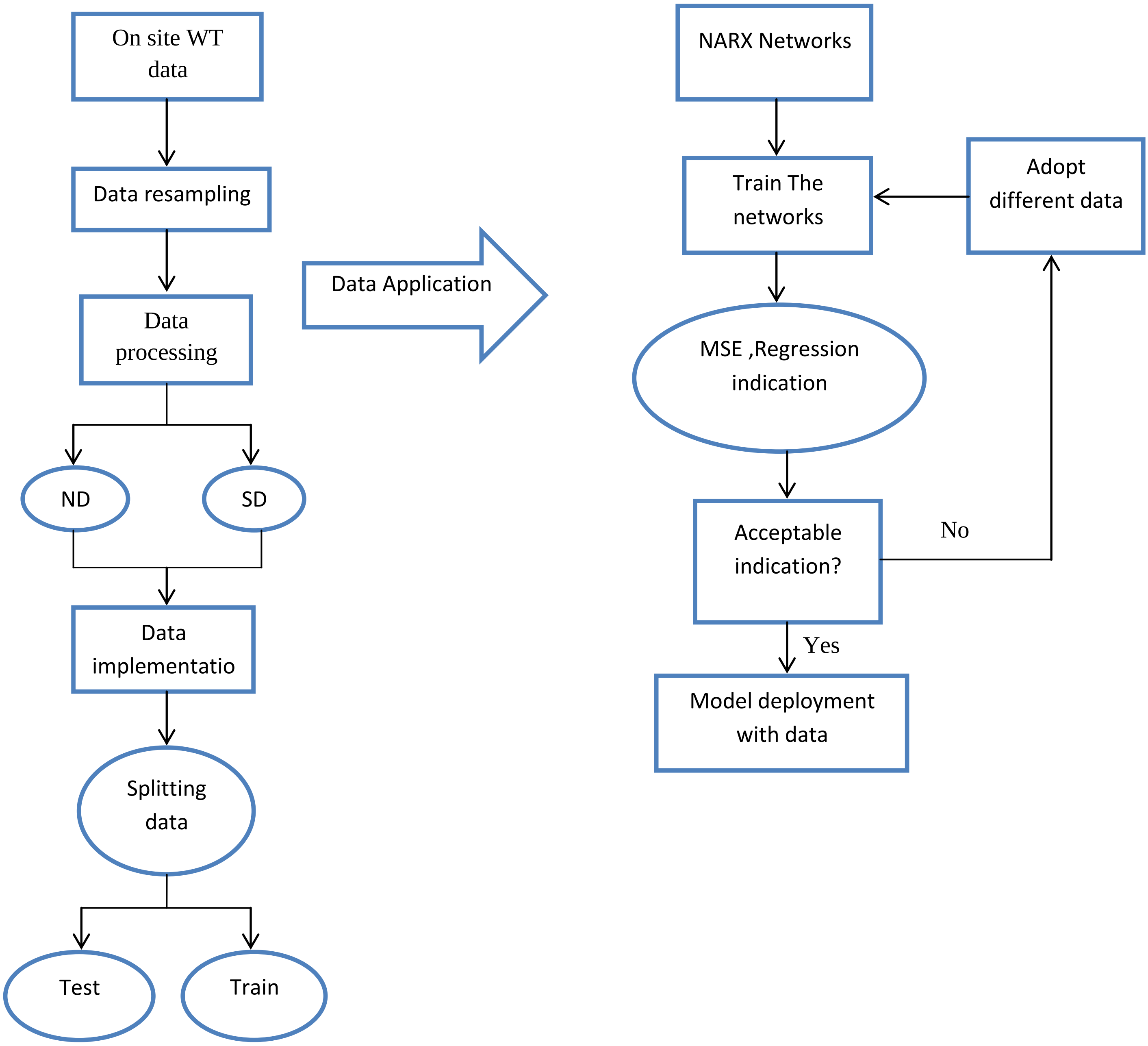
Another frequent rescaling method is to rescale the dataset to be approximately equal to means but with standard deviation equals to 1. Based on this, the mean will be zero and the ensuing distribution will have standard deviation of 1. Standardization, in contrast, may be advantageous if the dataset has a Gaussian distribution. In addition, unlike normalization, standardization process is not limited or restricted to a given range. This means that, when the obtained data contains outliers, standardization has not an influence on these outliers. Equation (4.2) depicts standardization technique's related formula (Ahmad *et al.*, 2022).

$$x_S = \frac{x - \mu}{\delta} \qquad (4.2)$$

Where δ is the standard deviation of the given data and μ is the data mean.The preceding equation shows that the output and input value are not limited to a specific range.

Moreover, the use of standardization or normalization will be determined by the type of dataset and the training-based approach used. In fact, no rule is used to state if data must be standardized / normalized.

To obtain best results, actual, standardized and normalized data were used in comparing the performance between the 3 types of the data preprocessing. This may be useful criteria in the employment of a WT power model. In Figure 4.2, it is focused to the model construction technique in this study.



Fig(4.2): flowchart which shows the model building in this study

4.3 The NARXS Model

Various designs may be evaluated across a large variety of trials to put up a precise WT power plant NARX system with adequate prediction accuracy, much as other dynamic neural networks. Inputs and outputs (the multi inputs multi-outputs (MIMO)) or multi-inputs single outputs (MISO) architecture, training techniques; hidden layers number; neurons number in every hidden layer; activation functions; epochs maximum number (iterations); feedback connections number; inputs delays or time delays in these feedback connections. These all different factors have an influence of the structure. In addition, the structure of data, in other words, data format, is added as an essential element in this study. The NARX model constructed based on the above parameters is shown in Fig 4.3. Here, the Tapped Delay Line (TDL) is employed to provide the previous values of both inputs and outputs to the network. As shown in the Figure, the proposed NARX model includes 4 inputs, 1 hidden layer, 1 output.

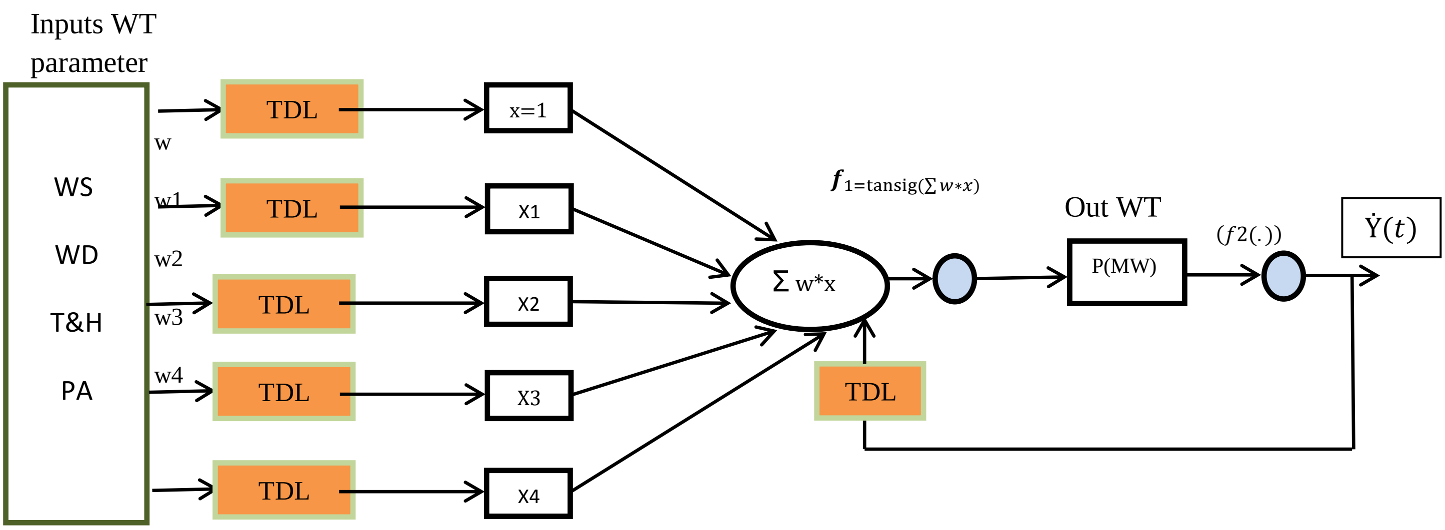


Fig 4.3: The structure of NARX model developed for the WT generation

In which the variables (x-x_4) are the inputs, and the weights connection are (w-w_4), and the f_1 symbolized for tangent sigmoid, and the f_2(.) represents the linear function symbol, and the Ẏ(t) is the expected output which is feedback to the system by the TDL and weighing connection. MATLAB programming code was used to build the NARX model with the complex generalization features. MathWorks created and developed MATLAB, a widely used programming environment for numerical computation in scientific and engineering industries. Several hyper parameters are included in the resulting code to train and customize the developed models for modeling the wind turbine generating station.

The following parameters which were used in building the NARX model which includes: hidden layer neurons Number, time delays, model structure (MIMO and MISO configurations), in addition to the data set kind (standardized, normalized, or real) are all variables that will be modified to get the best design. In the generated code, each of the aforementioned parameters has been taken into consideration as a combination of several settings. Additionally, our study uses a feed-forward multilayer dynamic ANN structure of1 input, 1 hidden, and 1 output layers using both a linear activation function a tangent sigmoid functions for the output layer. Furthermore, several NARX architectures were trained using the developed code, which uses 3 techniques in the training phase which are: Levenberg-

Marquardt (LM) technique, Bayesian regularization technique (BR), and Scaled Conjugate technique (SCG). Eventually, after adjusting all of these settings and parameters as well as the training method, the optimal performance and associated NARX model are obtained.

As stated in the developed code by the network performance function, the NARX training phase includes changing the weights and biases to enhance the NARX performance. Usually, the MSE will be used as the performance measure for feed-forward networks. MSE, as given below in Equation 4.3, measures the difference between the predicted outputs and the actual outputs [114]:

$$E = MSE = \frac{1}{n}\sum_{i=1}^{n}(e_i)^2 = \frac{1}{n}\sum_{i=1}^{n}(k(t) - \acute{k}(t))^2 \qquad (4.3)$$

Generally, two ways are used to perform training: whether gradually or in batches. After every input is applied to the NARX, the gradient is computed then weights are adapted in an incremental mode. Before the weights are changed in the batch process, each one of the inputs of the training dataset is fed to the network. With the train command in the MATLAB software, batch training with the three aforementioned training methods has been used in this study.

Any conventional numerical optimization approach may be used to train the NARX structure, however the three applied optimization techniques are selected because they have demonstrated an outstanding performance for training of ANN networks. These optimization techniques either based on Jacobian of network errors with respect to weights or based on the gradient of network performance relative to network weights. Both, Jacobian and gradient are determined with the back-propagation (BP) approach that includes performing calculations backward through the network.

It might be difficult to predict that training approach would be most effective in a certain situation. The amount of data in the training phase, the weights and biases numbers used, the target MSE, as well as if the developed NARX is employed for pattern recognition or function approximation are all variables that influence the regression. To get the best performance and the appropriate NARX network, the proposed model of the wind turbine power station was trained via wide varieties of trials, together with three distinct optimization approaches.

4.3.1 Taxonomy of the Used Training Algorithms

The process of selecting an ideal bias and weight for the ANN is called training. To achieve this, error between both the output of the ANN and the proposed target must be identified, which needs to be minimized relative to the weights. The BP training approach for feedforward neural networks (FFNN) is the main topic of the thesis. The supervised learning technique for MLP FFNN employs BP techniques. The name hints to the error BP that occurs while training the network. These machine learning techniques use the chain rule repeatedly to determine the partial derivative of the error function with regard to weights of the network. This is achieved by initial calculations at the output layer and then propagating them back in order for each weight connection to be adapted independently. Depending on difficulty of the problem and the network architecture, there are various variations of BP training algorithms; each one has its own advantages and weakness.

As mentioned earlier, our study will employ three kinds of training algorithms. These algorithms include LM, SCG, and BR. In addition, a comparison among them will be conducted in order to determine their performance and accuracy in predicting ahead of WT's parameters. These training algorithms are described mathematically in the following section.

4.3.1.1 Levenberg-Marquardt (LM) algorithm

The LM technique was created to achieve the 2nd order training speed without the requirement to calculate Hessian matrix, much like the quasi-Newton methods. Hessian matrix can be estimated as in Eq. (4.4) where the performance function is a sum of squares (customary in training FFNN) (Howard and Mark, 2004).

$$H = J^t J \tag{4.4}$$

Then gradient may then be obtained using eq. (4.5)

$$g = J^t e \tag{4.5}$$

In the above equation, (e) represents a vector of network errors. J represents a Jacobian matrix. It is far easier to compute the Jacobian matrix using a conventional back propagation method than it is to determine the Hessian matrix. This Hessian matrix estimation is used by the LM method in the following Newton-like upgrade, where x is an indicator of connection weights.

$$x_{k+1} = x_k - [J^T J + \mu I]^{-1} J^T e \tag{4.6}$$

This is merely Newton's approach, employing the approximate Hessian matrix, where scalarμ is zero which varies gradient drop with a small step size whenμ becomes large. The target is to switch as fast as possible to Newton's method due its quicker and accurate approaching to minimum errors. Based on this,μ is decreased after each good step and then raised when an uncertain step would get better performance. The network error (i.e., performance) function will always be decreased in this way and at each iteration. Compared to traditional gradient descent methods, the LM optimization model is more effective (Demirbaş and Çakır, 2019). The LM technique showed to be the best for training moderately sized NARX networks (i.e. few hundred weights), which is easily implemented in through the command train lm.

4.3.1.2 Bayesian regularization algorithm

A training algorithm called Bayesian regularization(BR)is based on modification the bias and weight values according to the LM technique [118](MacKay, 1992). For the BP to create a network, it first determines the optimal combination of squared errors and weights. In this algorithm, the training objective function, F(ω) and stated by (Khan *et al.*, 2020), is modified by BR to include network weights as given in Eq. (4.7).

$$F(\omega) \; = \alpha E_{\omega} + \beta E_{D} \tag{4.7}$$

Where E_D is the total of network errors and $E_{\omega}$is the squared total of network weights.The objective function variables are both β andα .The weights of the network are handled as random variables in the BR approach, and the distribution of both the network weights , training and testing sets is regarded as a Gaussian distribution .

The Bayes' theorem is used to define the both β and α parameters. Here, two variables are related by the Bayes' theorem, Based on A and B's prior (or marginal) and posterior (or conditional) probabilities, respectively, as shown in (4.8)(Khan *et al.*, 2020):

$$\mathrm{P(a|b)} = \frac{\mathrm{P(b|a)P(a)}}{\mathrm{P(b)}} \tag{4.8}$$

P((a|b)) denotes the posterior probability of (a) given that (b) exists, P((a|b)) denotes the prior probability of b given that (a) exists, and P(b) denotes the non-zero prior probability of event (b), that also serves as a normalizing constant. The objective function has to be minimized to find the ideal weight space that is equivalent to enhancing the posterior probability function specified in (4.9)(Khan *et al.*, 2020):

$$\mathrm{P}(\alpha,\beta|d,m) = \frac{\mathrm{P}(d|\alpha,\beta,m)\mathrm{P}(\alpha,\beta|m)}{\mathrm{P}(d|m)} \tag{4.9}$$

Where, (d) is the weight distribution, (m) is the specific ANN, P(d|m) is the normalization factor, P($\alpha,\beta,$ |m) is the uniform prior density for the regularization parameters, and P(d| α,β, m) is the probability function of (d) for given$\alpha,\beta,$ M.The probability function P(d| α,β, M) is maximized in the same way that the posterior function P(α,β |D, M) is maximized. This procedure yields the best values forα andβ for a particular weight space. The algorithm then enters the LM phase, in which Hessian matrix calculations are performed and also the weights are updated as necessary to minimize the forecasting. If the convergence condition is not satisfied, so the algorithm will estimates new values for α and β andrepeats the process until convergence is achieved (Khan *et al.*, 2020). Finally, the train br function in the MATLAB software effectively implements the BR training algorithm.

4.3.1.3 Scaled Conjugate Gradient (SCG) algorithm

The legacy BP algorithm varies the weights toward the negative gradient direction, or toward the extreme negative value. This means that the performance function is also decaying in a fast manner in the same direction. Despite the fact that this function ultimately degrades when the gradient is negative, it turns out that this doesn't always lead to the fastest convergence (Ahmad *et al.*, 2022). In conjugate gradient methods, the error mitigation attained in each preceding step is preserved while a search is conducted in a direction which typically provides a fast convergence compared to that of the steepest descent direction (Choudhary and Chauhan, 2014).The conjugate direction is what this movement is known as. Every iteration in the majority of CG methods involves adjusting the step size. To find the step size that will eliminate the performance function along that line, a search is performed on the conjugate gradient direction. All CG methods start their search along the direction of the steepest descent (Eq4.10) .Commonly, line search is frequently combined with CG algorithms. This shows that instead of computing the Hessian matrix in order to find the best distance to keep moving on the present search direction (Eq. 4.11) (Khan *et al.*, 2020), where the step size will be determined by a line search approach. Afterward, it is revealed that the next search direction is conjugate along the previous search direction (Eq. 4.12) (Khan *et al.*, 2020). Typically, the previous search direction and the new steepest descent path are combined to determine the current search direction .

$$p_o = -g_o \tag{4.10}$$

$$p_k = -g_k + \beta_k p_{k-1} \tag{4.12}$$

$$x_{k+1} = x_k + \alpha_k g_k \tag{4.11}$$

The method used to calculate the factor $\beta_k$serves as a defining characteristic of the various CG methods (Khan *et al.*, 2020).

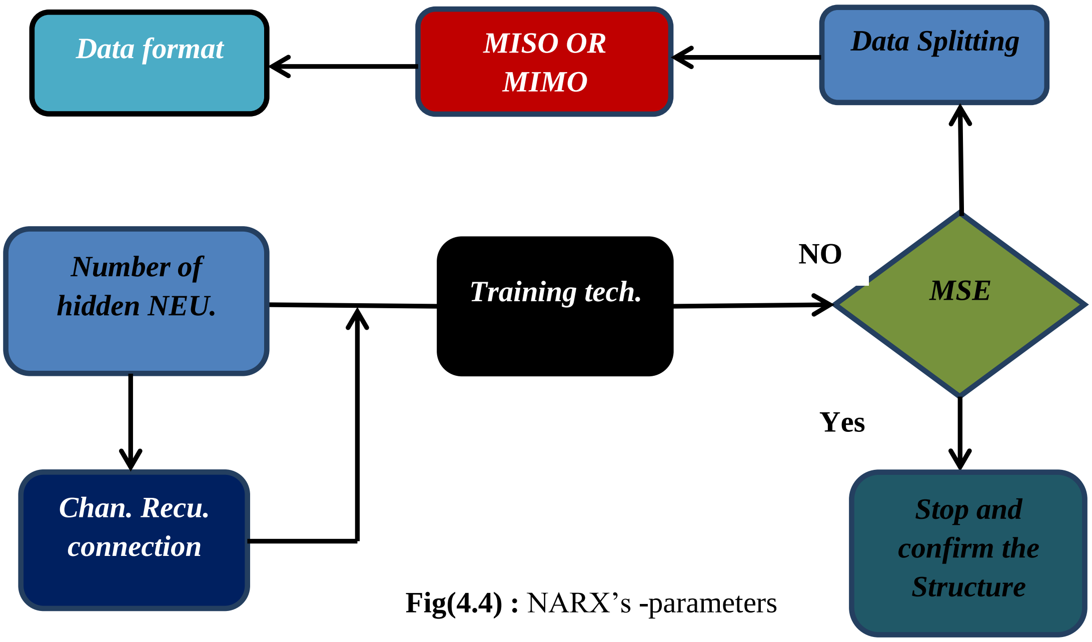
Moreover, step size can be estimated by a different method other than the line search method. This method is based on the combination of the CG approach with the model trust region method of the LM method. The resulted technique, also referred to as SCG, which was the first to be discussed in the literature by (Buduma and Locascio, 2017). In this approach and as it is explained in Eq. 4.13, where s represents the approximate Hessian matrix, E represents the overall error function, and E represents the gradient of E, scaling variables The user initializes σ_k and $\lambda_k$at the start of the algorithm so that they are $0<\lambda_k<10^{-6}$ and $0<\sigma_k<10^{-4}$to approximate the Hessian matrix. In Eq. 4.14 and Eq. 4.15, (Buduma and Locascio, 2017), the computation of both the β_k factor and the new search's direction are shown for SCG.

$$s_k = \frac{E^{`}(w_k + \sigma_k p_k) - E^{`}(w_k)}{\sigma_k} + \lambda_k p_k \quad (4.13)$$

$$\beta_k = \frac{(|g_{k+1}|^2 - {g_{k+1}}^T g_k)}{{g_k}^T g_k} \quad (4.14)$$

$$p_{k+1} = -g_{k+1} + \beta_k p_k \quad (4.15)$$

Iterative adjustments of the design factors separately are crucial to the algorithm's achievement. Compared to algorithms based on line searches, this is a significant benefit. Figure 4.4 summarizes the requirements for configuring the NARX model.



Fig(4.4) : NARX's -parameters

4.4 Performance Metrics

The output of the model for one output parameter is compared to the related actual values, or testing values, through the MSE, that indicates the gap between the expected and actual output, in addition to the determination of (R^2) for each of the three output variables, which assesses whether the model fits the data appropriately or not. This comparison is done to select the best model between all of the trained and developed NARX structures**.** The most effective model will be chosen from all the models under investigation based on its low MSE as well as high R^2accuracy .

4.4.1Mean Squared Error

The main goal of NNs training is to reduce errors as much as possible. Error reduction essentially entails improving training efficiency and getting a more precise model. While training a NN, different concepts and kinds of error may be considered. The gap between both the desired output (target) and the measured (actual) output, for instance, is called absolute error (Asgari, 2014).However, when training (NNs), MSE or RMSE are frequently used. As shown in the equation 4.21 and 4.22, respectively, MSE and RMSE are calculated (Asgari, 2014). Here, x_m is the measured (collected) data, x is the predicted (forecasted) output obtained from developed model, n_d is the size of data. By adjusting the weights and/or training techniques, errors can be minimized.

$$MSE = \frac{1}{n_d}\sum_{i=1}^{n_d}\left(\frac{x_{mi} - x_i}{x_{mi}}\right)^2 \tag{4.21}$$

$$RMSE = \sqrt{\frac{1}{n_d}\sum_{i=1}^{n_d}\left(\frac{x_{mi} - x_i}{x_{mi}}\right)^2} \tag{4.22}$$

In particular, when comparing the developed models, both the MSE and RMSE are mathematical measures to determine both reliability and accuracy of a given model in terms of the differences between the actual and predicted data values. MSE is used to find out how near a regression line is to a known line of points. The distances between both the known line and the regression line is called the error. If these values of errors are squared, the obtained value is called MSE (as given in Eq. 4.21). sometimes, negative error values may be obtained. These values should be squared to remove negative signs. As the MSE value approaches zero, then model has powerful predictive capabilities of the dependent variable. The smaller the MSE is, the better the model in predicting. Therefore, different models will be assessed and used based on their obtained MSE value.

4.4.2 The Regression Factor (R^2)

The regression coefficient or factor, also known as R^2, is a statistical criterion usually employed for assessing the accuracy of prediction depending on the variances values of the actual data and estimated ones. In addition, it is used to compare models and evaluate how close a given line fits a predicted data-set. Particularly, when contrasting models, regression provides the % of output differences which a model is able to consider and offers a measure of how well a line tracks the variations in a set of data. R^2(Asgari, 2014) is expressed mathematically in equation (4.23).

$$R^2 = 1 - \frac{ss_{rs}}{ss_{tss}} = 1 - \frac{\sum_i^n((x_i - \hat{x}_i)^2)}{\sum_i^n((x_i - \bar{x}_i)^2)} \tag{4.23}$$

In this equation, the x_i is the available data ,$\hat{x}_i$ is the model estimated value , n number of data, ss_{rs} represents the remaining sum of SE of the regression, while ss_{tss}denotes the sum of total of SE.R^2has a value in the range of 0 and 1.When ss_{rs} equal zero the regretion equal 1 That is if the modeled values perfectly match observed values. R^2in a base model, which consistently estimates$\bar{x}_i$. As the computed R^2 nears one, as the model's prediction capabilities will be better (Asgari, 2014).

4.5 Results, Discussion and Analysis

The dynamic behavior of a wind turbine has been captured by applying the proposed NARX models. Based on the obtained results, it is expected that the station efficiency, control, and general efficiency will be improved.

The NARX structure had been chosen because it can deal with noisy data from of the actual world, like the data from a wind turbine. It is also straightforward and simple to setup on various software. A number of hyper-parameters for NARX model has been adjusted for this reason in order to produce an ideal model which can be generalized and is a nominate to model such wind turbine systems with satisfactory results. Additionally, the (MIMO) architecture and the (MISO) architecture of the NARX model has been designed and evaluated. Building a MIMO architecture involves feeding the network with the predetermined 4 input factors and the one chosen output-output power from of the wind turbine, while simulating output parameters simultaneously. The MISO architecture, on the other hand, feeds every output variable to the network independently and simulates them all at once. When both architectures are evaluated, it will be possible to make a solid case for which one is more appropriate for certain uses based on computing, accuracy and complexity. In this study, collected data sets were split to 80% for training phase and 20% for testing phase, meaning that during the wind turbine operational hours, the network was tested for two hours and a half and the remaining data were used for the training phase. This section performed exceptionally well and produced good results, in part because the testing set was adequate in terms of the WT's operational hours for assessing the performance of the models.

In MATLAB programming, thorough computer code has been created to run up and operate the NARX model with complex generalization features.

4.5.1 Results of the Dynamic NARX NN Simulation Analysis

The written code in MATLAB programming indicates that the number's early ending condition was disabled and the total number of epochs have been set to 1000. If any of the mentioned below scenarios actually occur, the training process will stop directly.

i. The desired number of repeats, or epochs, has been reached.
ii. The time allotted has passed.

iii. There are no more enhancements because the performance has been reduced to the desired level.
iv. Performance gradient decreases to a value that is less than min grad.
v. The training process has been stopped when there was any overfitting using the written code on the validation set.

The practical WT data points were split into two groupings: the training set (80%) used to train the model while the other 20% for testing it as a new data for model generalization and accuracy assessment. Because the true output data seem to be accessible during in the training period, the split data points have been implemented to train an open-loop NARX structure to guarantee an efficient learning process.

Then, the best open-loop system will easily be transformed into a closed-loop structure for multiphase predicting after being identified as the optimum open-loop NARX system as a result of number of experiments. Several open-loop NARX structures using MIMO and parallel MISO have been tested in this research.

The following subsections discuss the MIMO and MISO NARX architectures which have been built to simulate a wind turbine, their construction experiments, and their MSE and R^2coefficient investigations.

4.5.1.1 Numerical Results for the Parallel MISO NARX Model

The MISO NARX wind turbine-based structure have been created including one hidden layer, a different tapped delay time values, a variety of hidden neurons and various data formats. The network had to have an output layer with one neuron. The three learning strategies: LM, Bayesian regularization, and SGC have all been implemented. The examples of attempts to build the MISO NARX design in terms of MSE performance and R^2 of the produced MISO NARX structures are extracted and shown in Table (4.1) .

Table 4.1: The output power P (MW) results for MISO NARX systems (MW). Bold is used to highlight the best.

Hidden layer neurons	**Time delay**	**Training algorithm**	**Data format**	**Performance MSE** Training	Validation	Test	$\boldsymbol{R^2}$ Training	Validation	Test
20	5	BR	Norm	5.0971E-05		5.4E-03	0.99963		0.87297
9	5	BR	Norm	4.5968E-04		1.6E-03	0.99665		0.9245
10	10	BR	Norm	2.5287E-04		2.1E-03	0.9981		0.91775
11	15	BR	Norm	1.2397E-04		1.6E-03	0.99908		0.91305
15	25	BR	Norm	2.1905E-06		2.2E-03	0.99998		0.93569
20	30	BR	Norm	1.9452E-13		1.5E-03	1		0.95937
5	2	BR	Norm	8.7808E-04		1.6E-03	0.99293		0.953
2	**5**	**BR**	**Norm**	**2.002E-03**		**2.4E-03**	**0.98435**		**0.97707**
2	**7**	**BR**	**Norm**	**1.874E-03**		**2.4E-03**	**0.9859**		**0.97199**
2	2	BR	Norm	1.6806E-03		2.6E-0-3	0.98722		0.96384
30	20	BR	Norm	1.5535E-12		1.9E-03	1		0.95461

2	**7**	**LM**	**Norm**	**2.2618E-03**	**3.3448E-03**	**2.9E-03**	**0.98474**	**0.98164**	**0.97688**
2	**5**	**LM**	**Norm**	**2.8551E-04**	**3.113E-03**	**2.7E-03**	**0.98148**	**0.97813**	**0.97646**
5	10	LM	Norm	1.4E-03	4.3362E-03	2.7E-03	0.98677	0.96343	0.97224
10	10	LM	Norm	9.08E-04	3.4247E-03	2.5E-03	0.98705	0.97659	0.97235
11	15	LM	Norm	4.43E-04	5.0999E-03	2.2E-03	0.99088	0.96419	0.97688
15	25	LM	Norm	2.51E-04	5.1525E-03	2.7E-03	0.98963	0.96177	0.96839
20	30	LM	Norm	3.05e-05	5.5777E-03	3.4E-03	0.98554	0.95016	0.95972
5	2	SCG	Norm	3.11E-03	9.6097E-03	4.5E-03	0.97618	0.93797	0.95703
9	5	SCG	Norm	5.01E-03	7.9147E-03	5.9E-03	0.95617	0.95188	0.95637
10	10	SCG	Norm	5.29E-03	9.9716E-03	6.9E-03	0.95703	0.92604	0.94689
11	**15**	**SCG**	**Norm**	**3.26E-03**	**4.3298E-03**	**3.9E-03**	**0.9743**	**0.96318**	**0.96551**
15	25	SCG	Norm	3.54E-03	4.2691E-03	4.3E-03	0.97056	0.96931	0.95989
20	30	SCG	Norm	2.97E-03	4.008E-03	4.6E-03	0.9753	0.97401	0.94047

Table 4.2: The output power P (MW) results for MISO NARX systems (MW). With Stand and Actual data type.

Hidden layer neurons	Time delay	Training algorithm	Data format	Performance MSE			R^2		
				Training	Validation	Test	Training	Validation	Test
5	2	SCG	Stand	3.11E-02	9.6097E-03	4.5E-02	0.97618	0.93797	0.95703
9	5	SCG	Stand	5.01E-02	7.9147E-03	5.9E-02	0.95617	0.95188	0.95637
10	10	SCG	Stand	5.29E-02	9.9716E-03	6.9E-02	0.95703	0.92604	0.94689
11	15	SCG	Stand	3.26E-02	4.3298E-03	3.9E-02	0.9743	0.96318	0.96551
15	25	SCG	Stand	3.54E-02	4.2691E-03	4.3E-02	0.97056	0.96931	0.95989
20	**30**	**SCG**	**Stand**	**2.97E-02**	**4.008E-03**	**4.6E-02**	**0.9753**	**0.97401**	**0.94047**
5	2	BR	Actual	1.2102E-01		6.7434E-06	1		1
9	5	BR	Actual	2.5613E-01		6.7436E-06	0.9524		0.9368
10	10	BR	Actual	1.4497E-01		2.5060E-07	1		1
11	15	BR	Actual	1.7642E-01		3.3149E-07	1		1
15	25	BR	Actual	1.4258E-01		1.4642E-07	1		1

This table shows the output power forecast readings with Actual and Stand data type but the error rate is high, so we used normalized data.

According to the table (5.1), It is noticeable that all MISO NARX structures that used the normalized data format performed well in terms of MSE performance in addition to regression. By averaging the test subset MSE performance for each WT's output parameter across all algorithm iterations, Table 4.3 shows the best outcomes for each training procedure.

Table 4.3: The average performance of the three MISO structures.

Training algorithm	Average performance MSE	R^2	Number of hidden layer neurons	Data type
LM	**2.2618E-03**	**0.98148**	**2**	**Norm**
BR	**1.874E-03**	**0.9859**	**2**	**Norm**
SCG	**3.26E-03**	**0.9743**	**11**	**Norm**

According to all tables, using a normalized data type, **Scaled Conjugate Gradient (SCG)** method, and a 15-sample time delay, the best MSE and R^2 of the 3 WT's factors were identified in the architecture of the 11 hidden layer neurons. As shown in Figure (4.5), the best training performance with an average MSE of **3.26E-03** was achieved after 1000 epochs. As the max number of epochs has been achieved, the same NARX network also included the best regression coefficient.

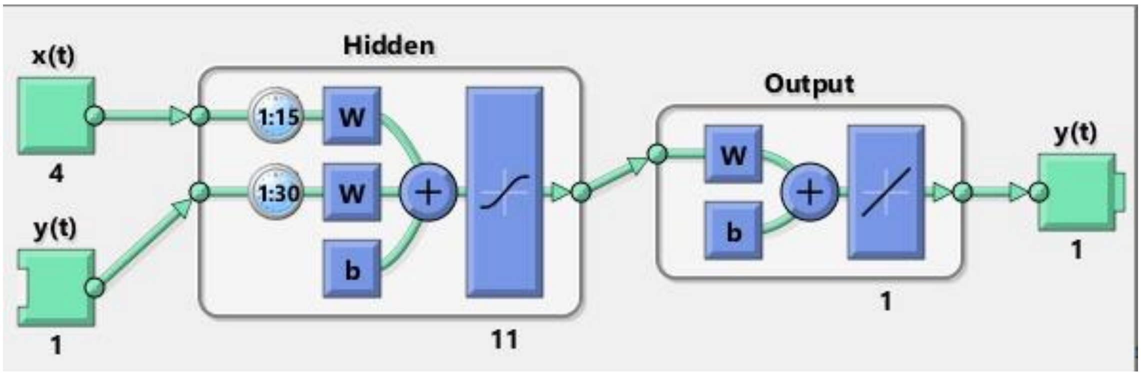


Figure 4.5: The optimal MIMO NARX system MATLAB schematic

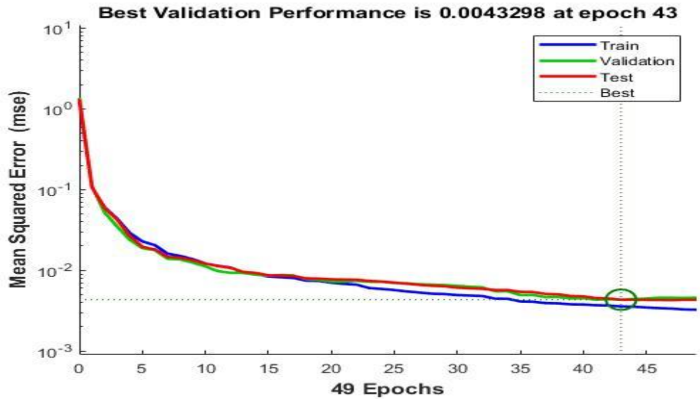
The three outputs (P) represented by y(t) can be seen connected with the input layer through a tapped delay line linked with adaptable weights in addition to the system's previously defined inputs, which are represented by x(t), indicating that the network has additional inputs that will be a function of weights optimization.

The ideal open loop MISO NARX model in Figure 5.1 has 11 neurons in the hidden layer. This figure shows that in addition to the four main system inputs, one output was also simultaneously fed into the output layer and 1 output was fed through into input layer.

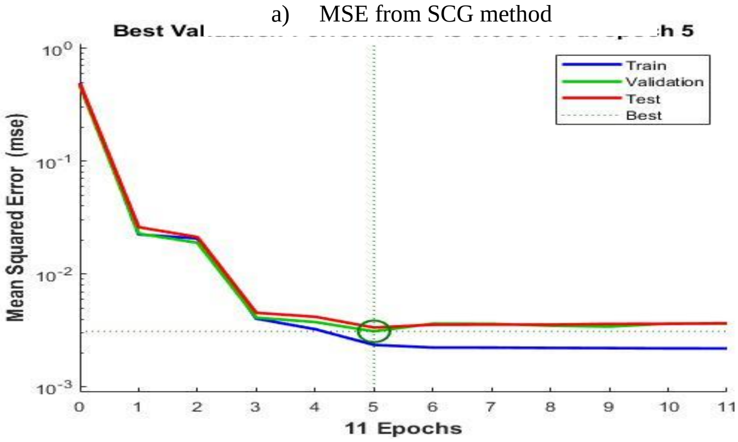
Although the MIMO NARX structures have relatively high regression coefficients and prediction performance, attempting to deal with single output at a time is more effective within the NARX system which may produce sufficient forecasting accuracy for each output parameter of the WT unit.

Figures (4.6 , 4.7, 4.8, 4.9) show the learning curve and regressions lines for every proposed MISO NARX model. These models were implemented using 4 inputs and single output. these figures show the test sets as well as their R^2 coefficient and the MSE trend of the training.

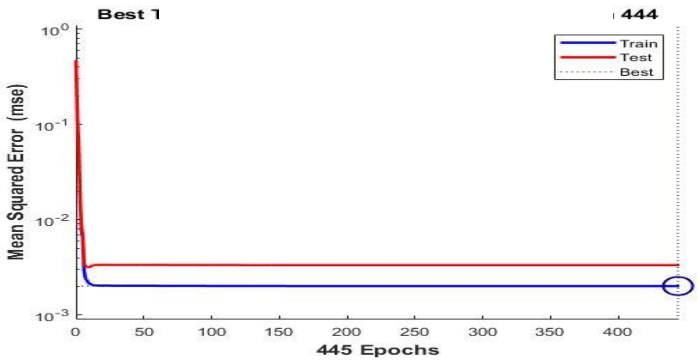
It is important to note that the SCG training algorithm's superiority in terms of computations can be attributed to the lack of an earlier stopping point and their optimization method, which is based on the previously discussed combination of weight values and the MSE performance function. Even though the SCG algorithm performed better than the BR and LM algorithms As shown in the figure (5.1), but the results from those methods were still very good and trustworthy. In addition to that, dealing with Normalized data in the ANN NARX system is much better than both actual and standardization data because of the harmonious relationship between the upper and lower limits of WT output in normalized values.



a) MSE from SCG method

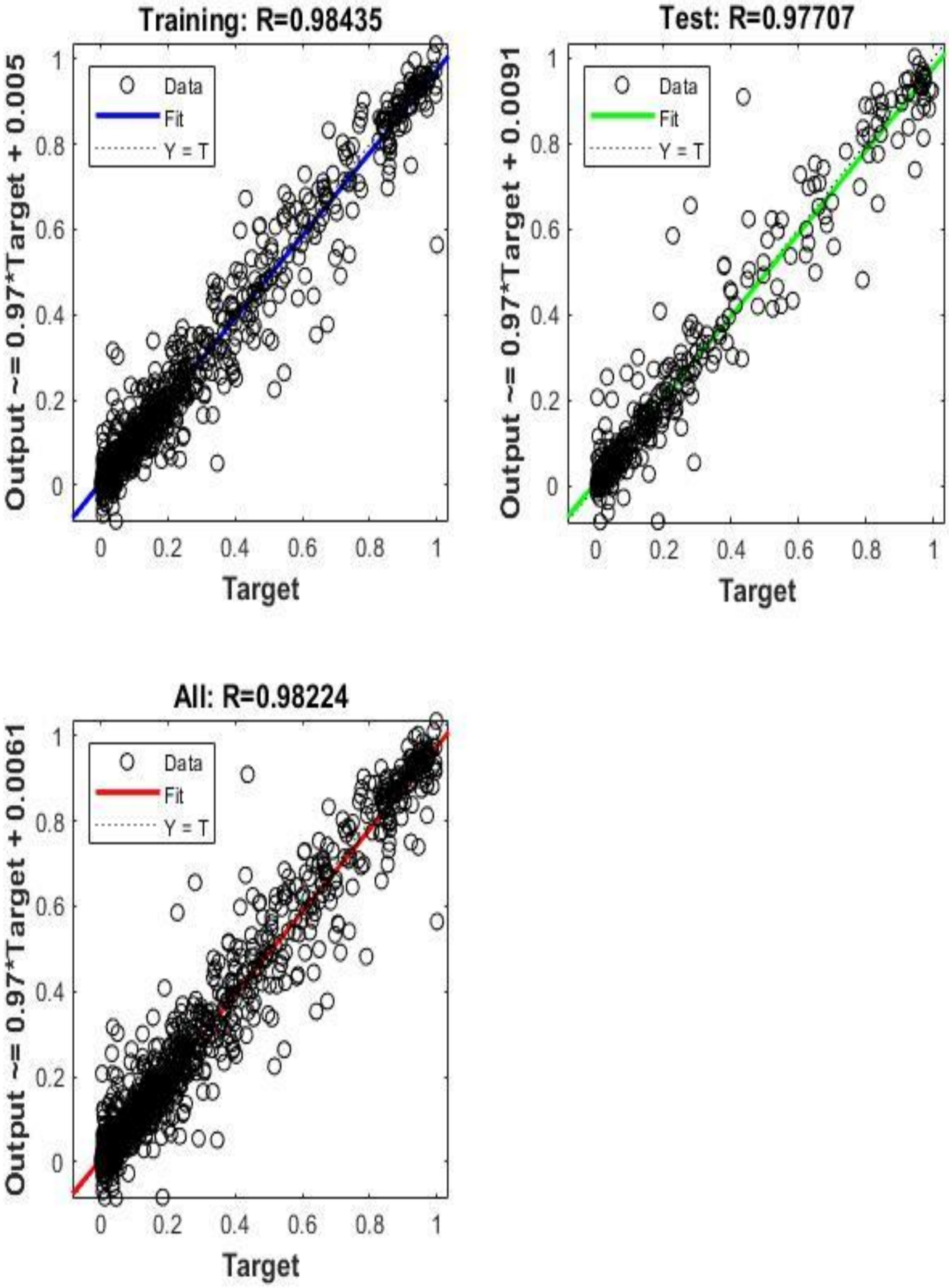


b) MSE from LM method

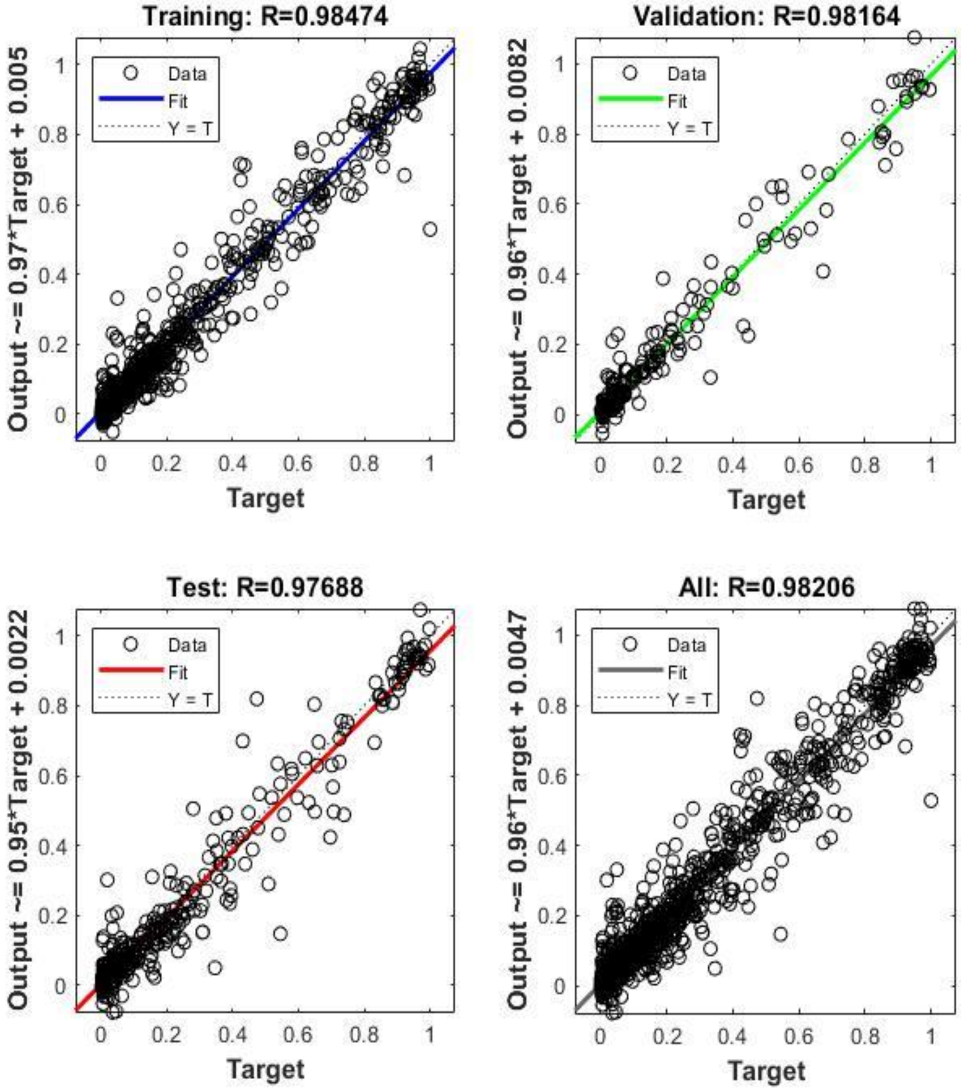


c) MSE from BR method

Figure4.6: Learning curve of the three MISO models

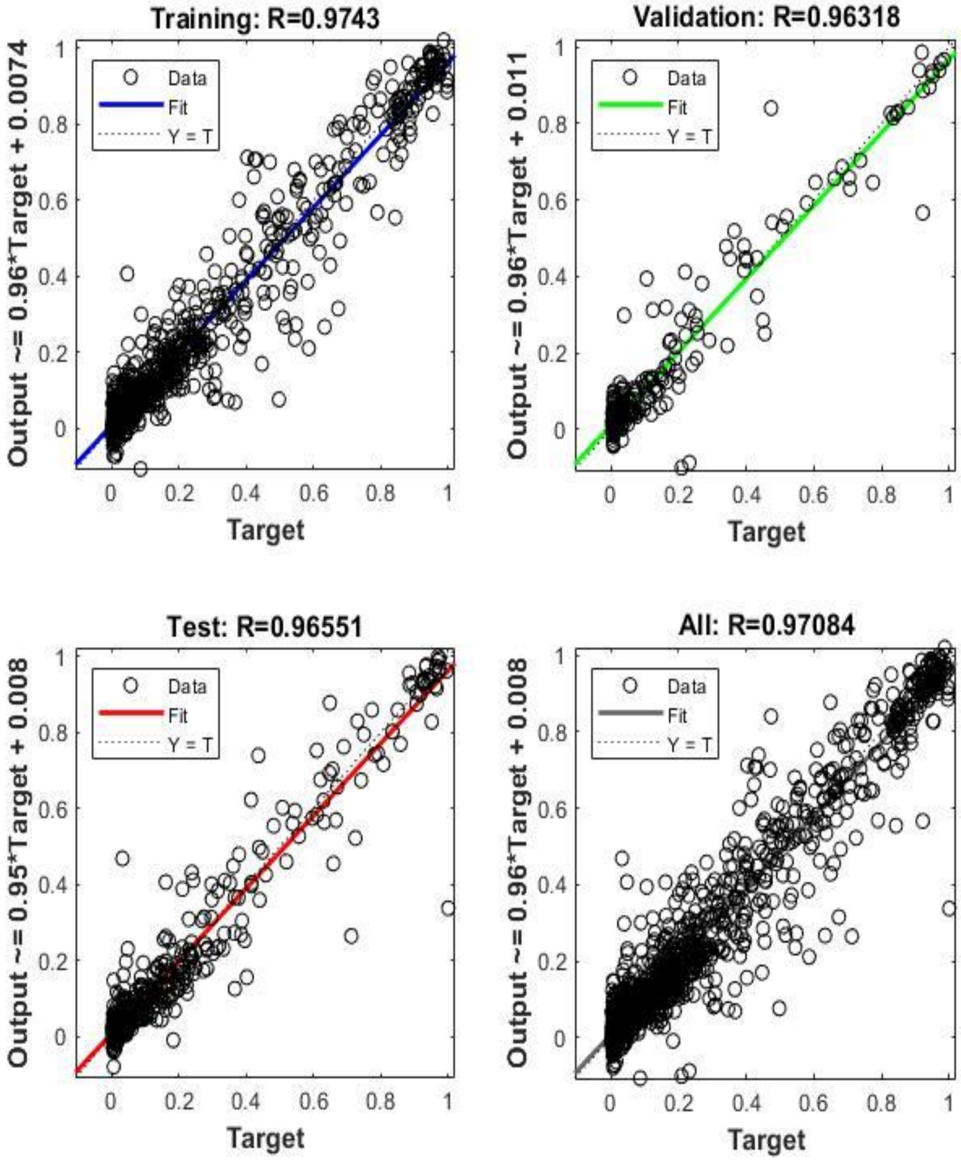


Fig(4.7): $R^2$from BR method



Fig(4.8):$R^2$from LM method

The declining MSE trend shows that the proposed MISO NARX model does not exhibit overfitting. Because the datasets are offset from the line where all of the outputs are equal to the targets, the R^2 lines show that the model produced the best fits.



Fig(4.9): R^2 from SCG method

4.5.2 MIMO and MISO Topologies

For NARX structures, MIMO results were less precise than MISO results, This is because of the way the parameters are handled; managing just one parameter will simplify the training process and/or weight optimization in relation to error. Accordingly, feeding a network by one parameter at a time results in smooth and precise network training. For instance, in the NARX structure, output parameter is feedback toward the input layer and it will be a function of adapting weights and biases.

4.5.3 Time-based Simulation Results and Discussion

Figures (4.10–4.11) show the structures and best simulation results for the NARX technique. Based on the outcomes and the related computed MSEs from of the prior analysis, It is clear that the dynamic NARX ANN have performed satisfactorily when applied to WTs. This means that these may be utilized to produce short- or long-term estimations, improving controllers, checking performance when a measurement device malfunctions, determining WT characteristics of various types, and more.

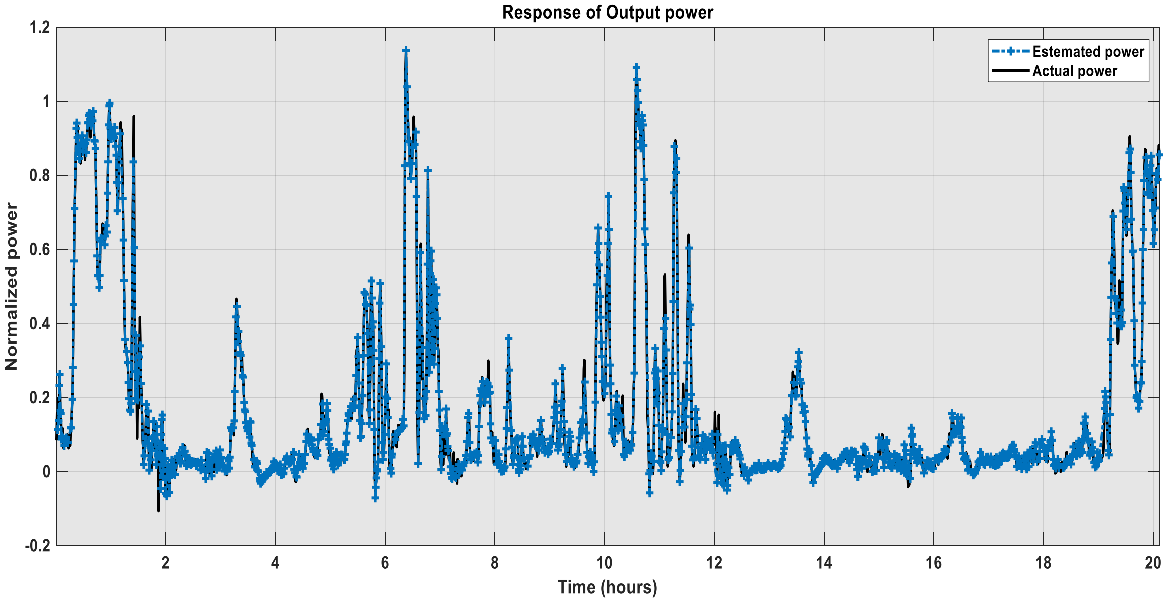


Figure 4.10: Norm power (MISO performance system)

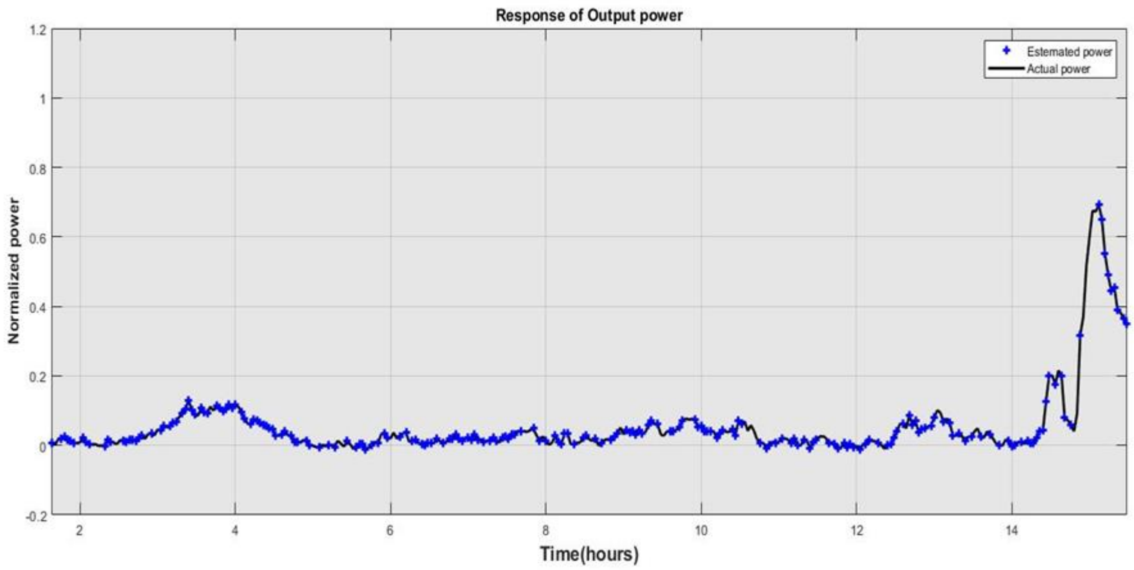


Figure 4.11: Zoomed power response

The NARX approach (ranges: 0-1 normalized and 2 MW actual power range for each WT) successfully tracks the trends, very small errors. For the accepted long operational hours of the WT, MSE have been achieved over a variety of trials.

Because wind turbine noise sources and unknowns are of large values and variations in realistic conditions, such accuracy rates in the responses of WTs may be challenging to achieve. The methods suggested in this study had already successfully operated for longer periods of time than those previously reported, covering more than 24 hours 0of operation.

Additionally, NARX model demonstrated marginal advantage in terms of absolute error and in increasing the outcomes of parallel MISO system; this may be due to followings:

1. Simplified design that shows a direct relationship between outputs and inputs, it produces more accurate responses of the inputs on the outputs.
2. As an example, sampling interval between the GT factor patterns in this study is 10 seconds, that indicates that the TDL in the NARX model can be set up using the same value to get the actual dynamic path of all these trends. The use of TDL in the NARX networks improves their ability to imitate for every parameter.
3. Using of feedback delayed output results as an another inputs to the system. This raises the inputs used and improves the accuracy of the output representation.
4. The current techniques, like SCG, BR, and LM which incorporate NARX to assess the ideal weight values, are marginally superior to other approaches because they depend on finding the best results in terms of MSE coefficient as well as gradient reasonable algorithm.

Generally, it can be concluded that the dynamic ANN is still a best available option for modeling and simulating WTs because of its low simulation error and high simulation performance of WT variation trends. Instead of using time-based simulations, the reader can refer to the references (Bai *et al.*, 2021)(Ragab *et al.*, 2020)(Wunsch, Liesch and Broda, 2021)(Wang and Chen, 2019) for more information on other successful applications of NARX ANN.

4.5.4 Conclusion

In this thesis, ANNs were employed to evaluate novel modeling and simulation methods for wind turbines. Dynamic NARX systems has delivered precise results using the most recent trends. These results support the scientific research in the area of dynamic ANNs in experimenting the performance of the WT station. The design and production of WTs that are more efficient, reliable, and durable may be facilitated by the new approaches that are presented. The models which are developed for this thesis could also be employed for wind turbine fault finding, monitoring, detector validation, optimization, and problem resolution.

Operational data sets were used to show how well the presented neural networks could capture the complex nonlinear dynamics of wind turbines, particularly in the absence of sufficient physics data. In conclusion, this thesis argued that, although some troublesome challenges associated with the application of ANNs for industrial uses, ANNs possess strong ability to be taken into consideration as a dependable approach compared to traditional modeling and simulation approaches.

The following improvements to wind turbine modeling and simulation are a result of this thesis:

1) This research provided a thorough analysis of the wind turbine modeling literature [Ch 2]. There were black-box models discussed. The most pertinent research projects for various types of WTs, in terms of methodology, benefits, and drawbacks, were studied and discussed. It was shown that despite outstanding research in the field, more study is still needed to address unexpected difficulties that crop up during manufacturing processes or the operation of industrial plants. These issues can arise during the design, commissioning, condition monitoring, fault detection, problem resolution, maintenance, detector validation, and control processes.

2) The structure of two cutting-edge ANNs techniques—dynamic (NARX) architectural style and deep (CNN)—was extensively discussed in this thesis. The research discusses all information related to the hyper parameters, such as the training algorithms' layers, and all significant mathematics, as well as their optimization approach. It [Ch 3] explained benefits of this approach and examined various difficulties that can occur while using ANN-based models for industrial systems.

3) It was demonstrated which the WT time-based dynamic performance can be accurately predicted using NARX ANN, with very minimal errors for NARX technique. The dynamic (NARX) methodology [Ch 4] and [Ch 5] can thus be used to model this particular class of wind turbine systems in a befitting manner.

4) When using ANNs techniques [ch5], it is typically actually advised to normalize the data of WTs other than working with actual and stand datasets.

5) the SCG training algorithm performs better than other training algorithms when setting up the NARX structure than the above-mentioned previous ones (BR and LM) [Ch 4] & [Ch 5].

4.5.5 Recommendations for Further Research

A very broad range of research tasks are covered by wind turbine simulation and modeling. There are many types of wind turbines, as well as numerous simulation and modeling systems and techniques. Numerous techniques and structures are taken into account, even when using an artificial neural network approach. A very broad range of research tasks are covered by gas turbine modeling and simulation. There are numerous types of gas turbines, as well as numerous simulation and modeling systems and methodologies. Numerous techniques and structures are taken into account, even when using an artificial neural network approach. However, the purpose of future initiatives and forthcoming research outputs in this line of work can be illustrated in relation to the focus and results of this thesis:

1) Despite the thesis's goals being met, there are still some deep learning approaches which have not yet been researched in the literature. These approaches may perform similarly, which prompts the mention of a few potential areas for future research.

2) Utilizing additional techniques for deep learning and suitably comparing them with advanced models is one of the most obvious future developments. Locally connected NNS and sophisticated deep RNN may be examples of this.

3) Another realistic future scenario involves creating a supervisory controller for the created ANN models and using it to control the premix and diffusion modes while aiming for greater efficiency. It may be helpful to conduct a comparison with other forecasting philosophies, like physics-based approach, and to High concentration on performance measures than just the accuracy's numerical value.

References

Abdmouleh, Z., Alammari, R. A. M. and Gastli, A. (2015) 'Review of policies encouraging renewable energy integration & best practices', *Renewable and Sustainable Energy Reviews*. Elsevier Ltd, pp. 249–262. doi: 10.1016/j.rser.2015.01.035.

Abiodun, O. I. *et al.* (2018) 'State-of-the-art in artificial neural network applications: A survey', *Heliyon*, 4(11), p. e00938. doi: 10.1016/j.heliyon.2018.e00938.

Ahmad, M. *et al.* (2022) *Modeling and Simulation of a Practical Dual-Fuel Gas Turbine Power Generation System using Dynamic Neural Network and Deep Learning Technique*.

Asgari, H. (2014) *Modelling, Simulation and Control of Gas Turbines Using Artificial Neural Networks*.

Awan, A. B. and Khan, Z. A. (2014) 'Recent progress in renewable energy - Remedy of energy crisis in Pakistan', *Renewable and Sustainable Energy Reviews*. Elsevier Ltd, pp. 236–253. doi: 10.1016/j.rser.2014.01.089.

Bai, M. *et al.* (2021) 'Convolutional neural network-based deep transfer learning for fault detection of gas turbine combustion chambers', *Applied Energy*, 302(August), p. 117509. doi: 10.1016/j.apenergy.2021.117509.

Bolouri Afshar, B. (2016) *Wind Power Forecasting Using Artificial Neural Networks with Numerical Prediction-A Case Study for Mountainous Canada*.

Buduma, N. and Locascio, N. (2017) *Fundamentals of deep learning: designing next-generation machine intelligence algorithms.*, *Designing next-generation machine intelligence algorithms*. Available at: http://perso.ens-lyon.fr/jacques.jayez/Cours/Implicite/Fundamentals_of_Deep_Learning.pdf.

Cadenas, E. and Rivera, W. (2010) 'Wind speed forecasting in three different regions of Mexico, using a hybrid ARIMA-ANN model', *Renewable Energy*, 35(12), pp. 2732–2738. doi: 10.1016/j.renene.2010.04.022.

Chinedu, E. (2019) *WIND FARM POWER OUTPUT PREDICTION BASED ON MACHINE LEARNING RECURRENT NEURAL NETWORKS*.

Choudhary, A. and Chauhan, V. (2014) 'Solar Energy Prediction using ANN', 21(1), pp. 1–9. doi: 10.9790/0661-2101010107.

Demirbaş, M. D. and Çakır, D. (2019) 'Thermal Stress Analysis in Two-Directional Functionally Graded Plates with Artificial Neural

Network Training Algorithms', *Uluslararası Muhendislik Arastirma ve Gelistirme Dergisi*, (June), pp. 442–450. doi: 10.29137/umagd.485604.
Edelen, A. L. *et al.* (2016) 'Neural Networks for Modeling and Control of Particle Accelerators', *IEEE Transactions on Nuclear Science*, 63(2), pp. 878–897. doi: 10.1109/TNS.2016.2543203.
Howard, D. and Mark, B. (2004) 'Neural Network Toolbox Documentation', *Neural Network Tool*, p. 846.
Iessa, A. *et al.* (2017) 'Method of estimating the maximum penetration level of wind power using transient frequency deviation index based on COI frequency', in *PECON 2016 - 2016 IEEE 6th International Conference on Power and Energy, Conference Proceeding*. Institute of Electrical and Electronics Engineers Inc., pp. 274–279. doi: 10.1109/PECON.2016.7951572.
Khan, T. *et al.* (2020) 'Agricultural Fruit Prediction Using Deep Neural Networks', *Procedia Computer Science*, 174(2019), pp. 72–78. doi: 10.1016/j.procs.2020.06.058.
Khan, Z. N. and Ahmad, J. (2021) 'Attention induced multi-head convolutional neural network for human activity recognition', *Applied Soft Computing*, 110, p. 107671. doi: 10.1016/j.asoc.2021.107671.
Kibet, N. and Nicholas, R. (no date) *Forecasting of wind speeds and directions with artificial neural networks*.
Lecun, Y., Bengio, Y. and Hinton, G. (2015) 'Deep learning', *Nature*, 521(7553), pp. 436–444. doi: 10.1038/nature14539.
Liu, Jiao *et al.* (2018) 'Fault detection for gas turbine hot components based on a convolutional neural network', *Energies*, 11(8). doi: 10.3390/en11082149.
Liu, Q. *et al.* (2020) 'An Optimal NARX Neural Network Identification Model for a Magnetorheological Damper With Force-Distortion Behavior', *Frontiers in Materials*, 7(February), pp. 1–12. doi: 10.3389/fmats.2020.00010.
Liu, Z. *et al.* (2012) *Wind Power Plant Prediction by Using Neural Networks: Preprint*. Available at: http://www.osti.gov/bridge.
Lu, N. *et al.* (2013) 'A comparison of forecast error generators for modeling wind and load uncertainty', *IEEE Power and Energy Society General Meeting*, pp. 0–4. doi: 10.1109/PESMG.2013.6672978.
MacKay, D. J. C. (1992) 'Bayesian Interpolation', *Neural Computation*, 4(3), pp. 415–447. doi: 10.1162/neco.1992.4.3.415.
Newton, P. W. and Rogers, B. C. (2020) 'Transforming built environments: Towards carbon neutral and blue-green cities', *Sustainability (Switzerland)*, 12(11). doi: 10.3390/su12114745.

Peiris, A. T., Jayasinghe, J. and Rathnayake, U. (2021) ‘Forecasting wind power generation using artificial neural network: “Pawan danawi”- A case study from Sri Lanka’, *Journal of Electrical and Computer Engineering*, 2021. doi: 10.1155/2021/5577547.

Qu, X. *et al.* (2016) ‘Short-term prediction of wind power based on deep Long Short-Term Memory’, in *Asia-Pacific Power and Energy Engineering Conference, APPEEC*. IEEE Computer Society, pp. 1148–1152. doi: 10.1109/APPEEC.2016.7779672.

Ragab, M. G. *et al.* (2020) ‘A novel one-dimensional cnn with exponential adaptive gradients for air pollution index prediction’, *Sustainability (Switzerland)*, 12(23), pp. 1–22. doi: 10.3390/su122310090.

Rahman, M. M. *et al.* (2021) ‘Prospective methodologies in hybrid renewable energy systems for energy prediction using artificial neural networks’, *Sustainability (Switzerland)*. MDPI AG, pp. 1–28. doi: 10.3390/su13042393.

Rashid, H., Haider, W. and Batunlu, C. (2020) ‘Forecasting of Wind Turbine Output Power Using Machine learning’, in *2020 10th International Conference on Advanced Computer Information Technologies, ACIT 2020 - Proceedings*. Institute of Electrical and Electronics Engineers Inc., pp. 396–399. doi: 10.1109/ACIT49673.2020.9208852.

Ren, Y. F. and Bao, G. Q. (2010) ‘Control strategy of maximum wind energy capture of direct-drive wind turbine generator based on neural-network’, in *Asia-Pacific Power and Energy Engineering Conference, APPEEC*. doi: 10.1109/APPEEC.2010.5448343.

Sun, T. *et al.* (2019) ‘Monthly Electricity Consumption Forecasting Method Based on X12 and STL Decomposition Model in an Integrated Energy System’, *Mathematical Problems in Engineering*, 2019. doi: 10.1155/2019/9012543.

Svensson, M. (no date) *Short-term wind power forecasting using artificial neural networks*.

Tu, J. V (1996) *Advantages and Disadvantages of Using Artificial Neural Networks versus Logistic Regression for Predicting Medical Outcomes*, *J CLIN EPIDEMIOL*.

Wang, S. and Chen, H. (2019) ‘A novel deep learning method for the classification of power quality disturbances using deep convolutional neural network’, *Applied Energy*, 235(March 2018), pp. 1126–1140. doi: 10.1016/j.apenergy.2018.09.160.

Wunsch, A., Liesch, T. and Broda, S. (2021) ‘Groundwater level forecasting with artificial neural networks: A comparison of long short-term memory (LSTM), convolutional neural networks (CNNs), and non-linear autoregressive networks with exogenous input (NARX)’, *Hydrology and Earth System Sciences*, 25(3), pp.

1671–1687. doi: 10.5194/hess-25-1671-2021.

Zhang, J. *et al*. (2019) 'Short-term forecasting and uncertainty analysis of wind turbine power based on long short-term memory network and Gaussian mixture model', *Applied Energy*, 241(January), pp. 229–244. doi: 10.1016/j.apenergy.2019.03.044.

المعلومات الشخصية

الاسم: أسامة نزار السبوع

التخصص: ماجستير هندسة الشبكات الذكية في أنظمة القوى الكهربائية

الكلية: الهندسة

سنة التخرج: 2023