UNIVERSIDAD PONTIFICIA COMILLAS

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

DELFT UNIVERSITY OF TECHNOLOGY

Erasmus Mundus Joint Master in Economics and Management of Network Industries (EMIN)

Master Thesis

SHORT TERM WIND POWER FORECASTING USING HYBRID MODELS

Author Sivaprasanth Masilamani

Supervisor Dr. Antonio Muñoz San Roque

Madrid, July 2017

Master's Thesis Presentation Authorization

THE STUDENT:

SIVAPRASANTH MASILAMANI

.......M. Sivaprasanth.......

THE SUPERVISOR

DR. ANTONIO MUÑOZ SAN ROQUE

Signed: .... Date: 3../ .7../ 2017

THE CO-SUPERVISOR

Signed: .... Date: ....../ ....../ ......

Authorization of the Master's Thesis Coordinator

Dr. Luis Olmos Camacho

Signed.: .... Date: 03../ 07../ 2017

UNIVERSIDAD PONTIFICIA COMILLAS

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

DELFT UNIVERSITY OF TECHNOLOGY

Erasmus Mundus Joint Master in Economics and Management of Network Industries (EMIN)

Master Thesis

SHORT TERM WIND POWER FORECASTING USING HYBRID MODELS

Author Sivaprasanth Masilamani

Supervisor Dr. Antonio Muñoz San Roque

Madrid, July 2017

This master thesis is part of the requirements of the Erasmus Mundus Joint Master in Economics and Management of Network Industries (EMIN), 2015-2017. Delft University of technology and Comillas Pontifical University are the two participating universities. In the end of this Erasmus Mundus EMIN Programme, student will be awarded both the Master degree of Electric Power Industry from Comillas Pontifical University, and the master degree of Engineering and Policy Analysis from Delft University of Technology.

ABSTRACT

This study improves the Short term wind power forecasting to help bid the wind power in the electricity market. Supplying power lesser/greater than the expected power creates imbalance in the Electricity system. Hence electricity markets impose penalty for supplying power lesser/greater than expected power. Bidding right amount of power is an important issue for the electricity power producers. This issue is very relevant for a wind power producer due to the inherent nature of wind. Wind is characterised by uncertainty and volatility. This study proposes hybrid approaches that use the meteorological forecast of wind power and statistical models to improve the accuracy of the wind power forecast over meteorological forecast. The statistical methods used in the study are linear regression model, Artificial Neural Network (ANN), Support Vector Machine (SVM) and K-Nearest Neighbour. The production data of ten wind farms in a portfolio, meteorological forecasts of the ten wind farms, total production of the portfolio and meteorological forecast of total production were collected for 532 days for every hour. These data were used to train and test the hybrid models. These hybrid models are then compared empirically with the meteorological forecasts. It is found that, for the data used for the study, hybrid model using artificial neural network performs the best but only slightly over the linear regression model. Followed by artificial neural network and linear regression model is support vector machine. Followed by support vector machine is K-Nearest Neighbour model. But all the hybrid models performed better than the meteorological forecast of wind power.

ACKNOWLEDGMENT

When I look back at the last four months, all I could find is the support and unconditional help rendered by several distinguished people to complete this thesis work. Above all, I am very proud at this point of time to be able to complete this thesis on time given the tight time constraint.

It is my genuine pleasure and gratefulness to express sincere thanks to my Professor and mentor Dr. Antonio Muñoz San Roque, Professor, Institute for Research in Technology, Universidad Pontificia Comillas, Escuela Técnica Superior de Ingeniería – ICAI, Madrid. His dedication and keen interest above all his willingness and interest to help his students not only made me to complete this work but also inspired me in great levels. His timely advice, careful scrutiny and scientific approach helped me to find a direction in this vast area of research.

Secondly, I would like to thank Professor Luis Olmos Camacho for coordinating and managing this joint master program "Economics and Management of Network Industry". It was his timely advice and guidance that helped me to spot my field of interest for research and approach the responsible person in the field, Dr. Antonio Muñoz San Roque, which turned out to be a period of intense learning and knowledge.

I am also thankful to all my friends and classmates in this master for creating a wonderful working and friendly environment to carry out this work

LIST OF TABLES

LIST OF FIGURES

Contents

1 Introduction

1.1 Background and Motivation

In recent years, both environmental benefit and economic advantage paved way for the rapid growth in the power generation by Renewable Energy Sources (RES) (Denny & O'Malley, 2006). As the world moves towards green and cleaner energy generation technologies, Wind energy has gained momentum in the recent years more than ever. Wind is developed by the uneven heating in the earth's atmosphere along with numerous other factors. Wind energy is a form of solar energy that is produced by harvesting the wind using wind turbines (Grogg, 2005). Wind energy is one of the most competitive renewable energy resources.

Global Wind Energy Council had estimated that more than 54 GW of new wind power plants had been installed globally in 2016 alone. The growth in the cumulative capacity was 12.6% and reached a total installed capacity of 486.8 GW (GWEC, 2017). GWEC, Global Wind Energy Council, also forecasted the future trends of wind power capacity based on the past experiences and data. By the end of 2021, GWEC estimated an addition of 357 GW of wind power capacity globally, of which Asian Markets alone is expected to add an additional 154 GW and Europe contributes around 73 GW (GWEC, 2017).

In spite of all these assuring numbers, one of the most important issues associated with the wind power is integration of the wind power to the existing grid. The biggest challenge for the integration of the wind power is the inherent nature of wind. The variability and uncertainty are the two important characteristics of wind energy that cause this difficulty (Hedayati-Mehdiabadi, Zhang, & Hedman, 2015). This is not only the case with wind power but also for other renewable sources as well. Dan Reicher, director of the Steyer-Taylor Center for Energy Policy and Finance at Stanford University, famously quoted "Sun shines in the day but power is used at night" (Aenlle, 2015).

In addition to technical complexities, the uncertainty complicates the process of economic dispatch (Hedayati-Mehdiabadi, Zhang, & Hedman, 2015). The uncertainty in the wind often causes the imbalance between the wind power expected and wind power generated. In order to match this imbalance, system operators are required to keep rapid reserves (Hedayati-Mehdiabadi, Zhang, & Hedman, 2015). The procurement of this instantaneously responsive reserved power capacity is not only costly but also variable in quantity depending on the randomness of wind power availability. This, as a result, hinders the smooth operation of the network and also deters the prospects of inexpensive prices applied to the final consumer (Giannitrapani, Paoletti, Vicino, & Zarrilli, 2014).

From the economic point of view of the wind farm owners, supplying a capacity lesser or greater than expected is expensive. In Spain, a penalty equivalent to the hourly market deviation costs is imposed on the producers if the deviation between the actual electricity generation and the forecast power is more than 5% (Fey, 2010).

Hence a solution to address all these issues is the need of the hour. One of the solutions to address these issues is forecasting of the wind power. This branch of study has gained rapid momentum as the wind energy grows. Current trends in wind power forecasting are concentrated in improving the accuracy of the forecasted wind power.

Forecasting of wind power shall be considered in different time scales. Soman et.al, in their paper, classified the time scales into four horizons. Very short-term forecasting forecasts wind power for a time range from few seconds to 30 minutes ahead. Short-term forecasting forecasts wind power for a time range from 30 minutes to 6 hours ahead. Medium-term forecasting forecasts wind power for a time range from 6 hours to 1 day ahead. Long-term forecasting forecasts wind power for a time range from 1 day to 1 week ahead (Soman, Zareipour, Malik, & Mandal, 2010).

The two main uses of short term forecasting of wind power are power system management and energy trading. The amount of conventional power plants production (Unit Commitment) and economic dispatch of the power production technologies need Short term wind forecasting. From the wind farm owner’s perspective, the short term wind power forecasting is very important to adjust their bids before the gate closure, thereby avoiding the penalty for the imbalance created by them in the system.

Due to these advantages, the accurate forecasting of wind power in short term has become one of the biggest talks of current times in the field of energy generation and plenty of researches have been initiated and continuing across the globe to increase the wind forecasting accuracy. There are many forecasting techniques that were proposed and discussed in popular literatures. Chapter 2 illustrates some of the state of the art techniques used for short term wind forecasting.

Although several models for short term wind forecasting have been developed and analysed in great depth, comparison between different models remains an area where deep research is needed. Also there is no standard error measure adopted to correctly interpret the performance of the model. There are several error measures available such as Mean Error (ME), Mean Absolute Error (MAE), Mean Square Error (MSE), Mean Absolute Percentage Error (MAPE) etc. (Costa, Crespo, Navarro, Lizcano, Madsen, & Feitosa, 2008). This simply complicates the standardization of measuring the accuracy of the wind forecasts.

As discussed above, short term wind power forecasting is from 30 minutes to 6 hours ahead. Forecasting in short term horizon needs high precision as it is used for the power grid to ensure power quality, to handle electricity market transactions and bidding of electricity on grid (Mao & Shaoshuai, 2016). This short term forecast can be based on Numerical Weather Prediction models (NWP).

Numerical weather prediction models are based on meteorological parameters such as wind speed, wind direction, humidity etc. These meteorological parameters are considered at different heights, at certain initial conditions and boundary conditions. Forecast of wind power based on NWP models takes into account contour and terrain of the location of the wind turbine to obtain an optimal estimation of wind speed. With the known wind speed at the known height of wind turbine, power curve of the wind turbine can be used to calculate the actual wind power output of the wind farm (Mao & Shaoshuai, 2016). There are several service providers that give meteorological forecasts of wind power. For example, in Spain, one of the service providers of meteorological forecast of wind power is Meteologica.

Portfolio management is a method to manage different entities of a portfolio so that company as a whole attains its corporate strategy. This is also applicable for wind portfolio management. Wind power producers aggregate their wind farms into one single portfolio in order to minimise the risk in the electricity market. The main advantage of making portfolio of wind farms is that the deviation created by some wind farms will be corrected by other wind farms in the portfolio. There by this helps to minimise the risk created by the volatile nature of wind.

1.2 Objective and Contribution

This study proposes a hybrid approach, which makes use of the meteorological forecasts and various statistical methods, to improve the short term wind power forecast. This study also makes a comparative empirical analysis of different hybrid approaches employed to improve the meteorological forecast of a wind power in short term horizon. Ultimately, the hybrid approach will be helpful to the wind farm owners with a portfolio of different wind farms to better manage the portfolio and also better devise his bidding strategy.

1.2.1 Main Research Question

The main research question that needs to be addressed to improve the meteorological forecast is

"What is the best forecasting method to improve the meteorological forecast of a portfolio of wind farms in the short term horizon?"

1.2.2 Sub research questions

In order to address the above main research question, following sub research questions are formed:

a. What is the performance of a Meteorological forecast in comparison with the Persistence method or naïve predictor?
b. What is the performance of a Meteorological forecast in comparison with a hybrid approach using linear regression model?
c. What is the performance of a Meteorological forecast in comparison with a hybrid approach using nonlinear regression models?
d. What is the deviation of the forecast of the production of individual wind farms in comparison with the forecast of a portfolio of wind farms?

1.2.3 Contribution

Penalties will be levied on the market agents who introduce imbalance in the Electrical system. For wind power producers, this problem is a major concern due to the volatile and uncertain nature of wind. The best way to address this issue is by developing or making use of accurate forecasting technique and bid the forecasted power in the market. One of the forecasted powers is the meteorological forecasted power from the meteorological forecast service provider. But there are ways to improve the accuracy of this meteorological forecasted power using advanced statistical techniques. This study aims to improve the meteorological forecasted power using hybrid models that make use of meteorological forecast of wind power and different statistical modelling techniques. The study also compares the performance of the different hybrid models with the meteorological forecast. Also for a wind power producer with several wind farms, the strategy to bid the power of individual wind farms against bidding of entire portfolio as a whole needs to be devised before bidding in the market. This study also compares the deviation, in the forecasted power output and the observed power output, of individual wind farms to the entire portfolio.

1.3 Research Scope and limitation

The first part of the study aims to improve the meteorological forecast of wind power. To achieve this, different hybrid models are used, that makes use of meteorological forecast of wind power and various statistical models. Statistical models used in this study are the regression models. In the regression models, both linear regression model and nonlinear regression models are used. In the nonlinear regression model, Artificial Neural Network (ANN), Support Vector Machine (SVM), and K-Nearest neighbours (KNN) techniques are used. There are numerous other advanced statistical techniques that are available to improve the forecast such as deep learning, neural fuzzy logic and so on. This study does not use those techniques for comparative empirical analysis.

1.4 Structure of the Report

The remainder of this thesis report is discussed as below: Chapter 2 describes a brief literature study that was conducted to identify the model used in the study. Chapter 3 describes briefly the problem setting for the further study. Chapter 4 gives a general description of the data collected. Chapter 5 describes Naïve predictor method or persistence method, linear regression model, Artificial Neural Network (ANN), Support Vector Machine (SVM), and K-Nearest neighbours (KNN) technique that are chosen for the study. Chapter 6 describes the empirical analysis of each models. Chapter 7 gives the conclusion from the above analysis.

2 Literature Review

Wind power is different from the conventional power generation units because of its uncertain and seasonal nature. This nature of the wind power affects not only the system operator to balance the Power balance equation but also the wind farm owners to bid the right amount of the power in the market. One way of solving this problem is by forecasting the wind power. The most important criterions of these forecasting are the accuracy and time horizons. Over the years, numerous researches have been conducted on this area and numerous methods have been proposed.

Soman et al, in their paper "A Review of Wind Power and Wind Speed Forecasting Methods with Different Time Horizons", classified the entire wind power prediction models based on physical approach, statistical approach, new approaches and hybrid approaches. The paper also stressed the usefulness of naïve predictor in the short term horizon and importance in using naïve predictor as benchmark model (Soman, Zareipour, Malik, & Mandal, 2010).

The physical approach, which is based on NWP models, uses the physical conditions such as surface temperature, pressure, humidity, obstacle etc. Many times the output of this model is used as the input to the other statistical models. (Lei, Shiyan, Chuanwen, Yan, & Hongling, 2009).

Yang Mao and Wang Shaoshuai, in their paper "A review of wind power forecasting & prediction", also classified the forecasting of wind power based on various parameters such as forecasting models, time scale, accuracy of the model, input parameters etc. Under the classification based on the forecasting models, the paper discusses the wide usage of statistical models. The most commonly used statistical models for the wind power prediction are based on the Artificial Intelligence. The most commonly used techniques under the Artificial Intelligence are Artificial Neural Network, Support Vector Machine, Wavelet Analysis and Fuzzy Logic (Mao & Shaoshuai, 2016).

Catalao et al, in their paper "An Artificial Neural Network Approach for Short-Term Wind Power Forecasting in Portugal", proposed an artificial Neural Network Approach for short term forecasting and the model was compared with Naïve Predictor based on Average MAPE. The paper used only the historical data as input without the use of any meteorological forecasted data. The proposed artificial Neural Network Method was found better than Naïve predictor method (Catalao, Pousinho, & Mendes, 2009).

Gevrey et al, in their paper “Review and comparison of methods to study the contribution of variables in artificial neural network models”, reviewed and compared most commonly studied and used methods to find the relative importance of input variables on output variables of Artificial Neural Network models. The methods studied in the study are Partial Derivate method, Profile method, Perturb method, Weights method, improved stepwise method A, Improved stepwise method B, Classical stepwise method. The study concluded that Partial Derivative method gives the best result (Gevrey, Dimopoulos, & Lek, 2003).

Yang et al, in their paper “Support-Vector-Machine-Enhanced Markov Model for Short-Term Wind Power Forecast”, developed a support vector machine (SVM)-enhanced Markov model to forecast short term wind power. With the aid of the historical data of the wind turbine power output, the model performed better in terms of accuracy that the naïve predictor model (Yang, He, Zhang, & Vittal, 2015).

Zhou et al, in their paper “Fine tuning support vector machines for short-term wind speed forecasting”, studied the fine tuning of LS-SVM model parameters. The paper applied three types of kernels such as linear, polynomial and Gaussian kernel to forecast the wind speed for one step ahead. The study compared the LS- SVM technique with the naïve predictor and found that LS-SVM betters naïve predictor most of time (Zhou, Shi, & Li, 2011).

In recent years, hybrid models are gaining momentum due to its ability to compensate the weakness of one approach using the strength of another approach. Some of the commonly used hybrid approaches include combination of Numerical Weather Prediction and Neural Network, combination of spatial correlation and Neural Network, and combination of Numerical weather prediction and time series (Soman, Zareipour, Malik, & Mandal, 2010).

Sideratos et al, in their paper “An Advanced Statistical Method for Wind Power Forecasting”, proposed an advanced model combining neural network and fuzzy logic technique. The inputs used in the model are past historical power data and meteorological forecast values of wind speed and direction. The method is suitable for short term prediction of wind power up to 48 hours (Sideratos, 2007).

K-Nearest Neighbour algorithm is not a commonly used method for Wind Power prediction. Al-shehri et al, in their paper “Student performance prediction, using Support Vector Machine and K-Nearest Neighbour” used Support Vector Machine and K-Nearest Neighbour method to predict the performance of students and compared the prediction result against correlation coefficient as comparison parameter. The paper concluded Support Vector Machine has a very slight edge over K-Nearest Neighbour (Al-Shehri, et al., 2017).

The literature review was conducted to study some of the state of the art approaches used for the short term wind power prediction. The statistical methods were widely used in short term wind power forecasting. Support Vector Machine and Neural networks were some of the commonly used statistical methods. Wind power prediction using hybrid approaches have gained significant momentum in the recent years. Some important hybrid approaches made use of Numerical Weather Prediction models and statistical methods such as neural network and time series model. But other statistical approaches such as support vector machine, linear regression model and K-nearest neighbour technique haven't been used in the hybrid approaches. It may therefore be advantageous to develop hybrid approaches using Numerical Weather Prediction data and support vector machine, linear regression model and K-Nearest neighbour to wind power forecasting. Also the comparative empirical analysis of hybrid approaches is another area where not many researches have been carried out. Hence a comparative empirical analysis of hybrid models would be beneficial to evaluate the performance of each model.

3 Problem Setting

In a day-ahead market, wind power producers have to bid their power more than 24 hours ahead. In this case, NWP models can be used to forecast the wind power. But predicting wind power for more than 24 hours ahead is often difficult and less accurate. Intraday market allows continuous trading of electricity power till the gate closure. In Spain, the intraday market is organised as six different sessions with intraday lead time (time between the gate closure and the real time) varying between 3.25h to 6.25h (Chaves-Ávila & Fernandes, 2015). In the European Power Exchange (EPEX), the intraday market takes place until 45 minutes before the real time in European Power Exchange (EPEXSPOT, 2013). Hence EPEX allows market agents to participate in the intraday market till 45 minutes before real time.

Most electricity markets impose penalties for creating imbalance in the electricity system. Hence the profit of the electric power producers is given by the following equation

$$Profit = Revenue - Cost - Penalty \qquad \text{Eq. (1)}$$

For a conventional energy producer, this problem is relatively minor due to its predictable nature and ability to adapt the generation during the time of mismatch between what is produced and what is expected. But for renewable energy producers, this problem is one of the most relevant problems due to unpredictable nature of the renewable energy sources. For wind power producers, this problem is of great relevance as it directly affects the profit.

Due to the heavy penalty for creating imbalance in the electricity system, bidding the right amount of power becomes a paramount importance for the wind power producers. Short term wind power forecasting is a better solution for this problem. The important parameter that needs to be considered in short term forecast of wind power is the accuracy. Wind power producers can obtain the meteorological forecast of wind power from different commercial providers and bid in the market. Relying only on the meteorological forecast of wind power is not always adequate. Hence the short term forecast of wind power needs to be improved over meteorological forecast. This study aims to improve the Short term wind power forecasting using a hybrid approach of meteorological forecast of wind power and statistical models to forecast the wind power for a time horizon of one hour.

4 Data Collection and Data treatment

Data can make or break the performance of a model to forecast the future values effectively. Equally important is how the input variables enter the model (Kuhn & Johnson, 2013). One of the important problems of the using the statistical models to forecast the wind power is the availability of relevant data.

In this study, data is collected from a portfolio of wind farms with an aggregated generation capacity of 500 MW. From the portfolio of wind farms, production data of specific ten wind farms for every hour for 523 days were collected. Hence the total number of data points in the data set are 24*523 (=12552). In addition to production data, the meteorological forecast of wind powers of ten wind farms was obtained. Along with individual wind farms, the total wind power production of the portfolio and total meteorological forecast of the portfolio were collected.

As some of the statistical model depends on the scale of the data points in the data set, the entire data set is normalised to a scale from 0 to 1. The entire data set is now divided into two sets as

1. Training Set or In sample data (75% of total data)
2. Test Set or Out-of-sample data (25% of total data)

The first part, training set, is used to train the models. The second part of the data set, also called as Test set or Out of Sample set, is used to measure the generalisation capability of trained models.

5 Methodology

The following sections briefly explain the models used in this study for the short term forecasting of wind power.

5.1 Naïve Predictor

Naïve predictor or Persistence Method is one of the commonly used benchmark models for comparing the performance of the forecasting models. This is because of its predicting capability and better performance over many other models in short term horizon (Foley, Leahy, Marvuglia, & McKeogh, 2012). Naïve Predictor, in simple terms, is whatever is the wind power now is going to be wind power in the next time step. Mathematically,

$$P(t+k) = P(t) \qquad \text{Eq. (2)}$$

Where P(t+k) – Wind power at time t+k, observed value of next time step

P(t) – Wind Power at time t, the observed value of current time step

Due to its better performance than many of the physical and statistical methods for very short term to short term forecasts, the performance of forecast models are compared and tested against this predictor to check how much accuracy has improved compared to the benchmark method (Milligan, Schwartz, & Wan, 2003). This study performs persistence method on the observed data as the first step in order to test whether the meteorological forecast of wind power is better than the persistence method.

5.2 Regression Models

Regression is a statistical technique to establish a relationship between two or more variables (Campbell & Campbell, 2008). In general, it is used to establish a relationship between independent variables and dependent variables. Independent variables are commonly called as Input variables and dependent variables are commonly called as Output variables. The relationship can be linear or non-linear. The broad family of regression models are classified according to the relationship they establish as below.

1. Linear Regression models
2. Nonlinear Regression models

The linear regression models, which establish linear relationship between inputs and outputs, can be adapted to nonlinear trends in the data by manually adding nonlinear terms such as square, cube, sine or other nonlinear terms. The exact nonlinear relationship between the inputs and outputs are often unknown. Some of the nonlinear regression models do not need to know the exact nonlinear relationship (Kuhn & Johnson, 2013). Some of the commonly used nonlinear regression models are as follows

2.1. Artificial Neural Network
2.2. Support Vector Machine
2.3. K- Nearest Neighbour model

In the following sections, a brief introduction about linear and various nonlinear regression models that will be used in the study is given.

5.2.1 Linear Regression methods

To identify a model, it is not just enough to recognize whether a statistical significance exists between the input variables and the output variable. In order to make a proper estimation of wind power, a quantitative model should be formed (Heijnen, 2016). The Linear regression model assumes that a linear relationship exists between input and output variables. Assuming a simple system with one input variable and one output variable, mathematically, a linear relationship is given by

$$y = \beta_0 + \beta_1 x + \varepsilon \qquad \text{Eq. (3)}$$

Where y is the dependent variable

x is the independent variable

β_0 and β_1 – coefficients of regression or unknown population parameters, constant and slope of the linear relationship respectively

ε – error term or residuals of the linear model where error term is assumed to have zero mean, constant variance, normally distributed and independent of each other.

For a system with i input variables and one output variable, the linear regression model can be given mathematically by

$$y = \beta_0 + \sum_{i=1}^{i} \beta_i * x_i + \varepsilon \qquad \text{Eq. (4)}$$

Where y is the output variable

x_i are the input variables

β_0 and β_i – coefficients of regression or unknown population parameters, constant and slope of the linear relationship respectively

ε – error term or residuals of the linear model where error term is assumed to have zero mean, constant variance, normally distributed and independent of each other.

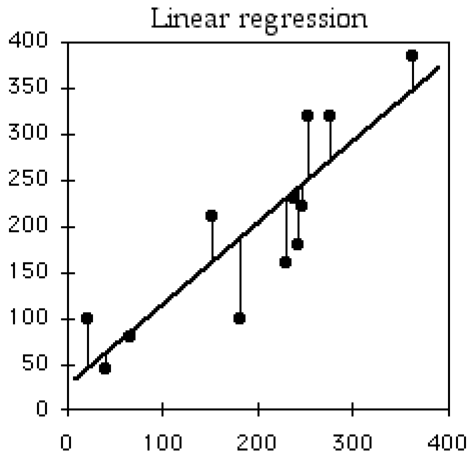

Figure 1 Linear Regression Line and Data points

Figure 1 (McDonald, 2014) shows the data points and regression line (indicated by think line) in a scatter plot. Linear regression line is the line that minimises the Sum of the Squared Errors (SSE) between the observed value and the predicted value. Sum of the Squared Error is given by

$$SSE = \sum_{i=1}^{n} (p_i - \hat{p_i})^2 \qquad \text{Eq. (5)}$$

Where p_i is the observed value and $\hat{p_i}$ is the predicted value.

Based on how the input variables are included/excluded in/from the linear model, three methods are classified as follows

1. Forward addition
2. Backward Elimination
3. Stepwise Estimation

Forward addition

This method fits the linear model in a step by step manner but only in one direction, namely addition. In every step one variable is added until a best model is found and no variable is removed from the model (Heijnen, 2016). The method is explained as follows using a system of 5 input variables, x_i, where i=1, 2,3,4,5 and output variable y.

Step 1: Firstly, the input variable that has the largest correlation with the output variable is determined. Using this variable a linear model is first established. Assuming x_2 being the input variable with the largest correlation, the initial regression model is found as follows

$$y = \beta_0 + \beta_2 * x_2 + \varepsilon \qquad \text{Eq. (6)}$$

Step 2: The input variable that has the second largest correlation with the output variable is determined and using this variable and already existing variables, another linear model is established. Assuming x_3 being the variable with the second largest correlation with y, the regression model obtained now is as follows

$$y = \beta_0 + \beta_2 * x_2 + \beta_3 * x_3 + \varepsilon \qquad \text{Eq. (7)}$$

Step 3: Now step 1 and step 2 is repeated till all the input variables that has correlation with the output variable is added.

Backward Elimination

This method fits the linear model in step by step manner but only in one direction, namely elimination. In every step one variable is removed until a best model is found and no variable is added to the model (Heijnen, 2016). The method is explained as follows using the same system used for explaining forward addition method (Heijnen, 2016)

Step 1: Using all the input variables, a linear regression model is determined. This is given as follows

$$y = \beta_0 + \sum_{i=1}^{5} \beta_i * x_i + \varepsilon \qquad \text{Eq. (8)}$$

Step 2: Now the input variable that has no significance to the output variable is identified. That input variable is removed from the model and a new model using the remaining input variables is determined. For example, in the above linear regression model, if the variable with the least significance is x_5, then the new linear regression model will be given as follows

$$y = \beta_0 + \sum_{i=1}^{4} \beta_i * x_i + \varepsilon \qquad \text{Eq. (9)}$$

Step 3: Step 2 is repeated till all the non-significant input variables are removed from the linear model.

Stepwise Estimation

Stepwise estimation method fits the linear model in step by step manner but in both directions, namely addition and elimination. In every step, one new variable is added or one variable is removed from the model until a best model is found (Heijnen, 2016). The step by step procedure to perform the step wise elimination method is as follows (Heijnen, 2016). Assuming the same system with 5 input variables, x_i, where i=1, 2,3,4,5 and output variable y,

Step 1: A linear regression model between output variable and the input variable, which has the highest correlation with the output variable, is created. Assuming x_2 being the input variable with the largest correlation, the initial regression model is found as follows

$$y = \beta_0 + \beta_2 * x_2 + \varepsilon \qquad \text{Eq. (10)}$$

Step 2: Another input variable that explains the output variable second best is determined. A multiple linear regression model among new input variable, existing input variables and output variable, is created. Assuming the second best input variable that explains the output variable is x_3

$$y = \beta_0 + \beta_2 * x_2 + \beta_3 * x_3 + \varepsilon \qquad \text{Eq. (11)}$$

Step 3: Now each input variable is removed one by one to check whether the removed input variable is significant to the output variable or not. For example, in the above model first, x_3 is removed and significance of x_3 on determining y is measured.

$$y = \beta_0 + \beta_2 * x_2 + \varepsilon \qquad \text{Eq. (12)}$$

Again the variable x_2 is removed from the current model and the significance of x_2 on y is determined.

$$y = \beta_0 + \varepsilon \qquad \text{Eq. (13)}$$

Step 4: In each step, if the significance of the input variable removed is nil, then this input variable is permanently removed from the model.

Step 5: Now step 2, step 3 and step 4 are repeated till all the significant input variables are included and non-significant input variables are removed and a final multiple linear regression model is determined. In this study, stepwise estimation is used to determine the forecasting model.

5.2.1.1 Assumption of Linear Regression model

In linear regression models, there are two main assumptions. They are as follows

1. Linear relationship exists between input and output variables
2. Error terms have zero mean, constant variance, normally distributed and independent of each other.

In a linear regression model, a linear relation is established between input and output variables. Hence the first and basic assumption is the linearity between input and output variables. One way to test this assumption is to examine the error term or residual plot. Residual plot shows residuals on the Y axis and predicted values on the X axis. If the residuals spread around the zero line in the residual plot, then the assumption of linearity is satisfied. Such a plot is called null plot (Heijnen, 2016). Figure 2 (Harvey, 2016) shows an example of a null plot.

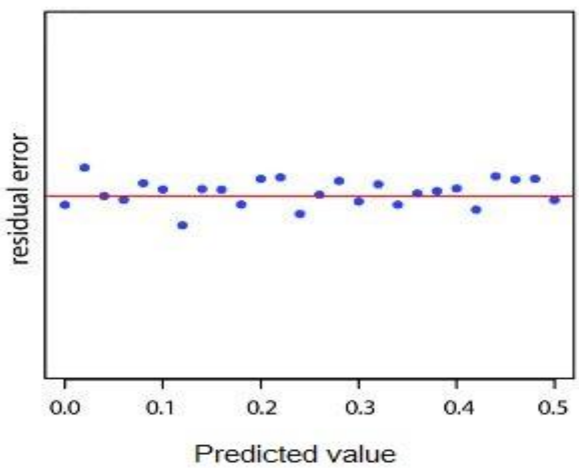

Figure 2 Residual Plot

Another important assumption is error terms are normally distributed, independent of each other (non-auto correlated) and have zero mean and constant variance. The normal distribution of error terms can also be tested by inspecting the histogram of residuals. The normal distribution of the error terms can also be tested by inspecting the P-P Plot or normal probability plot of the regression model. If the residuals are aligned along the normality line, then normal distribution of errors is confirmed (Heijnen, 2016). Figure 3 shows an example of P-P Plot with residuals aligned along the normality line

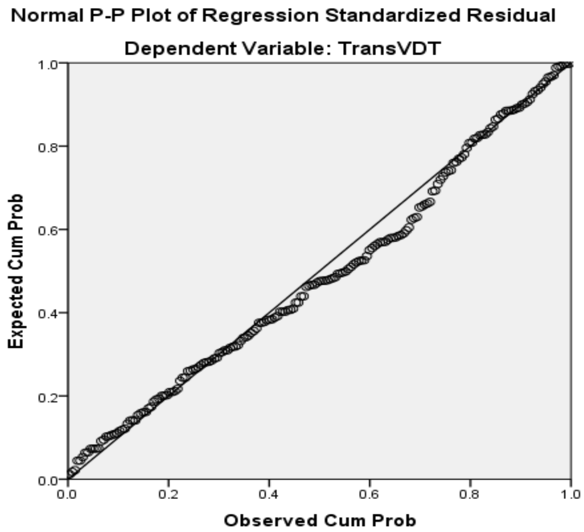

Figure 3 P-P Plot (Normal Probability Plot)

The independence of the error term can be tested by Durbin-Watson procedure (Heijnen, 2016). Mathematically, Durbin Watson procedure is represented as follows

$$d = \frac{\sum_{t=2}^{n}(e_t - e_{t-1})^2}{\sum_{t=1}^{n} {e_t}^2} \qquad \text{Eq. (14)}$$

Where e_t is the error term at time t and e_{t-1} is the error term of the previous time step t-1. The value of ‘d’ lies between 0 to 4. If the d value is near 2, then there is no serial auto correlation.

The phenomenon of constant variance of error terms is called Homoscedasticity. If the error terms do not have constant variance, then it is called Heteroscedasticity. Heteroscedasticity can be found by observing the plot between standardised residuals and standardised predicted values. If Heteroscedasticity exists, then the residuals in the standardised residual plot take a diamond or triangular shape (Heijnen, 2016). Figure 4 shows heterscedastic pattern

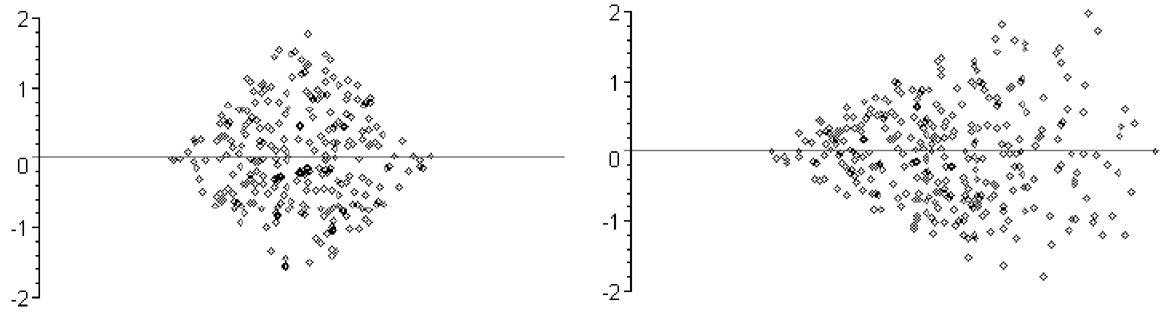

Figure 4 Residual plots in case of Heteroscedasticity

5.2.2 Artificial neural network

The following section introduces artificial neural network and sensitivity analysis that are used in this study.

5.2.2.1 Artificial Neural Network

Neural Networks were inspired from how brain learns its reactions (Bishop, 2006). To understand the architecture of a Neural Network, it is wise to start with the architecture of human brain. The brain consists of cells called neurons. Neurons are the building block of the nervous system. There are over 100 billion neurons in the nervous system and each connected through pathways that transmit the signals with each other. This idea is the basic foundation of the artificial neural network. Just like brain needs a large number of neurons to transmit information and learn reactions, with enough number of neurons in the hidden layers, the Artificial Neural network can be trained to learn any complex function. A typical architecture of Neural Network is shown in Figure 5 (Kuhn & Johnson, 2013).

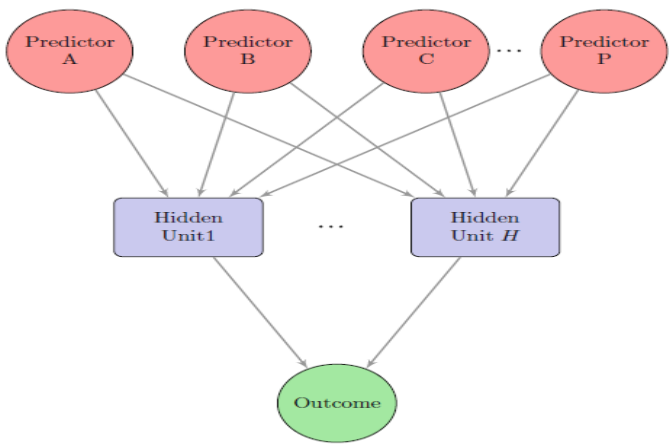

Figure 5 Architecture of ANN

Figure 5 (Kuhn & Johnson, 2013) shows a neural network with a single hidden layer. Multilayer perceptron (MLP) is one of the commonly used neural networks. A multilayer perceptron consists of an input layer, one or several hidden layers and one output layer. Each layer consists of a given number of neurons. The number of neurons in the input layer equals the number of input variables. The number of neurons in the hidden layers needs to be determined in the identification phase of the model. The number of neurons in the output layer equals the number of output variables (Wang, 2003). Every neuron in each layer is connected to every neuron in the succeeding layer through connecting lines. Each connecting lines are associated with weights. Similarly each neuron is associated with a bias. Figure 6 (Cazala, 2017) shows the multilayer perceptron with four neurons in input layer, five neurons in the hidden layer and one neuron in the output layer.

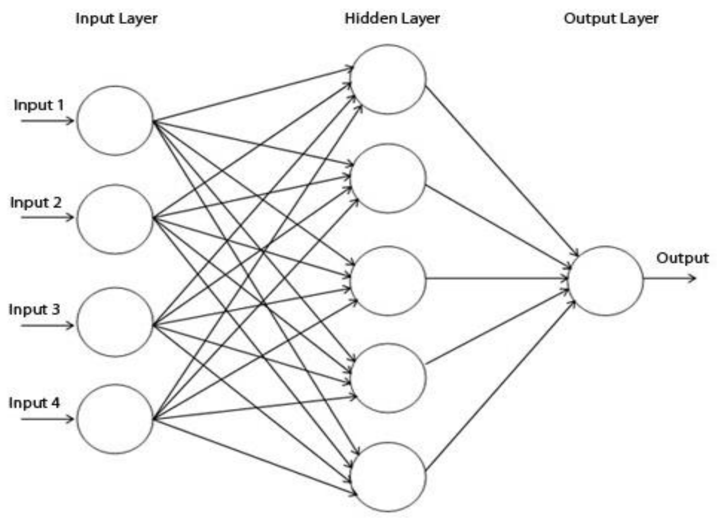

Figure 6 Multilayer perceptron Architecture with one layer for 4 inputs, 1 output and 5 neurons in the hidden layer

There is no definite analytical method to find the number of neurons in the hidden layers. The number of neurons in the hidden layer is changed iteratively till the out of sample error metric is the lowest. Each neuron is also associated with an activation function. Mathematically,

For Input layer neurons (with i=1,2,..p. p being the number of neurons in the input layer or equals the number of input variables)

$$n_i = x_i \qquad \text{Eq. (15)}$$

where n_i are the outputs of the neuron from input layer

x_i are the input variables

For hidden layer neurons with q number of neurons in the hidden layer.

$$n_j = g(b_j + \sum_{i=1,j=1}^{p,q} w_{ji} * n_i) \qquad \text{Eq. (16)}$$

Where g(.) is the activation function of the hidden layer and the function is usually a nonlinear function that converts the linear combination of input variables into nonlinear combination. Commonly used activation functions are hyperbolic tangent function, log-sigmoidal function etc. In this study, tan-sigmoid function is used as activation function to train the neural network. This function is explained elaborately in the following section.

b_j are the biases of neurons of the hidden layer

n_i are the outputs of the neuron from input layer

w_{ji} are the weights of connecting lines from input layer to the first hidden layer

For output layer neurons (with r being the number of neurons in the output layer or r equals the number of output variables)

$$n_k = f(b_k + \sum_{j=1,k=1}^{q,r} w_{jk} * n_j) \qquad \text{Eq. (17)}$$

Where n_k is the output of the output neuron or predicted output of the trained neural network model

f(.) is the activation function of the neurons of the output layer. This layer is generally linear function. But it is also possible to use the same activation function used for the hidden layers

b_k are the biased of the output layer neurons

n_j is the output of the neurons from the last hidden layer

w_{jk} are the weights of connecting lines from last hidden layer to output layer

Sigmoid function is a mathematical function which has a characteristic curve in the shape of 'S'. There are several forms of sigmoid function that are used as activation function of neurons in the artificial neural network. Logistic and hyperbolic tangent functions are commonly used functions. One form of variation of hyperbolic tangent function is tan-sigmoid function. Mathematically tan-sigmoid function is given by Eq. (18)

$$a = tansig(u) \quad \text{Eq. (18)}$$

Or

$$a = \frac{2}{(1 + e^{(-2*u)})} - 1 \quad \text{Eq. (19)}$$

The Figure 7 shows the tan-sig function characteristics

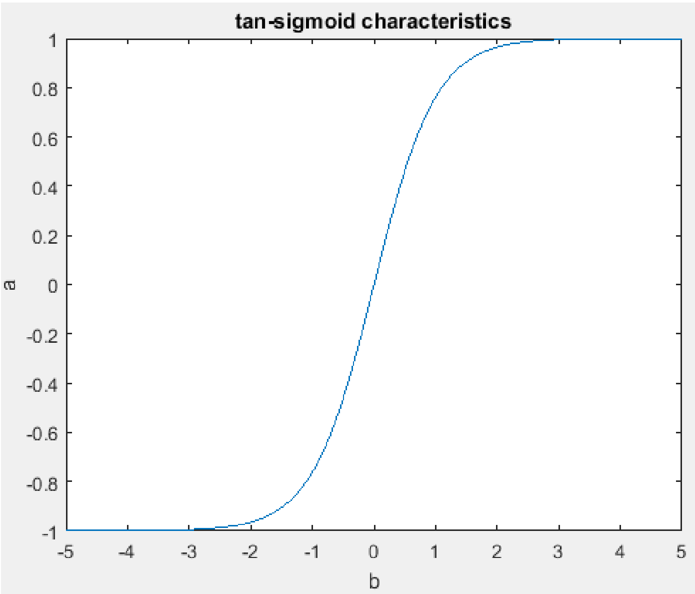

Figure 7 Tan-Sigmoid Transfer Function

For a regression problem, the output layer is generally modelled with a linear function.

Figure 8 shows the transfer function of linear relation.

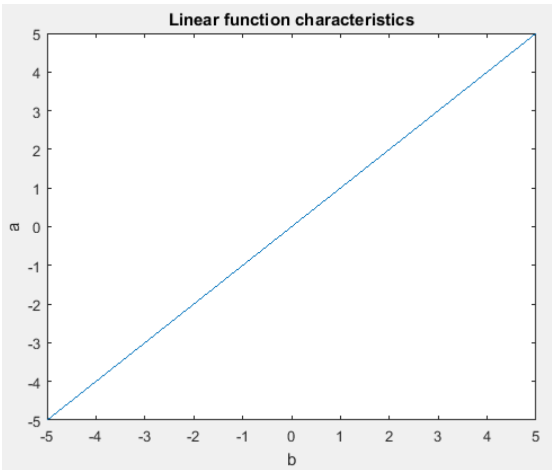

Figure 8 Linear Transfer Function

Mathematically, multilayer perceptron with one hidden layer is given by

$$\hat{y} = \sum_{k=1}^{r} b_k + \sum_{k=1}^{r}\sum_{j=1}^{q}\sum_{i=1}^{p} tansig\big(b_j + w_{ji} * x_i\big) * w_{jk} \qquad \text{Eq. (20)}$$

where $\hat{y}$ is the predicted output

x_i are the input variables (where i=1,2,…p. p is the number of input variables)

b_k are the biases of the neurons of the output layer (where k=1,2,..r. r is the number of neurons in the output layer or number of output variables)

b_j are the biases of the neurons in the hidden layer (where j=1,2,… q. q is the number of neurons in the hidden layer)

w_{ji} are the weights of the connecting lines from input layer to the first hidden layer

w_{jk} are the weights of connecting lines from last hidden layer to output layer

In this study, initially several neural networks were trained with all the input variables with different number of neurons in the hidden layer and different seeds using Eq. (20). The model that gives the least cross validation error was considered the optimal model. Sensitivity analysis was conducted for the optimal model and the input variable that contributed the least to find the output variable was removed from the input data set. Sensitivity analysis is elaborately explained in Section 5.2.2.2. Now the neural networks were trained again with different number of neurons and different seeds, only this time with input data set without the variable that contributes the least to find the output variable. New optimal model was found and sensitivity analysis was conducted on the new optimal model. Another input variable that contributed the least to the output variable is determined and eliminated. This step by step procedure was performed till the cross validation error is found lowest.

5.2.2.2 Sensitivity Analysis

5.2.2.2.1 Introduction

Sensitivity analysis is a method to find the contribution of individual input variables to find the output variables. In other words, sensitivity analysis explains the relative importance of input variables with output variables. There are various advantages of sensitivity analysis. David J Pannell, in his paper “Sensitivity analysis of normative economic models: theoretical framework and practical strategies” listed elaborately the advantages of conducting sensitivity analysis on the model. Some of the relevant advantages for this study are listed below (Pannell, 1997).

1. Identification of important variables – Sensitivity analysis helps in identifying the important input variables that affects the output variable.
2. Model Simplification – This essentially helps in understanding the variable that has least impact on the output and helps to eliminate from the model
3. Model Calibration – Sometimes too many input variables in the model reduces the generalising capacity of the model. Hence sensitivity analysis helps in removing those input variables that makes an over fit model.

There are several methods to perform sensitivity analysis in an artificial neural network. Gevrey et al (Gevrey, Dimopoulos, & Lek, 2003), explained and ranked seven methods to find the relative importance of each input variables on the output variable in an Artificial neural network. They are

1. Partial derivative method (PaD)
2. The weights method
3. Perturb method
4. The Profile method
5. Classical Stepwise method
    a. Improved stepwise A
    b. Improved stepwise B

Table 1 shows the ranking of different methods of finding the relative importance between input variables and output variables.

Table 1 Ranking of methods to find relative importance of independent variables on dependent variables (Gevrey, Dimopoulos, & Lek, 2003)

RANK	METHOD
1	PaD method
2	Profile method
3	Perturb method
4	Weights method
5	Improved stepwise method A
6	Improved stepwise method B
7	Classical stepwise method

From here, PaD (Partial Derivative) method will be used to find the relative importance of the input variables on the output variable. The following section illustrates this method with a simple example.

5.2.2.2.2 Partial derivative method (PaD)

Partial Derivative method is a mathematically sound method to find the individual contribution of input variables on output variables. This method is based on finding the partial derivative of output with respect to each input, keeping all other inputs constant, and finding the relative importance through the result (Gevrey, Dimopoulos, & Lek, 2003).

The predicted value of the artificial neural network is given by

$$\hat{y} = \sum_{k=1}^{r} b_k + \sum_{k=1}^{r}\sum_{j=1}^{q}\sum_{i=1}^{p} tansig\left(b_j + w_{ji} * x_i\right) * w_{jk} \qquad \text{Eq. (21)}$$

The partial derivative of the predicted value of artificial neural network is given by

$$\frac{\partial \hat{y}}{\partial x_i} = \sum_{k=1}^{r}\sum_{i=1}^{p}\sum_{j=1}^{q} \frac{\partial}{\partial x_i}\left(tansig\left(b_j + w_{ji} * x_i\right)\right) * w_{jk} \qquad \text{Eq. (22)}$$

From Eq. (18) and (19)

$$tansig(u) = \frac{2}{(1 + e^{(-2*u)})} - 1 \qquad \text{Eq. (23)}$$

Partial derivative of tansig(u) is given by

$$\frac{\partial}{\partial u}(tansig(u)) = \frac{(4 * e^{-2*u})}{(e^{-2*u} + 1)^2} \qquad \text{Eq. (24)}$$

Hence Eq. (22) is rewritten as

$$S_i = \frac{\partial \hat{y}}{\partial x_i} = \sum_{k=1}^{r}\sum_{i=1}^{p}\sum_{j=1}^{q} \frac{\left(4 * e^{-2*(b_j + w_{ji}*x_i)}\right)}{\left(e^{-2*(b_j + w_{ji}*x_i)} + 1\right)^2} * w_{jk} \qquad \text{Eq. (25)}$$

Once the partial derivatives of each input are found, each of the data points in the in-sample is substituted to the partial derivative and the value of substitution is found and tabulated. The mean and standard deviation of the partial derivatives are calculated for the training set and the distance $D_i = \sqrt{mean(S_i)^2 + std(S_i)^2}$ for each input variable is calculated as a measure of its relevance. The input variable corresponding to the lowest distance is considered the least important input to find the output variable (White & Racine, 2001).

Sensitivity analysis is illustrated empirically using a simple system with four input variables x_1, x_2, x_3, and x_4 and an output variable y. The relation between the inputs and output is given as follows.

$$y = \tanh(x_1) + \sinh(x_4) \qquad \text{Eq. (26)}$$

It can be observed from Eq. (26), only the input variables x_1 and x_4 affect the output y but not the input variables x_2 and x_3. A thousand observations of random numbers are created for each input variables. Several neural networks were developed using the activation function – tan-sigmoid function. The predicted output is given by the Eq. (21). Now partial derivative of the predicted output is determined by Eq. (25)

Now data points or observations of each input variables are substituted in Eq. (25). For the obtained result of the substitution, the mean and standard deviation is calculated. Figure 9 shows the graphical representation of mean vs. standard deviation.

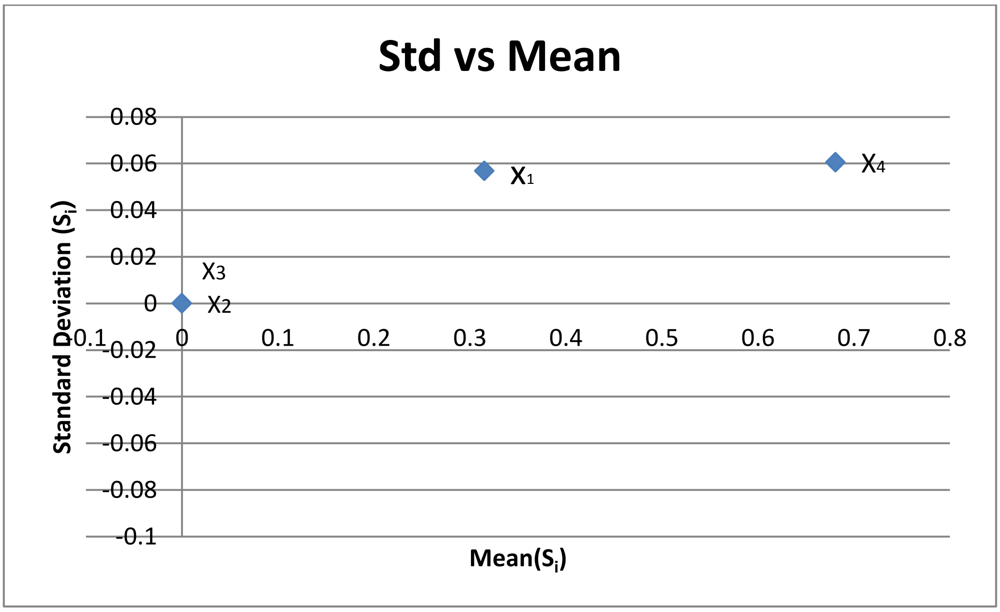

Figure 9 Graph between Standard Deviation and Mean for the example system considered

The distance between origin and the points, x_i(mean, standard deviation), is calculated as per equation (27).

$$Distance = \sqrt{(mean)^2 + (standard\ deviation)^2} \qquad \text{Eq. (27)}$$

Table 2 shows the distance between standard deviation and mean for the example system considered.

Table 2: The distance between Standard Deviation and Mean after sensitivity analysis for the example system

Input Variables	Mean	Standard deviation	Distance = sqrt(mean^2+SD^2)
x_1	0.315183	0.05679	0.320258123
x_2	-6.50E-07	3.68E-06	3.73E-06
x_3	-2.60E-06	4.34E-06	5.04E-06
x_4	0.680999	0.06058	0.683688376

The distance is then ranked from smallest to highest. The one with smallest distance is the least important variable that influences the output variable. From Table 2, it is evident that input variables x_3 and x_2 are the least important variables to determine the output variable and x_1 and x_4 are the most important variables to determine output variable y. Recalling the system equation, $y = \tanh(x_1) + \sinh(x_4)$, the method detects the right input variables effectively. This is represented graphically in Figure 10

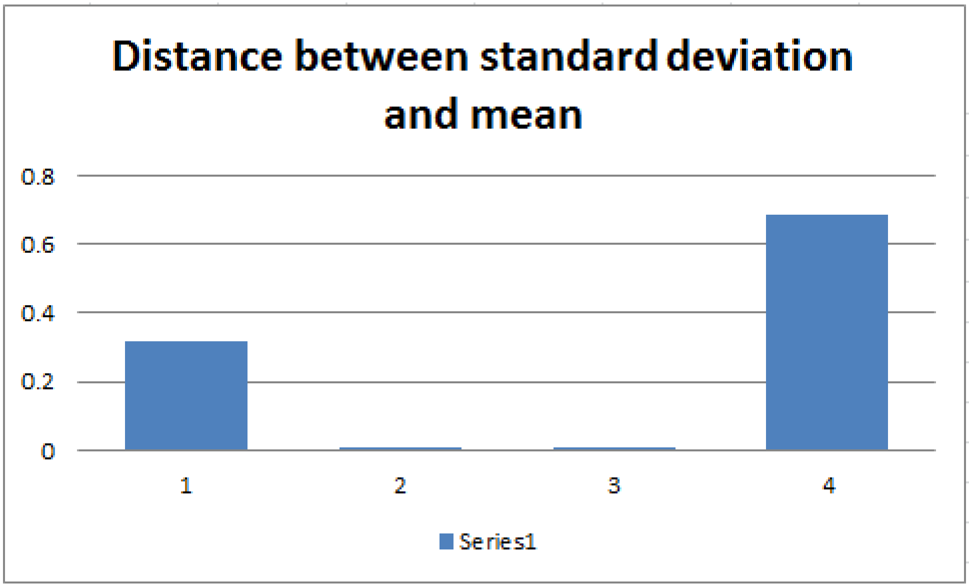

Figure 10 Distance between standard deviation and mean

5.2.3 Support Vector Machine

Support Vector Machines, also known as SVM, is a popular machine learning technique since it had been proposed by Vladimir Vapnik and his colleagues (V.Vapnik, 2013). The starting point to understand SVM regression is SVM classification. In the following section, a system of two feature variables x_1 and x_2 are used to illustrate the SVM classification problem. From classification, the regression problem is further extended.

Assuming there are two features, x_1 and x_2, in the problem, both variables can be represented graphically as follows

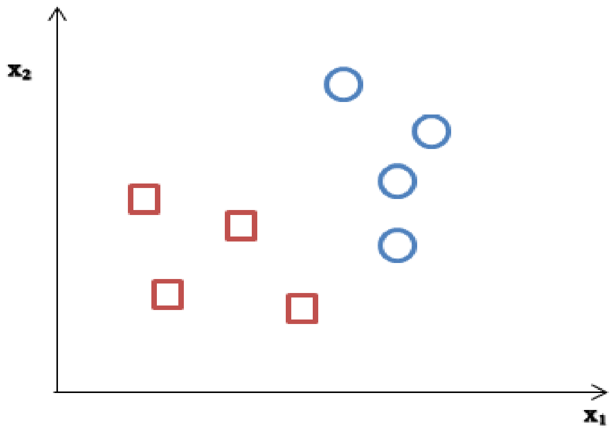

Figure 11 Feature Variables x_1 vs x_2

The basic idea of SVM classification method is to find the hyper plane that classifies the entire data into two classes. But it is to be noted that, various hyper planes (shown by green lines) exists as shown below. Hyper planes are the lines that classify the entire data points into two classes. For simplicity, let's assume the two classes as Class 1 and Class 2. In the following figure, data points that are left to the hyper plane falls under Class 1 and right to the hyper plane falls under Class 2. Given data points, several hyper planes can be drawn to classify the data points. Figure 12 (OpenCV, 2011) shows the data points classified by several hyper planes (represented by dark green lines)

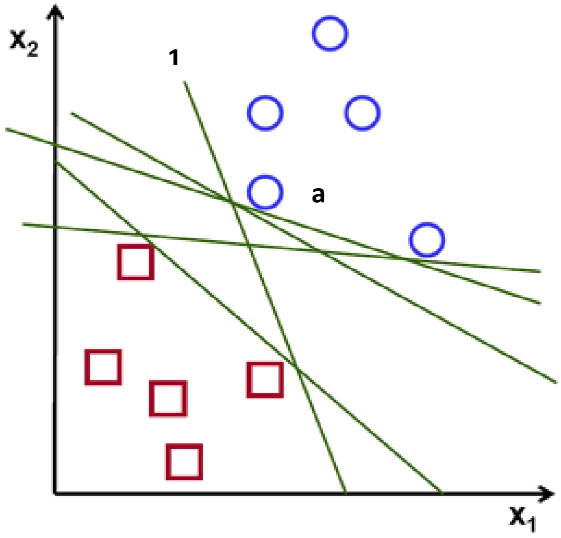

Figure 12 Feature Variables X_1 vs. X_2

In the Figure 12 (OpenCV, 2011) above, assume the line 1(shown by green) and the point 'a' (shown by blue circle) as the hyper plane and one of the data points from Class 2. Assuming there is a small error in measuring the data point 'a', the hyper plane '1' would classify the data point 'a' under Class 1 instead of Class 2. Hence the problem is to find the most optimal hyper plane. That leaves the question – What is an optimal plane and how to find it?

The optimal plane is the hyper plane that allows higher margin between the hyper plane and the nearest data point to the hyper plane on both sides.

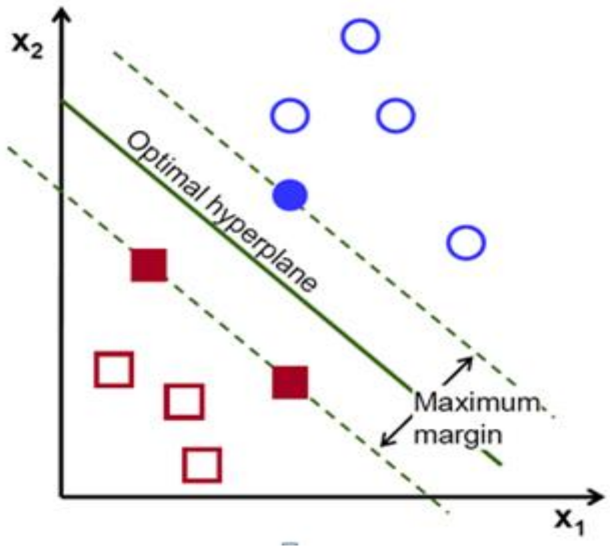

Figure 13 Feature Variables X1 vs. X2 with optimal hyper plane

In the Figure 13 (OpenCV, 2011), the optimal hyper plane (shown by continuous green line) solves the problem of misclassification. For instance, if there is a small perturbation in measuring the nearest data point, it still classifies the class correctly. Hence the best way to find the optimal hyper plane is to find the hyper plane that gives the maximum margin between the hyper plane and the nearest data points from both classes. Hence this problem can be converted into an optimization problem with the objective function being Maximisation of the margin.

Mathematically, the hyper plane is represented as follows

$$g(x) = w^T.x + w_0 \qquad \text{Eq. (28)}$$

where w^T is the weight vector or the slope of the hyper plane.

w_0 is the bias or the y intercept.

g(x) is the equation of the hyper plane (equation of the straight line)

Geometrically margin distance, Z, can be calculated by

$$Z = \frac{|g(x)|}{||w||} \qquad \text{Eq. (29)}$$

For a binary problem, $|g(x)|$ is equal to 1.

Hence Z becomes,

$$Z = \frac{1}{||w||} \qquad \text{Eq. (30)}$$

This Z is the distance between hyper plane and nearest data point of one of the classes. Hence the total margin Z_{TOT} is given as follows

$$Z_{TOT} = \frac{1}{||w||} + \frac{1}{||w||} \qquad \text{Eq. (31)}$$

Hence,

$$Z_{TOT} = \frac{2}{||w||} \qquad \text{Eq. (32)}$$

Now the problem is maximizing Z_{TOT} or it can be expressed in another way of minimising$\|w\|$. So far, the hyper planes and feature variables were dealt in a two dimensional coordinate plane. The same idea shall be extended to multi-dimensional coordinates (OpenCV, 2011).

The idea described thus far is a simple problem of classification with two feature variables. The same idea can be extended to a regression problem.

Assuming a system with u_i new samples and x_{ij} individual data set points. The Support Vector Machine Regression equation is given by the Eq. (33)

$$g(x) = b + \sum_{i=1}^{n} w^T * \varphi(x_i, u) \quad \text{Eq. (33)}$$

Where $\varphi_n(.)$ is the function that defines the relationship between the input and output variables. It is also called as Kernel function.

There are several Kernels that are used in SVM-Regression. They are given as follows (Kuhn & Johnson, 2013).

Linear Kernel

$$\varphi(x_i, u) = \ x_i * u \quad \text{Eq. (34)}$$

Polynomial Kernel

$$\varphi(x_i, u) = (x_i * u + c)^d \quad \text{Eq. (35)}$$

Where c is the bias and d is the degree of polynomial
Radial basis kernel

$$\varphi(x_i, u) = \exp(-\sigma * \|x_i - u\|^2) \quad \text{Eq. (36)}$$

where σ is the scaling parameter.

In this study, linear kernel is used to train the SVM regression model.

There are several advantages of using SVM over basic regression model.

In a linear regression model, the regression line reduces the sum of squared errors. Sum of squared errors are largely affected by outliers. If a large data point is introduced, then the regression line moves towards that point. In order to solve this problem, SVM introduces a threshold (often denoted by ε). If the data point lies within this threshold, then that points will not be used to determine the regression line. If the absolute residual value is greater than this threshold value, then this data point will be used to determine the regression line (Kuhn & Johnson, 2013). To find the model parameters, SVM technique introduces a cost function given by Eq. (37) (Zhou, Shi, & Li, 2011)

$$R(w,e) = \frac{1}{2}\|w\|^2 + \frac{1}{2}\|e\|^2 \qquad \text{Eq. (37)}$$

Where w is the weight vector and e is the residual which is the difference between the predicted value and observed value.

5.2.4 K-Nearest Neighbours Model

The K-Nearest neighbour technique is not as straight forward as artificial neural network and support vector machine, which were known for its ability to convert into mathematical equations to analyse and interpret analytically. K-Nearest Neighbour on the other hand identifies the value of the test set using the K nearest neighbours of the sample from the training set (Kuhn & Johnson, 2013).

It is to be noted that, the prediction using K – Nearest neighbour method is solely based on samples from the training data. Hence the mapping accuracy depends on the sample data to a great extent. Predicted response is the mean of the K –Nearest neighbours. Other statistical measure such as median can also be used (Kuhn & Johnson, 2013).

The most important parameter that influences the performance of this model is the definition of the distance between the data points. The most commonly used distance is Euclidean Distance (Kuhn & Johnson, 2013). Euclidean distance, in mathematics, is the length of the straight lie draw between two points in the Euclidean Space.

Consider two points P1(X1, Y2) and P2(X2,Y2) in the Euclidean Space. This is represented in Figure 14 (Dumas, 2017)

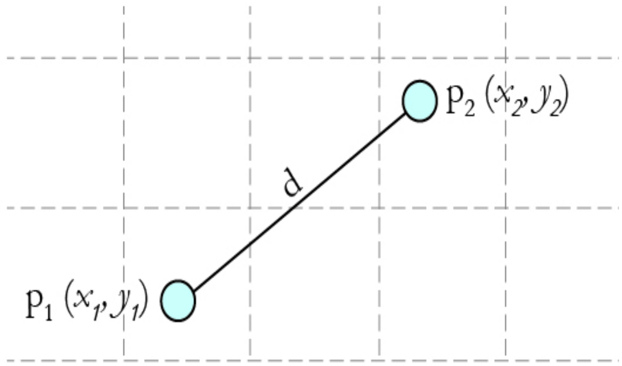

Figure 14: Euclidean plane with two points P1 and P2

The distance between the points is the length of the straight line, d which is given by the equation (38)

$$Distance = \sqrt{(X1 - X2)^2 + (Y1 - Y2)^2} \qquad \text{Eq. (38)}$$

Alternate way of representing the distance is using Minkowski distance formula. Minkowski distance formula is the generalised representation of the Euclidean distance, which is given as follows (Liu, 2007)

$$Distance = \sqrt[n]{(X1 - X2)^n + (Y1 - Y2)^n} \qquad \text{Eq. (39)}$$

It is to be noted if n = 1, then it is the Manhattan distance and if n = 2, then it is the Euclidean distance.

The two important problems exist with K-Nearest neighbour approach, mainly due to its over-dependence on the sampling data, namely problem of scaling and problem of missing data

K-Nearest neighbour approach depends on the training data. The input variables with high scale contribute more to the distance between the samples than input variables with lesser scales. One way to overcome this problem is to scale the input variables before using it in the model (Kuhn & Johnson, 2013).

Also if some sample data for input variables are missing, then it will give erroneous predictions. There are few ways to overcome this issue. Removing the input variable completely out of the model is one alternative but it is also the preferred alternative. Still this can be effective in case a huge amount of data samples are missing. Predicting the missing data using Naïve predictor method such as taking the mean of the preceding and succeeding data is another effective alternative.

It is not the case that more neighbours we consider, better the result is. Sometimes, k= 2 will give a better prediction that k=3 and so on. Hence finding an optimal 'k' value is of paramount importance. Low value of K over-fit the model and high value will under-fit (Kuhn & Johnson, 2013). Hence tuning of the model is required that involves finding the optimal number of neighbours to be considered for the purpose of prediction.

The tuning of the method is often done by resampling. One way to cross validate the model is by trial and error and calculate the cross validation error for each value of k. For example, consider a system for which prediction needs to be made using K-Nearest neighbour method. By assuming k= 1 to 20, the cross validation error metric can be found for each value of K. An example error metric (Root-mean-square-error (RMSE)) is taken for explanation purpose and a graph is shown in Figure 15 (Kuhn & Johnson, 2013).

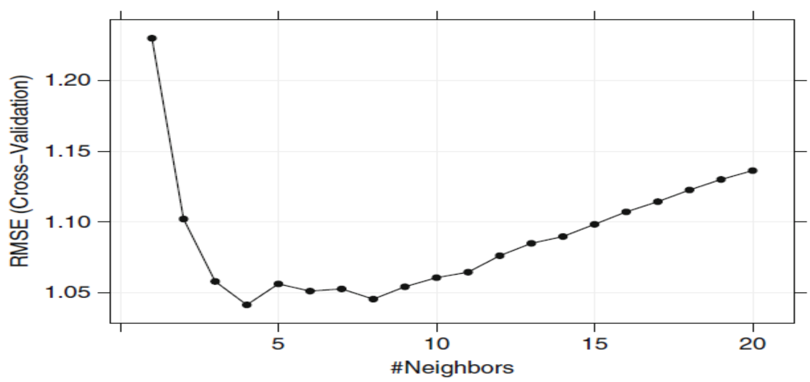

Figure 15: The profile of RMSE with respect to number of neighbours

From the Fig 14 it can be seen that, cross validation error (RMSE) is lowest for k=4. Hence K=4 is considered the most ideal value of k for the forecasting problem of this system.

6 Error Metrics

The following section explains the commonly used error metrics in the wind power forecasting.

6.1 Goodness of fit

The Goodness of fit expresses generalisation capability of the trained model. In other words, Goodness of fit tells how well a model is capable of predicting the new ouputs given new inputs. Goodness of Fit helps to evaluate the discrepancy between the observed value and predicted value of the out of sample. To measure the Goodness of fit, several error metrics are available. Each measure has its own advantage and disadvantage. Below, some of the most commonly used Error metrics are explained briefly.

6.2 Popular Error metrics

There are several error metrics that can be used to compare the forecasted wind power to the actual wind power. Some of the most commonly used error metrics are

1. Mean Squared Error (MSE)
2. Root Mean Square Error (RMSE)
3. Mean Absolute Error
4. Mean Absolute Scaled Error (MASE)
5. Mean Absolute Percentage Error (MAPE)
6. Weighted Absolute Percentage Error (WAPE)

6.2.1 Mean Squared Error

This metric measures the average of squares of the deviation between the observed value and forecasted value (Hyndman & Koehler, 2006). Mathematically,

$$MSE = \frac{1}{T}\sum_{t=1}^{T}(\widehat{P}_t - P_t)^2 \qquad \text{Eq. (40)}$$

Where $\widehat{\boldsymbol{P}}$ is the predicted value and P is the actual value, T is the total number of data points.

6.2.2 Root Mean Square Error

This metric is the square root of the mean squared error (Hyndman & Koehler, 2006)

$$RMSE = \sqrt{MSE} \qquad \text{Eq. (41)}$$

(or)

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(\widehat{P_t} - P_t)^2} \qquad \text{Eq. (42)}$$

Where $\widehat{\boldsymbol{P}}$ is the predicted value and P is the actual value.

6.2.3 Mean Absolute Error

This metric is the mean of the absolute deviation between the forecasted value and the observed value (Hyndman & Koehler, 2006). Mathematically

$$MAE = \frac{\sum_{t=1}^{T}|e_t|}{T} \qquad \text{Eq. (43)}$$

Where e_t is the deviation between the forecasted value and observed value (=$\left(\hat{P}_t - P_t\right)$) for the whole data points.

6.2.4 Mean Absolute Scaled Error

Hyndman et al, in their paper, proposed a new method of measuring the errors for the purpose of forecasting using the in sample error called Scaled error. The Scale error, mathematically, is

$$S_t = \frac{1}{T}\sum_{t=1}^{T}\left(\frac{|e_t|}{\frac{1}{(T-1)}\sum_{t=2}^{T}|P_t - P_{t-1}|}\right) \qquad \text{Eq. (44)}$$

Where S_t is the scaled error,

e_t is the deviation between the forecasted value and observed value (=$\left(\hat{P} - P\right)$)

P_i is the observed in sample value at t

P_{i-1} is the observed in sample value at t-1

$$MASE = mean(|S_t|) \qquad \text{Eq. (45)}$$

6.2.5 Mean Absolute Percentage Error

In the case of normalised data set, it is advantageous to use percentage errors to make a clear comparison as the absolute and scaled errors tends to give error values way too small to make any comparison. Mean Absolute Percentage Error, shortly MAPE, is one of the most widely used percentage error metrics to solve this issue.

The percentage Error is given by (Hyndman & Koehler, 2006)

$$MAPE = \frac{100}{T}\sum_{t=1}^{T}\left|\frac{\widehat{P_t} - P_t}{P_t}\right| \qquad \text{Eq. (46)}$$

6.2.6 Weighted Absolute Percentage Error

The problem with MAPE is that when the wind power falls to 0, then it will give an infinite error due to the denominator term. In order to address this issue, Weighted Absolute Percentage Error, WAPE, is used widely for expressing percentage of power forecasting models.

Weighted Absolute Percentage Error is given by

$$WAPE = \frac{\sum_{t=1}^{T}\left|P_t - \hat{P}_t\right|}{\sum_{t=1}^{T}\left|P_t\right|} \qquad \text{Eq. (47)}$$

where n= number of samples

When the normalised data points are used, the Mean Squared Error (MSE), Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Scaled Error measures will be very small to make a comparison. Hence expressing the errors in terms of percentage is the best solution for normalised data set. With Mean Absolute Percentage Error (MAPE), if the wind power value reaches zero, the denominator of Eq. (46) will become zero. Therefore, error measure will be always infinity. To overcome this problem, Weighted Absolute Percentage Error (WAPE) is used. In this study, Weighted Absolute Percentage Error is used from here to study and compare the performance of each forecasting model.

7 Empirical Analysis of different short term wind power forecasting models

The following section illustrates the input used and results obtained from the models.

7.1 Variables used in the models

Before using the statistical methods to improve the forecast of wind power, it is important to use the right input variables in the model. In Section 4, the data collected for the study is discussed briefly. The available data are production data of 10 wind farms in a portfolio, the meteorological forecast of wind power for those 10 wind farms, the total wind power of the portfolio and the meteorological forecast of total wind power of the portfolio.

Along with the collected data, the data set is further extended by introducing delay in the data points to capture the dynamics of the data. In other words, the forecasting of wind power does not only depend on the data at time, t but also at time t-1 and t-2. For example, production data of wind farm 1 at time period, t-1 is available. The production of wind farm 1 at time t-2 is also used as another input variable in the model. Table 3 shows the list of input and output variables used in the models with their time period. Also Table 3 shows the notation used for each input variables in the models.

Table 3 List of variables used in the study

SNO	Variable	Variables used in the study	Time Period	Notations Used
1	INPUT	Production of Wind farm 1	t-1	D1_PROD1
2		Production of Wind farm 2	t-1	D1_PROD2
3		Production of Wind farm 3	t-1	D1_PROD3
4		Production of Wind farm 4	t-1	D1_PROD4
5		Production of Wind farm 5	t-1	D1_PROD5
6		Production of Wind farm 6	t-1	D1_PROD6
7		Production of Wind farm 7	t-1	D1_PROD7
8		Production of Wind farm 8	t-1	D1_PROD8
9		Production of Wind farm 9	t-1	D1_PROD9
10		Production of Wind farm 10	t-1	D1_PROD10
11		Production of Wind farm 1	t-2	D2_PROD1
12		Production of Wind farm 2	t-2	D2_PROD2
13		Production of Wind farm 3	t-2	D2_PROD3
14		Production of Wind farm 4	t-2	D2_PROD4
15		Production of Wind farm 5	t-2	D2_PROD5
16		Production of Wind farm 6	t-2	D2_PROD6
17		Production of Wind farm 7	t-2	D2_PROD7

SNO	Variable	Variables used in the study	Time Period	Notations Used
18		Production of Wind farm 8	t-2	D2_PROD8
19		Production of Wind farm 9	t-2	D2_PROD9
20		Production of Wind farm 10	t-2	D2_PROD10
21		Meteorological forecast of Wind farm 1	t	EST1
22		Meteorological forecast of Wind farm 2	t	EST2
23		Meteorological forecast of Wind farm 3	t	EST3
24		Meteorological forecast of Wind farm 4	t	EST4
25		Meteorological forecast of Wind farm 5	t	EST5
26		Meteorological forecast of Wind farm 6	t	EST6
27		Meteorological forecast of Wind farm 7	t	EST7
28		Meteorological forecast of Wind farm 8	t	EST8
29		Meteorological forecast of Wind farm 9	t	EST9
30		Meteorological forecast of Wind farm 10	t	EST10
31		Meteorological forecast of Wind farm 1	t-1	D1_EST1
32		Meteorological forecast of Wind farm 2	t-1	D1_EST2
33		Meteorological forecast of Wind farm 3	t-1	D1_EST3
34		Meteorological forecast of Wind farm 4	t-1	D1_EST4
35		Meteorological forecast of Wind farm 5	t-1	D1_EST5
36		Meteorological forecast of Wind farm 6	t-1	D1_EST6
37		Meteorological forecast of Wind farm 7	t-1	D1_EST7
38		Meteorological forecast of Wind farm 8	t-1	D1_EST8
39		Meteorological forecast of Wind farm 9	t-1	D1_EST9
40		Meteorological forecast of Wind farm 10	t-1	D1_EST10
41		Meteorological forecast of Wind farm 1	t-2	D2_EST1
42		Meteorological forecast of Wind farm 2	t-2	D2_EST2
43		Meteorological forecast of Wind farm 3	t-2	D2_EST3
44		Meteorological forecast of Wind farm 4	t-2	D2_EST4
45		Meteorological forecast of Wind farm 5	t-2	D2_EST5
46		Meteorological forecast of Wind farm 6	t-2	D2_EST6
47		Meteorological forecast of Wind farm 7	t-2	D2_EST7
48		Meteorological forecast of Wind farm 8	t-2	D2_EST8
49		Meteorological forecast of Wind farm 9	t-2	D2_EST9
50		Meteorological forecast of Wind farm 10	t-2	D2_EST10
51		Total production of all wind farms	t-1	D1_PROD_TOT
52		Total production of all wind farms	t-2	D2_PROD_TOT
53		Meteorological forecast of all wind farms	t	EST_TOT
54		Meteorological forecast of all wind farms	t-1	D1_EST_TOT
55		Meteorological forecast of all wind farms	t-2	D2_EST_TOT
56	OUTPUT	Total production of all wind farms	t	PROD_TOT

where D represents the Time delay

7.2 Performance of the meteorological forecast

The primary goal of the study is to improve the short term forecast of wind power. One of the forecasts that are available is meteorological forecast of wind power. This forecasted wind power can be bid in the market. But still this forecast can be improved. Hence in this study, the performance of meteorological forecast is used as the benchmark model. The WAPE (Weighted Absolute Percentage Error) is calculated between the meteorological forecasted wind power and the actual observed wind power for both in-sample and out of sample data. The formula to calculate the WAPE is given by Eq. (48)

$$WAPE = \frac{\sum_{t=1}^{T} |P_t - \hat{P}_t|}{\sum_{t=1}^{T} |P_t|} \qquad \text{Eq. (48)}$$

The Weighted Absolute Percentage Error between the Meteorological forecast wind power and observed wind power for in sample and out of sample is given in the Table 4.

Table 4 WAPE between Meteorological forecast of wind power and the Observed Wind Power (in %)

Meteorological forecast error Expressed in percentage (%)	
In-sample WAPE	**Out-of-sample WAPE**
19.47528554	17.77047116

7.3 Performance of the Persistence method or naïve predictor

The naïve predictor method is commonly used as benchmark model due to its effectiveness and simplicity of usage in short term horizon. The persistence model, in simple terms, is whatever the observed wind power now is going to be wind power in the next hour. Mathematically, the naïve predictor is expressed in Eq. (49)

$$P(t+1) = P(t) \qquad \text{Eq. (49)}$$

WAPE (Weighted Absolute Percentage Error) is calculated between the wind power at t and wind power at t-1. WAPE of in sample and out-of-sample using persistence method is given in Table 5.

Table 5 WAPE of Naive Predictor technique (in %)

Naïve predictor method error Expressed in percentage (%)	
In-sample WAPE	**Out-of-sample WAPE**
12.57028525	13.82242777

7.4 Performance of Stepwise Linear regression model

The general mathematical form of linear regression model is given by Eq. (50)

$$y = \beta_0 + \sum_{i=1}^{i} \beta_i * x_i + \varepsilon \qquad \text{Eq. (50)}$$

The stepwise linear regression model is conducted using IBM SPSS Statistics 22. In this approach, all the 55 input variables discussed in section 7.1 is given as input variables. The estimation used is stepwise estimation method. As discussed in section 5.1, stepwise estimation creates a linear regression model with the output variable and one input variable that has the highest correlation with the output variable. The input variable with the second highest correlation is then determined. Another linear regression model is created using the current inputs in the model and the newly found second input. Each input is removed one by one and the significance of that input with the output is estimated. If the input is non-significant, then the input is removed permanently. If the input is significant, then the input is kept in the model. This step by step process is iteratively performed until all the input variable with significance is added in the model. Appendix 1 shows all the linear models determined in step by step estimation. Table 6 below shows the summary of the final model determined.

Table 6 Summary of the final linear regression model using stepwise estimation

	Unstandardized Coefficients	**Sig.**
	B	
(Constant)	0.002493138	0.001273892
D1_PROD_TOT	1.038202183	0
EST_TOT	0.444184804	5.04775E-59

	Unstandardized Coefficients	Sig.
	B	
D2_EST_TOT	-0.063518196	0.002787591
D2_PROD_TOT	-0.162235833	6.01845E-33
D1_EST_TOT	-0.235588027	1.91713E-09
D1_PROD9	-0.055426048	5.95357E-14
EST8	0.102729503	5.33996E-10
D1_EST8	-0.058099255	0.000553254
D2_PROD9	0.020956397	0.000947971
D2_PROD8	-0.049859459	1.14103E-14
D1_PROD8	0.034821409	1.32319E-07
D2_PROD6	-0.016791636	1.14852E-07
D1_PROD1	-0.025332265	2.55571E-11
D1_EST6	0.020876229	6.58191E-07
EST1	0.012565939	0.006047911
D1_PROD5	-0.017536304	4.42002E-07
D1_PROD3	-0.014032034	0.018524962
EST5	0.009066895	0.020889853

From Table 6 it can be seen that only 18 variables out of 55 input variables are found significant as a result of stepwise estimation. Apart from that, Table 6 also shows the regression coefficients of the model, denoted by B. Note that the regression coefficients of all the input variables are significant at a chosen confidence level of $\alpha = 0.05$.

Eq. (51) shows the linear regression model in its mathematical form.

$$PROD\_TOT = 0.002493138 + 1.038202183 * D1\_PROD\_TOT + 0.444184804 * EST\_TOT - 0.063518196 * D2\_EST\_TOT - 0.162235833 * D2\_PROD\_TOT - 0.235588027 * D1\_EST\_TOT - 0.055426048 * D1\_PROD9 - 0.102729503 * EST8 - 0.058099255 * D1\_EST8 + 0.020956397 * D2\_PROD9 - 0.049859459 * D2\_PROD8 + 0.034821409 * D1\_PROD8 - 0.016791636 * D2\_PROD6 - 0.025332265 * D1\_PROD1 + 0.020876229 * D1\_EST6 + 0.012565939 * EST1 - 0.017536304 * D1\_PROD5 - 0.014032034 * D1\_PROD3 + 0.009066895 * EST5 \quad \text{Eq. (51)}$$

Using this Eq. (51), the in sample and out of sample output is predicted. From the predicted values, WAPE for in sample and out of sample are calculated and the results are presented below.

Table 7 WAPE of Stepwise Linear Regression Model (in %)

Stepwise linear regression error Expressed in percentage (%)	
In-sample WAPE	**Out-of-sample WAPE**
10.8443597	13.12598817

From Figure 16 it is observed that residuals or error terms scattered around the zero line. From Figure 17, the histogram of the residuals is given. It can be seen that the residuals approximate a normal distribution with mean 0.

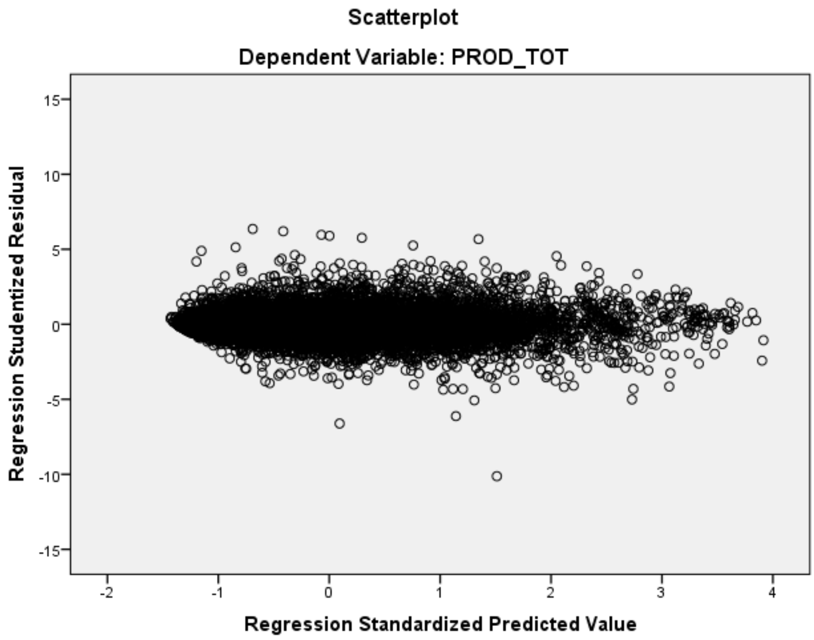

Figure 16 Standardized Residual Plot

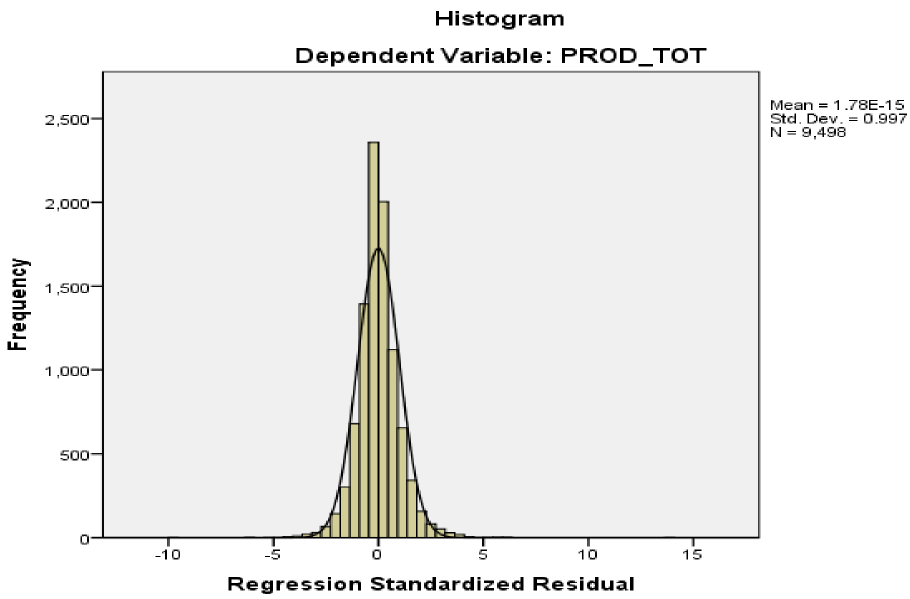

Figure 17 Histogram of the residuals

Another important assumption is error terms are normally distributed. From P-P Plot in Figure 18, it is observed that residuals are aligned along the normality line. Hence the error terms are normally distributed.

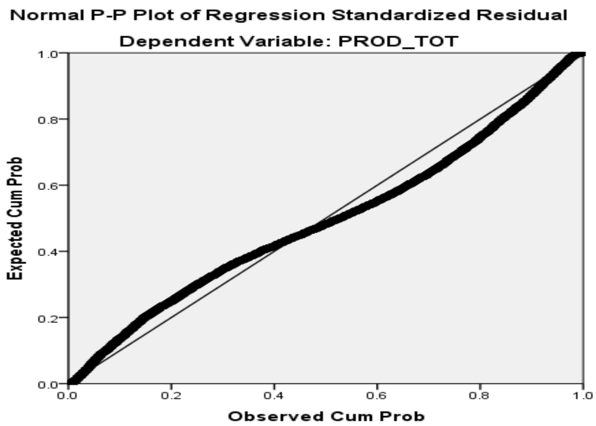

Figure 18 P-P Plot (Normal Probability Plot)

Durbin-Watson test is conducted to show that the error terms are independent of each other. If the value of ‘d’ of Durbin Watson test is close to 2, then it can be assumed that error terms are independent of each other. The result of Durbin Watson test shows value of ‘d’ as 1.980 which is very close to 2. Hence the error terms are independent of each other. The result of Durbin Watson test is attached in the Appendix 1.

7.5 Performance of the Artificial Neural Network

The Artificial neural Network is executed in MATLAB 2016a. The coding for the training of ANN and sensitivity analysis is given in Appendix 2. Using the 55 input variables, different artificial neural network models with different seed with different number of neurons are trained. The model with the smallest out of sample error is considered the optimal model. Sensitivity analysis is applied on this optimal model and the input with least importance in predicting the output is determined. This input is removed from the input data set. With the remaining inputs, different ANN models are trained and optimal model is found again. For that optimal model, sensitivity analysis is applied. This step by step approach is repeated iteratively till reducing the number of inputs to 10.

Table 8 shows the error measures of the optimal models during each iteration. Along with that, the input variable removed during every iteration is also shown in the Table 8.

It is to be noted that the lowest out of sample WAPE is found with 23 inputs and 3 neurons in the hidden layer. This model is chosen as the best model for the Artificial Neural Network.

Table 8 Summary of WAPE of ANN after removing an input variable using Sensitivity Analysis

Artificial Neural Network Error				
Expressed in percentage (%)				
Number of Inputs	Input variable removed	Number of Neurons	In-sample WAPE	Out-of-sample WAPE
55	D2_PROD4	1	0.108707974	0.131000585
54	D2_PROD10	2	0.108156248	0.131367913
53	EST10	1	0.108930043	0.131338521
52	EST3	2	0.108430467	0.13159849
51	D1_PROD8	1	0.108472347	0.131523393
50	D2_PROD1	3	0.108837601	0.130849305
49	EST6	1	0.108625287	0.131094176
48	D1_EST7	1	0.108743864	0.131119179
47	D2_EST7	1	0.1087513	0.131266397
46	D2_PROD6	1	0.108728541	0.131100521
45	D1_PROD4	1	0.108765201	0.130993799
44	D2_EST2	1	0.108917826	0.131413717
43	EST4	2	0.108934093	0.131425258
42	D2_PROD7	3	0.10828579	0.131432998
41	D2_PROD5	2	0.109847024	0.131806177
40	EST2	1	0.108891658	0.131851676
39	D2_PROD3	1	0.108642304	0.131368452
38	EST9	1	0.109650393	0.131556603
37	D2_EST9	2	0.10834233	0.131303587
36	D2_EST4	1	0.108890818	0.131016116
35	D1_EST10	1	0.109269462	0.131690377
34	D1_EST5	2	0.108583724	0.131673478
33	D1_EST2	1	0.108823109	0.130806617
32	D2_PROD2	1	0.108709504	0.131359626
31	D1_EST1	2	0.1089982	0.131574114
30	D1_EST4	4	0.109048654	0.130983136
29	D1_PROD2	2	0.109565262	0.130927795
28	D2_EST10	3	0.108483644	0.131310572
27	D1_PROD10	2	0.108591111	0.130962834
26	D2_EST5	2	0.109157265	0.130764055
25	D2_EST3	1	0.109162723	0.131190395
24	D2_EST1	1	0.108993483	0.131013128

Artificial Neural Network Error				
Expressed in percentage (%)				
Number of Inputs	**Input variable removed**	**Number of Neurons**	**In-sample WAPE**	**Out-of-sample WAPE**
23	D2_EST8	3	0.108279916	0.130780487
22	D1_EST9	1	0.109031321	0.13085534
21	D1_PROD6	1	0.108960539	0.131332903
20	D1_PROD7	2	0.109123985	0.131079486
19	D1_EST3	11	0.107965094	0.131082321
18	EST7	1	0.109643616	0.131258318
17	D1_PROD5	1	0.109650625	0.131563486
16	EST1	3	0.109144398	0.131635593
15	D2_EST6	3	0.109046576	0.131638206
14	D1_PROD3	2	0.110153724	0.131732941
13	EST5	3	0.109222539	0.131739359
12	D1_PROD1	1	0.109327925	0.131934046
11	D2_PROD9	2	0.109402501	0.1318518

7.6 Performance of the Support Vector Machine

The Support vector Machine is executed in R Studio using the library “e1071”. The R code for the Support vector machine is given in Appendix 3.

Using the function svm() from the “e1071” library, an initial SVM model is trained. Two important tuning parameters in the SVM model are ε and cost function. For the initial SVM model, these values are assumed the default values that are shown in Table 9. This model is made as the reference model. Now the reference SVM model is tuned for different values of ε and cost function. R studio provides a function that allows us to set a range of values for a parameter and find the best model. The function is called tune(). The best way to tune a SVM regression model is by fixing ε value and changing the cost function. This is because cost function provides more flexibility (Kuhn & Johnson, 2013). The cost function is changed over a range from 0.001 to 1. At the cost function 0.001, the best model is obtained. For this best model, out of sample and in sample WAPE are calculated and tabulated in Table 9.

It can be seen from the Table 9 that ε is fixed at 0.1 and the other parameters are changed over a range value to find the best model.

Table 9 Summary of Support Vector Machine and WAPE

Support Vector Machine - Regression Error			
Expressed in percentage (%)			
Model	**Model Parameter**	**In-sample WAPE**	**Out-of-sample WAPE**
Before Tuning	SVM-Type: eps-regression	10.79380834	13.26017863
	SVM-Kernel: linear		
	cost: 1		
	gamma = 1/(data dimension)		
	epsilon: 0.1		
After Tuning	SVM-Type: eps-regression	11.06425199	13.19285875
	SVM-Kernel: linear		
	cost: 0.001		
	gamma: 0.001		
	epsilon: 0.1		

7.7 Performance of the K-Nearest Neighbour Algorithm

The K Nearest Neighbour Algorithm is executed in R studio using the library “caret”. The R code for the K-Nearest Neighbour Algorithm is given in Appendix 4.

For K-Nearest Neighbour regression, knnreg() function under the library “caret” is used. As explained in the Section 5.2.4, finding the optimal value of k is of paramount importance. In this study, K-Nearest neighbour model is trained for k values from 1 to 30. For each k value, in sample and out of sample WAPE measure is calculated and tabulated in Table 10.

Table 10 Summary of WAPE using K-Nearest Neighbour algorithm

K-Nearest Neighbour method error		
Expressed in percentage (%)		
K	**In Sample WAPE**	**Out-of-sample WAPE**
1	7.52655E-14	21.82490291
2	7.277582662	19.65362711
3	7.365428386	18.71702448
4	8.779580172	18.01450138
5	9.758502936	17.65298768

K-Nearest Neighbour method error Expressed in percentage (%)		
K	In Sample WAPE	Out-of-sample WAPE
6	10.44055757	17.33329539
7	10.95358056	17.30089002
8	11.35815845	17.21620673
9	11.73128653	17.10461376
10	11.98068836	16.99788351
11	12.157292	16.96538242
12	12.34796386	16.92282612
13	12.48473624	16.85642078
14	12.62757026	16.8176745
15	12.73777582	16.7696513
16	12.86419412	16.77249499
17	12.98265301	16.75808236
18	13.09767085	16.74116004
19	13.21355992	16.72621784
20	13.30076399	16.71231932
21	13.37790485	16.70295431
22	13.4474652	16.7036186
23	13.50800957	16.71355635
24	13.58043437	16.68369156
25	13.64196664	16.69131082
26	13.69875099	16.69367877
27	13.77486429	16.71005442
28	13.81781067	16.71143487
29	13.86625728	16.7073663
30	13.91179551	16.72606622

It can be observed that the lowest WAPE value is found for k=24. Hence this value is considered the optimal k value and the model obtained is the optimal KNN model.

7.8 Performance of individual wind farms

To compare the performance of individual wind farms with portfolio of wind farms, three wind farms are selected. From the above models, Artificial Neural Network is one of the important techniques used for the portfolio forecast. The same approach is used to train the model to predict the wind power output of individual wind farms using the available data. The inputs for the prediction of individual wind farm power are production data of all the wind farms, and meteorological forecast of the individual wind farm. Around 1 to 50 neurons were given in the hidden layer and models were trained for each number of neurons iteratively. For each model, WAPE of in sample and out of sample are calculated. The best performance of the entire model is presented in Table 11 for first three wind farms.

Table 11 Summary of WAPE of Individual wind farms

	Individual vs. Portfolio error Expressed in percentage (%)	
Wind Farm	**In-sample WAPE**	**Out-of-sample WAPE**
Wind Farm 1	21.0475135	21.1549595
Wind Farm 2	23.5628343	22.8738399
Wind Farm 3	23.6078241	27.9737239

8 Conclusions

8.1 Summary of the main findings

In this study, Weighted Absolute Percentage Error (WAPE) of out of sample is used as the goodness of fit measure. Table 12 shows the summary of in sample and out-of-sample WAPE error obtained from all the hybrid approaches used in the study. Also different techniques are ranked from 1 to 6 based on the Out-of-sample WAPE. It can be observed from the Table 12 that Artificial Neural Network with meteorological forecast of wind power performed the best among all the hybrid approaches. But Artificial Neural Network performs only slightly above the linear regression model. Followed by linear regression model are the Support vector Machine, Naïve Predictor or Persistence Method, and K-Nearest Neighbour Algorithm respectively.

Table 12 Summary of Error Matrix

OVERALL ERROR MATRIX			
Expressed in percentage (%)			
RANK	**MODEL/METHODOLOGY/ALGORITHM**	**In-sample WAPE**	**Out-of-sample WAPE**
1	Artificial Neural Network	10.8279916	13.0780487
2	Stepwise Linear Regression Model	10.8443597	13.12598817
3	SVM Regression	11.06425199	13.19285875
4	Naïve Predictor or Persistence Method	12.57028525	13.82242777
5	K-Nearest Neighbour Algorithm	13.58043437	16.68369156
6	Meteorological Forecast	19.47528554	17.77047116

8.2 Conclusion and Implication

8.2.1 What is the performance of Meteorological forecast in comparison with the Persistence method?

It can be observed from the Table 13 that Persistence model betters Meteorological Forecast by 22.21%. Hence one of the best and easiest methods to forecast the wind power one hour ahead and bid in the market is by Persistence method as it performs better than Meteorological forecast.

Table 13 Comparison between Meteorological forecast vs. Persistence method

Comparison between Meteorological Forecast vs. Naïve Predictor Expressed in percentage (%)				
Model 1	**Model 2**	**Out of Sample WAPE of Model 1**	**Out of Sample WAPE of Model 2**	**Percentage improvement (%)** $(\frac{WAPE_1 - WAPE_2}{WAPE_1} * 100)$
Meteorological Forecast	Naïve Predictor or Persistence method	17.77047116	13.82242777	22.21%

8.2.2 What is the performance of Meteorological forecast in comparison with hybrid approach using linear regression model?

From the Table 14, it can be observed that linear regression model bettered meteorological forecast by 26.13%. Hence linear regression model gives a better performance than the meteorological forecast.

Table 14 Comparison between Meteorological forecast vs. linear regression model

Comparison between Meteorological Forecast vs. linear regression model Expressed in percentage (%)				
Model 1	**Model 2**	**Out of Sample WAPE of Model 1**	**Out of Sample WAPE of Model 2**	**Percentage improvement (%)** $(\frac{WAPE_1 - WAPE_2}{WAPE_1} * 100)$
Meteorological Forecast	Linear regression model	17.77047116	13.12598817	26.13%

8.2.3 What is the performance of Meteorological forecast in comparison with hybrid approach using nonlinear regression model?

From the Table 15, it can be observed that all the nonlinear models perform better than meteorological forecast of wind power. Out of the nonlinear models artificial neural network provides the best performance. Followed by ANN, are SVM, and KNN. The performance of KNN is small compared to other nonlinear models.

Table 15 Comparison between Meteorological forecast vs. nonlinear regression model

Comparison between Meteorological Forecast vs. nonlinear regression model Expressed in percentage (%)				
Model 1	**Model 2**	**Out of Sample WAPE of Model 1**	**Out of Sample WAPE of Model 2**	**Percentage improvement (%)** $(\frac{WAPE_1 - WAPE_2}{WAPE_1} * 100)$
Meteorological Forecast	Artificial Neural Network	17.77047116	13.0780487	26.40%
	SVM Regression	17.77047116	13.19285875	25.75%
	K-Nearest Neighbour Algorithm	17.77047116	16.68369156	6.11%

8.2.4 What is the deviation of individual wind farms in comparison with portfolio?

Table 16 Comparison between Individual wind farm and portfolio of wind farms

	Individual vs. Portfolio error Expressed in percentage (%)	
Wind Farm	**In-sample WAPE**	**Out-of-sample WAPE**
Wind Farm 1	21.0475135	21.1549595
Wind Farm 2	23.5628343	22.8738399
Wind Farm 3	23.6078241	27.9737239
Entire Portfolio	10.8837601	13.0780487

It is observed from the Table 16 that bidding in the market with entire portfolio outperforms individual wind farms as the individual wind farms deviation is higher than the portfolio deviation between the forecasted wind power from Artificial Neural network and the actual observed wind power.

8.2.5 Conclusion

The main research question of the study is “What is the best forecasting method to improve the meteorological forecast of a portfolio of wind farms in the short term horizon”. To answer this question, four sub research questions were formulated. Findings of this study to answer those sub research question gives following important conclusion.

1. The deviation of forecasting individual wind farm productions is higher than deviation of forecasting wind production for the entire portfolio. Hence while bidding in the market, aggregating the entire wind farms into one portfolio is an effective strategy for wind power producers to reduce the penalty due to imbalance created.
2. To forecast the wind power in short term horizon, meteorological forecast of the wind power is a useful forecast data. But for a short term forecasting, persistence method or naïve predictor method is still found more accurate than the meteorological forecast.
3. The short term forecast of wind power can be improved further using hybrid approaches. Hybrid approaches make use of the meteorological forecast of wind power and statistical methods.
4. Among the hybrid approaches, for this data set, Artificial Neural network (ANN) using the meteorological forecast proved to be the approach that gives better performance than other hybrid approaches and also Naïve Predictor. ANN slightly bettered linear regression method.
5. For this data set, hybrid approaches using Support vector Machine is also performing better than both meteorological forecast and naïve predictor but slightly lower than ANN and linear regression approaches.
6. Finally, for this data set, K-Nearest Neighbour approach performed better than meteorological forecast but performed lower than all other models.

9 Bibliography

Aenlle, C. D. (2015, April 23). Renewables' Hope Springs Long-Term. *New York Times*, p. F4.

Al-Shehri, H., Al-Qarni, A., Al-Saati, L., Batoaq, A., Badukhen, H., Alrashed, S., et al. (2017). Student performance prediction using Support Vector Machine and K-Nearest Neighbor. *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, 1-5.

BigData. (2017, June 26). *Big Data Analysis*. Retrieved JUNE 22, 2016, from Neural Network: http://www.big-data.tips/neural-network

Bishop, C. M. (2006). *Pattern recognition and Machine Learning.* Springer.

Briand, L. C., Freimut, B., & Vollei, F. (2004). Using multiple adaptive regression splines to support decision making in code inspections. *Journal of Systems and Software*, 73(2), 205-217.

Brownlee, J. (2016, March 14). *Parametric and Nonparametric Machine Learning Algorithms*. Retrieved June 28, 2017, from Machine Learning Algorithms: http://machinelearningmastery.com/parametric-and-nonparametric-machine-learning-algorithms/

Campbell, D., & Campbell, S. (2008). *Introduction to regression and data analysis.* Stat Lab Workshop.

Catalao, J. P., Pousinho, H. M., & Mendes, V. M. (2009). An Artificial Neural Network Approach for Short-Term Wind Power Forecasting in Portugal. *2009 15th International Conference on Intelligent System Applications to Power Systems*, 1-5.

Cazala, J. (2017). *Architect*. Retrieved June 2017, 30, from Perceptron: https://github.com/cazala/synaptic/wiki/Architect

Chaves-Ávila, J. P., & Fernandes, C. (2015). The Spanish intraday market design: A successful solution to balance renewable generation? *Renewable Energy* , 74, 422-432.

Costa, A., Crespo, A., Navarro, J., Lizcano, G., Madsen, H., & Feitosa, E. (2008). A review on the young history of the wind power short-term prediction . *Renewable and Sustainable Energy Reviews, 12(6), 1725-1744.*, 2-3.

Denny, E., & O'Malley, M. (2006). Wind generation, power system operation, and emissions reduction. *IEEE Transactions on Power Systems (Volume: 21, Issue: 1, Feb. 2006),* 341 - 347.

Dumas, J. (2017). *K-Means Cluster Analysis of Poker Hands in Python*. Retrieved June 24, 2017, from https://jasdumas.github.io/2016-05-25-kmeans-analysis-in-python/

EPEXSPOT. (2013). *Strengthening Europe's power Market.* Paris: European Power Exchange.

Fey, B. (2010). *Grid Codes for Wind Power Integration in Spain and Germany: Use of Incentive Payments to Encourage Grid-Friendly Wind Power Plants* . Stuttgart: Fichtner.

Foley, A. M., Leahy, P. G., Marvuglia, A., & McKeogh, E. J. (2012). Current Methods and Advances in Forecasting of Wind Power Generation. *Renewable Energy, 37(1), ,* 2.

Gevrey, M., Dimopoulos, I., & Lek, S. (2003). Review and comparison of methods to study the contribution of variables in artificial neural network models. *Ecological modelling*, 160(3), 249-264.

Giannitrapani, A., Paoletti, S., Vicino, A., & Zarrilli, D. (2014). Wind power bidding in a soft penalty market. *IEEE,* 1.

Grogg, K. (2005). *Harvesting the Wind: The Physics of.* Northfield: Carleton College.

Guoyang.W., Yang.X., & Shasha.W. (2005). *Discussion about short-term forecast of wind speed on wind farm.* China: Jilin Electric Power.

GWEC. (2017). *GLOBAL WIND REPORT.* Mongolia: Global Wind Energy Council.

Hagan, M. T., & Menhaj, M. B. (1994). Training Feedforward Networks with the Marquardt Algorithm . *IEEE transactions on Neural Networks*, 5(6), 989-993.

Harvey, D. (2016, October 24). *Linear Regression and Calibration Curves*. Retrieved June 2017, 30, from Libretexts: https://chem.libretexts.org/Textbook_Maps/Analytical_Chemistry_Textbook_Maps/Map%3A_Analytical_Chemistry_2.0_(Harvey)/05_Standardizing_Analytical_Methods/5.4%3A_Linear_Regression_and_Calibration_Curves

Hedayati-Mehdiabadi, M., Zhang, J., & Hedman, K. W. (2015). Wind Power Dispatch Margin for Flexible Energy and Reserve Scheduling With Increased Wind Generation. *IEEE Transactions on Sustainable Energy (Volume: 6, Issue: 4, Oct. 2015),* 1.

Heijnen, P. (2016). Statistical Modelling reader. Delft: Delft University of Technology.

Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, 679-688.

Kuhn, M., & Johnson, K. (2013). Regression Models : Nonlinear Regression Models. In M. Kuhn, & K. Johnson, *Applied predictive modeling* (pp. 159-161). New York: Springer.

Lei, M., Shiyan, L., Chuanwen, J., Yan, Z., & Hongling, L. (2009). A review on the forecasting of wind speed and generated power. *Renewable and Sustainable Energy Reviews,Volume 13, Issue 4*, 915-920.

Liu, B. (2007). *Web data mining: exploring hyperlinks, contents, and usage data.* Berlin / Heidelberg: Springer Science & Business Media.

Mao, Y., & Shaoshuai, W. (2016). A review of wind power forecasting & prediction. *Probabilistic Methods Applied to Power Systems (PMAPS), 2016 International Conference*, 1-7.

McDonald, J. (2014). Correlation and linear regression. In J. McDonald, *Handbook of Biological Statistics (3rd ed.)* (pp. 190-208). Baltimore, Maryland: Sparky House Publishing .

Milligan, M., Schwartz, M., & Wan, Y. (2003). Statistical wind power forecasting models: Results for US wind farms. *National Renewable Energy Laboratory, Golden, CO (2003)*.

OpenCV. (2011). Retrieved June 22, 2017, from http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html

Pannell, D. J. (1997). Sensitivity analysis of normative economic models: theoretical framework and practical strategies. *Agricultural economics*, 16(2), 139-152.

Qize. (2013, December 4). *Qize – Big Data Knowledge Sharing Blog.* Retrieved June 26, 2017, from https://qizeresearch.wordpress.com/2013/12/04/a-short-example-of-multivariate-adaptive-regression-splines-mars/

Sideratos, G. &. (2007). An advanced statistical method for wind power forecasting. *IEEE Transactions on power systems*, 22(1), 258-265.

Soman, S. S., Zareipour, H., Malik, O., & Mandal, P. (2010). A Review of Wind Power and Wind Speed Forecasting Methods With Different Time Horizons . *North American Power Symposium (NAPS), 2010 (pp. 1-8). IEEE.*, 2.

Suykens, J. A., & Vandewalle, J. (1999). Least squares support vector machine classifiers. In *Neural processing letters* (pp. 9(3),293-300).

V.Vapnik. (2013). In *The nature of statistical learning theory.* Springer science & business media.

Wang, S. C. (2003). Artificial neural network. In *Interdisciplinary computing in java programming* (pp. 81-100). Springer US.

White, H., & Racine, J. (2001). Statistical inference, the bootstrap, and neural-network modeling with application to foreign exchange rates. *IEEE Transactions on Neural Networks*, 12(4), 657-673.

Yang, L., He, M., Zhang, J., & Vittal, V. (2015). Support-Vector-Machine-Enhanced Markov Model for Short-Term Wind Power Forecast. *IEEE Transactions on Sustainable Energy (Volume: 6, Issue: 3*, 1 - 5.

Zhang, W. &. (2016). Multivariate adaptive regression splines and neural network models for prediction of pile drivability. *Geoscience Frontiers*, 7(1), 45-52.

Zhou, J., Shi, J., & Li, G. (2011). Fine tuning support vector machines for short-term wind speed forecasting. *Energy Conversion and Management*, 1-8.

10 Appendix

10.1 Appendix 1 Result of the stepwise linear regression model and Durbin-Watson Test

10.1.1 Result of Stepwise linear regression model

Table 17 Result of stepwise linear regression model

Coefficients[a]							
		Unstandardized Coefficients		Standardized Coefficients		
Model		B	Std. Error	Beta	t	Sig.
1	(Constant)	.007	.001		9.395	.000
	D1_PROD_TOT	.972	.002	.972	399.846	0.000
2	(Constant)	.000	.001		.591	.554
	D1_PROD_TOT	.783	.006	.783	120.496	0.000
	EST_TOT	.210	.007	.202	31.095	.000
3	(Constant)	.003	.001		3.818	.000
	D1_PROD_TOT	.817	.006	.817	130.478	0.000
	EST_TOT	.478	.011	.459	45.432	0.000
	D2_EST_TOT	-.310	.010	-.298	-32.130	.000
4	(Constant)	.003	.001		3.763	.000
	D1_PROD_TOT	.964	.010	.964	93.333	0.000
	EST_TOT	.432	.011	.416	40.563	0.000
	D2_EST_TOT	-.228	.011	-.220	-21.694	.000
	D2_PROD_TOT	-.184	.010	-.184	-17.750	.000
5	(Constant)	.002	.001		3.192	.001
	D1_PROD_TOT	.965	.010	.965	93.665	0.000
	EST_TOT	.574	.021	.552	26.774	.000
	D2_EST_TOT	-.087	.021	-.084	-4.060	.000

Coefficients[a]						
Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
	D2_PROD_TOT	-.183	.010	-.183	-17.781	.000
	D1_EST_TOT	-.282	.037	-.272	-7.603	.000
6	(Constant)	.003	.001		3.551	.000
	D1_PROD_TOT	.962	.010	.962	93.479	0.000
	EST_TOT	.564	.021	.543	26.306	.000
	D2_EST_TOT	-.089	.021	-.085	-4.149	.000
	D2_PROD_TOT	-.185	.010	-.185	-17.959	.000
	D1_EST_TOT	-.278	.037	-.267	-7.494	.000
	EST6	.010	.002	.016	5.674	.000
7	(Constant)	.003	.001		3.990	.000
	D1_PROD_TOT	.956	.010	.956	92.993	0.000
	EST_TOT	.557	.021	.536	26.070	.000
	D2_EST_TOT	-.085	.021	-.081	-3.978	.000
	D2_PROD_TOT	-.205	.010	-.205	-19.489	.000
	D1_EST_TOT	-.273	.037	-.263	-7.402	.000
	EST6	.019	.002	.029	9.056	.000
	D2_PROD10	.017	.002	.027	8.800	.000
8	(Constant)	.002	.001		3.029	.002
	D1_PROD_TOT	.952	.010	.953	92.548	0.000
	EST_TOT	.584	.022	.562	26.248	.000
	D2_EST_TOT	-.077	.021	-.074	-3.613	.000
	D2_PROD_TOT	-.213	.011	-.213	-19.962	.000

Coefficients[a]						
Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
	D1_EST_TOT	-.281	.037	-.271	-7.614	.000
	EST6	.014	.002	.022	6.303	.000
	D2_PROD10	.024	.003	.038	9.529	.000
	EST10	-.015	.003	-.022	-4.282	.000
9	(Constant)	.002	.001		2.443	.015
	D1_PROD_TOT	.966	.011	.966	88.721	0.000
	EST_TOT	.578	.022	.556	25.925	.000
	D2_EST_TOT	-.080	.021	-.077	-3.743	.000
	D2_PROD_TOT	-.211	.011	-.211	-19.806	.000
	D1_EST_TOT	-.276	.037	-.266	-7.471	.000
	EST6	.018	.003	.028	7.312	.000
	D2_PROD10	.021	.003	.033	8.094	.000
	EST10	-.015	.003	-.022	-4.377	.000
	D1_PROD9	-.012	.003	-.017	-3.753	.000
10	(Constant)	.003	.001		4.473	.000
	D1_PROD_TOT	.985	.011	.985	88.216	0.000
	EST_TOT	.528	.023	.508	22.653	.000
	D2_EST_TOT	-.072	.021	-.069	-3.385	.001
	D2_PROD_TOT	-.208	.011	-.208	-19.529	.000
	D1_EST_TOT	-.271	.037	-.261	-7.361	.000
	EST6	.009	.003	.014	3.172	.002
	D2_PROD10	.018	.003	.029	6.864	.000
	EST10	-.008	.004	-.011	-2.130	.033
	D1_PROD9	-.031	.004	-.044	-7.545	.000
	EST8	.039	.005	.051	7.228	.000
11	(Constant)	.003	.001		4.161	.000

Coefficients[a]						
		Unstandardized Coefficients		Standardized Coefficients		
Model		B	Std. Error	Beta	t	Sig.
	D1_PROD_TOT	.981	.011	.981	87.765	0.000
	EST_TOT	.471	.027	.453	17.445	.000
	D2_EST_TOT	-.069	.021	-.066	-3.224	.001
	D2_PROD_TOT	-.205	.011	-.205	-19.226	.000
	D1_EST_TOT	-.217	.039	-.208	-5.545	.000
	EST6	.010	.003	.015	3.497	.000
	D2_PROD10	.018	.003	.028	6.647	.000
	EST10	-.007	.004	-.010	-1.893	.058
	D1_PROD9	-.028	.004	-.040	-6.701	.000
	EST8	.102	.016	.132	6.438	.000
	D1_EST8	-.067	.016	-.087	-4.199	.000
12	(Constant)	.003	.001		3.999	.000
	D1_PROD_TOT	1.002	.013	1.002	79.116	0.000
	EST_TOT	.465	.027	.447	17.216	.000
	D2_EST_TOT	-.070	.021	-.067	-3.276	.001
	D2_PROD_TOT	-.234	.013	-.234	-17.336	.000
	D1_EST_TOT	-.203	.039	-.196	-5.180	.000
	EST6	.010	.003	.015	3.458	.001
	D2_PROD10	.020	.003	.031	7.218	.000
	EST10	-.008	.004	-.012	-2.322	.020
	D1_PROD9	-.044	.006	-.064	-7.002	.000
	EST8	.110	.016	.143	6.903	.000
	D1_EST8	-.080	.016	-.104	-4.888	.000
	D2_PROD9	.022	.006	.032	3.456	.001
13	(Constant)	.003	.001		3.947	.000
	D1_PROD_TOT	1.002	.013	1.002	79.131	0.000
	EST_TOT	.460	.027	.443	17.013	.000

Coefficients[a]						
Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
	D2_EST_TOT	-.071	.021	-.068	-3.332	.001
	D2_PROD_TOT	-.221	.014	-.221	-15.627	.000
	D1_EST_TOT	-.209	.039	-.202	-5.335	.000
	EST6	.011	.003	.016	3.733	.000
	D2_PROD10	.017	.003	.027	6.159	.000
	EST10	-.007	.004	-.010	-1.778	.075
	D1_PROD9	-.043	.006	-.062	-6.796	.000
	EST8	.111	.016	.144	6.958	.000
	D1_EST8	-.070	.017	-.090	-4.169	.000
	D2_PROD9	.024	.006	.034	3.693	.000
	D2_PROD8	-.014	.005	-.021	-2.989	.003
14	(Constant)	.003	.001		3.971	.000
	D1_PROD_TOT	.971	.013	.971	72.384	0.000
	EST_TOT	.479	.027	.461	17.668	.000
	D2_EST_TOT	-.068	.021	-.065	-3.196	.001
	D2_PROD_TOT	-.200	.014	-.200	-13.871	.000
	D1_EST_TOT	-.222	.039	-.213	-5.657	.000
	EST6	.010	.003	.015	3.482	.000
	D2_PROD10	.018	.003	.029	6.540	.000
	EST10	-.007	.004	-.011	-1.995	.046
	D1_PROD9	-.050	.006	-.072	-7.852	.000
	EST8	.088	.016	.114	5.360	.000
	D1_EST8	-.058	.017	-.075	-3.448	.001
	D2_PROD9	.029	.006	.042	4.520	.000
	D2_PROD8	-.044	.007	-.065	-6.795	.000
	D1_PROD8	.044	.006	.064	6.791	.000
15	(Constant)	.003	.001		3.851	.000
	D1_PROD_TOT	.970	.013	.970	72.335	0.000

Coefficients[a]						
Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
	EST_TOT	.474	.027	.456	17.457	.000
	D2_EST_TOT	-.066	.021	-.064	-3.113	.002
	D2_PROD_TOT	-.193	.015	-.193	-13.097	.000
	D1_EST_TOT	-.225	.039	-.217	-5.744	.000
	EST6	.018	.004	.028	4.338	.000
	D2_PROD10	.017	.003	.027	5.902	.000
	EST10	-.006	.004	-.008	-1.539	.124
	D1_PROD9	-.050	.006	-.073	-7.880	.000
	EST8	.085	.016	.110	5.198	.000
	D1_EST8	-.055	.017	-.071	-3.268	.001
	D2_PROD9	.028	.006	.040	4.350	.000
	D2_PROD8	-.045	.007	-.065	-6.829	.000
	D1_PROD8	.044	.006	.065	6.876	.000
	D2_PROD6	-.008	.003	-.014	-2.677	.007
16	(Constant)	.003	.001		4.385	.000
	D1_PROD_TOT	.971	.013	.971	72.467	0.000
	EST_TOT	.461	.026	.444	17.916	.000
	D2_EST_TOT	-.068	.021	-.065	-3.205	.001
	D2_PROD_TOT	-.186	.014	-.186	-13.185	.000
	D1_EST_TOT	-.223	.039	-.214	-5.687	.000
	EST6	.020	.004	.030	4.900	.000
	D2_PROD10	.014	.002	.022	6.441	.000
	D1_PROD9	-.050	.006	-.072	-7.864	.000
	EST8	.088	.016	.114	5.426	.000
	D1_EST8	-.054	.017	-.071	-3.240	.001
	D2_PROD9	.027	.006	.039	4.192	.000
	D2_PROD8	-.046	.006	-.067	-7.029	.000
	D1_PROD8	.044	.006	.065	6.838	.000
	D2_PROD6	-.009	.003	-.015	-2.963	.003

Coefficients[a]						
Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
17	(Constant)	.003	.001		4.047	.000
	D1_PROD_TOT	.980	.014	.980	71.356	0.000
	EST_TOT	.462	.026	.445	17.976	.000
	D2_EST_TOT	-.064	.021	-.062	-3.023	.003
	D2_PROD_TOT	-.182	.014	-.182	-12.863	.000
	D1_EST_TOT	-.224	.039	-.216	-5.720	.000
	EST6	.018	.004	.028	4.528	.000
	D2_PROD10	.012	.002	.018	5.066	.000
	D1_PROD9	-.052	.006	-.075	-8.139	.000
	EST8	.089	.016	.115	5.465	.000
	D1_EST8	-.056	.017	-.072	-3.323	.001
	D2_PROD9	.026	.006	.037	3.994	.000
	D2_PROD8	-.046	.006	-.068	-7.125	.000
	D1_PROD8	.043	.006	.063	6.645	.000
	D2_PROD6	-.010	.003	-.017	-3.391	.001
	D1_PROD1	-.006	.002	-.010	-2.832	.005
18	(Constant)	.003	.001		3.950	.000
	D1_PROD_TOT	.978	.014	.978	71.256	0.000
	EST_TOT	.476	.026	.458	18.175	.000
	D2_EST_TOT	-.064	.021	-.062	-3.034	.002
	D2_PROD_TOT	-.180	.014	-.180	-12.655	.000
	D1_EST_TOT	-.239	.040	-.230	-6.035	.000
	EST6	-.011	.012	-.017	-.985	.325
	D2_PROD10	.012	.002	.018	4.996	.000
	D1_PROD9	-.052	.006	-.076	-8.154	.000
	EST8	.096	.016	.125	5.852	.000
	D1_EST8	-.064	.017	-.083	-3.742	.000
	D2_PROD9	.025	.006	.036	3.876	.000

Coefficients[a]						
		Unstandardized Coefficients		Standardized Coefficients		
Model		B	Std. Error	Beta	t	Sig.
	D2_PROD8	-.047	.006	-.068	-7.201	.000
	D1_PROD8	.044	.006	.064	6.734	.000
	D2_PROD6	-.013	.003	-.021	-3.974	.000
	D1_PROD1	-.006	.002	-.010	-2.835	.005
	D1_EST6	.032	.012	.050	2.743	.006
19	(Constant)	.003	.001		3.932	.000
	D1_PROD_TOT	.978	.014	.978	71.271	0.000
	EST_TOT	.471	.026	.453	18.315	.000
	D2_EST_TOT	-.064	.021	-.062	-3.036	.002
	D2_PROD_TOT	-.180	.014	-.180	-12.701	.000
	D1_EST_TOT	-.234	.039	-.225	-5.958	.000
	D2_PROD10	.012	.002	.018	5.038	.000
	D1_PROD9	-.052	.006	-.075	-8.141	.000
	EST8	.093	.016	.121	5.773	.000
	D1_EST8	-.061	.017	-.079	-3.631	.000
	D2_PROD9	.025	.006	.036	3.914	.000
	D2_PROD8	-.047	.006	-.068	-7.180	.000
	D1_PROD8	.043	.006	.064	6.710	.000
	D2_PROD6	-.013	.003	-.021	-3.970	.000
	D1_PROD1	-.006	.002	-.010	-2.796	.005
	D1_EST6	.022	.004	.033	5.203	.000
20	(Constant)	.003	.001		4.033	.000
	D1_PROD_TOT	.989	.014	.989	68.699	0.000
	EST_TOT	.460	.026	.442	17.579	.000
	D2_EST_TOT	-.065	.021	-.063	-3.075	.002
	D2_PROD_TOT	-.181	.014	-.181	-12.741	.000
	D1_EST_TOT	-.236	.039	-.227	-6.013	.000
	D2_PROD10	.011	.002	.017	4.749	.000

Coefficients[a]						
		Unstandardized Coefficients		Standardized Coefficients		
Model		B	Std. Error	Beta	t	Sig.
	D1_PROD9	-.054	.006	-.078	-8.377	.000
	EST8	.097	.016	.126	5.977	.000
	D1_EST8	-.059	.017	-.076	-3.498	.000
	D2_PROD9	.025	.006	.037	3.979	.000
	D2_PROD8	-.047	.006	-.069	-7.223	.000
	D1_PROD8	.041	.007	.061	6.345	.000
	D2_PROD6	-.013	.003	-.022	-4.080	.000
	D1_PROD1	-.013	.004	-.022	-3.580	.000
	D1_EST6	.023	.004	.035	5.422	.000
	EST1	.010	.004	.016	2.424	.015
21	(Constant)	.003	.001		4.051	.000
	D1_PROD_TOT	1.008	.017	1.008	58.464	0.000
	EST_TOT	.461	.026	.444	17.634	.000
	D2_EST_TOT	-.065	.021	-.062	-3.053	.002
	D2_PROD_TOT	-.170	.015	-.170	-11.209	.000
	D1_EST_TOT	-.234	.039	-.225	-5.968	.000
	D2_PROD10	.005	.004	.009	1.512	.130
	D1_PROD9	-.059	.007	-.085	-8.588	.000
	EST8	.096	.016	.124	5.917	.000
	D1_EST8	-.059	.017	-.077	-3.533	.000
	D2_PROD9	.022	.007	.032	3.381	.001
	D2_PROD8	-.049	.007	-.071	-7.432	.000
	D1_PROD8	.038	.007	.056	5.698	.000
	D2_PROD6	-.015	.003	-.025	-4.504	.000
	D1_PROD1	-.019	.005	-.031	-4.112	.000
	D1_EST6	.021	.004	.033	5.072	.000
	EST1	.009	.004	.014	2.061	.039
	D1_PROD5	-.007	.003	-.011	-2.049	.040
22	(Constant)	.003	.001		3.933	.000
	D1_PROD_TOT	1.022	.015	1.022	68.948	0.000
	EST_TOT	.462	.026	.445	17.659	.000

Coefficients[a]						
Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
	D2_EST_TOT	-.065	.021	-.062	-3.045	.002
	D2_PROD_TOT	-.160	.013	-.160	-11.821	.000
	D1_EST_TOT	-.234	.039	-.225	-5.969	.000
	D1_PROD9	-.062	.007	-.089	-9.451	.000
	EST8	.095	.016	.124	5.882	.000
	D1_EST8	-.059	.017	-.076	-3.499	.000
	D2_PROD9	.019	.006	.028	3.083	.002
	D2_PROD8	-.050	.006	-.074	-7.835	.000
	D1_PROD8	.036	.007	.053	5.498	.000
	D2_PROD6	-.017	.003	-.028	-5.303	.000
	D1_PROD1	-.023	.004	-.038	-6.137	.000
	D1_EST6	.021	.004	.032	4.923	.000
	EST1	.008	.004	.013	1.982	.048
	D1_PROD5	-.011	.002	-.017	-4.946	.000
23	(Constant)	.003	.001		3.712	.000
	D1_PROD_TOT	1.028	.015	1.028	68.317	0.000
	EST_TOT	.462	.026	.444	17.648	.000
	D2_EST_TOT	-.066	.021	-.063	-3.089	.002
	D2_PROD_TOT	-.161	.014	-.161	-11.932	.000
	D1_EST_TOT	-.234	.039	-.225	-5.965	.000
	D1_PROD9	-.054	.007	-.078	-7.390	.000
	EST8	.095	.016	.123	5.864	.000
	D1_EST8	-.058	.017	-.075	-3.419	.001
	D2_PROD9	.020	.006	.029	3.170	.002
	D2_PROD8	-.050	.006	-.073	-7.713	.000
	D1_PROD8	.037	.007	.054	5.624	.000
	D2_PROD6	-.016	.003	-.027	-5.164	.000
	D1_PROD1	-.023	.004	-.039	-6.329	.000
	D1_EST6	.020	.004	.031	4.797	.000

Coefficients[a]						
Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
	EST1	.008	.004	.014	2.009	.045
	D1_PROD5	-.011	.002	-.018	-5.199	.000
	D1_PROD3	-.013	.006	-.018	-2.224	.026
24	(Constant)	.002	.001		3.223	.001
	D1_PROD_TOT	1.038	.016	1.038	66.022	0.000
	EST_TOT	.444	.027	.427	16.313	.000
	D2_EST_TOT	-.064	.021	-.061	-2.991	.003
	D2_PROD_TOT	-.162	.014	-.162	-12.002	.000
	D1_EST_TOT	-.236	.039	-.227	-6.011	.000
	D1_PROD9	-.055	.007	-.080	-7.520	.000
	EST8	.103	.017	.133	6.215	.000
	D1_EST8	-.058	.017	-.075	-3.455	.001
	D2_PROD9	.021	.006	.030	3.307	.001
	D2_PROD8	-.050	.006	-.073	-7.735	.000
	D1_PROD8	.035	.007	.051	5.280	.000
	D2_PROD6	-.017	.003	-.028	-5.306	.000
	D1_PROD1	-.025	.004	-.042	-6.678	.000
	D1_EST6	.021	.004	.032	4.977	.000
	EST1	.013	.005	.020	2.746	.006
	D1_PROD5	-.018	.003	-.028	-5.053	.000
	D1_PROD3	-.014	.006	-.019	-2.355	.019
	EST5	.009	.004	.013	2.310	.021
a. Dependent Variable: PROD_TOT						

Where the definition of the variables is given in Section 7.1.

10.1.2 Durbin-Watson Test result

Model Summary[y]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	Durbin-Watson
1	.972[a]	.944	.944	.045204718	
2	.974[b]	.949	.949	.043067422	
3	.977[c]	.954	.954	.040903210	
4	.978[d]	.956	.956	.040243011	
5	.978[e]	.956	.956	.040123138	
6	.978[f]	.956	.956	.040057373	
7	.978[g]	.956	.956	.039897029	
8	.978[h]	.956	.956	.039860631	
9	.978[i]	.957	.956	.039833176	
10	.978[j]	.957	.957	.039726028	
11	.978[k]	.957	.957	.039691249	
12	.978[l]	.957	.957	.039668378	
13	.978[m]	.957	.957	.039651793	
14	.978[n]	.957	.957	.039557808	
15	.978[o]	.957	.957	.039544952	
16	.978[p]	.957	.957	.039547806	
17	.978[q]	.957	.957	.039533178	
18	.978[r]	.957	.957	.039519579	
19	.978[s]	.957	.957	.039519517	
20	.978[t]	.957	.957	.039509362	
21	.978[u]	.957	.957	.039502700	
22	.978[v]	.957	.957	.039505382	
23	.978[w]	.957	.957	.039497165	
24	.978[x]	.957	.957	.039488132	1.980

10.2 Appendix 2 MATLAB Code for ANN

MATLAB Code for Artificial Neural Network

```
% Training the Neural network "net"
% Finding the WAPE value of In sample and Out-of-sample
% number of neurons
n=20
% initialising error matrix with zeros
Matrix_Out= zeros(n,2);
% For loop for different number of neurons
for n=1:n
% Seeting the seed value to get the optimal model
RandStream.setGlobalStream(RandStream('mt19937ar','seed',8));
% inputing insample file into matlab workspace
filename='in.xlsx';
x=xlsread(filename);
% p is the input variables and t is the target or output variable
p=x(:,2:56); % Assignment of Input of In sample
t=x(:,1); % Assignment of Output of In sample
% initialising feedforward net with optimizing function "Levenberg-Marquardt"
net=feedforwardnet([n],'trainlm');
% Training the Neural Network with input variables 'p' and target variable't'
net=train(net,transpose(p),transpose(t));
% Using the neural networ 'net', estimated value is found for in-sample
y_est=net(transpose(p))';
% Estimation of WAPE of In sample
Perf_In= sum(abs(y_est-t))/sum(t);
% inputing out-of-sample file into matlab workspace
filename= 'out.xlsx';
a=xlsread(filename);
% d is the input variables and c is the target or output variable
d=a(:,2:45); % Assignment of Input of Out of sample
c= a(:,1); % Assignment of Output of Out of sample
% Using the neural networ 'net', estimated value is found for out-of-sample
y_out=net(transpose(d))'; %Estimated value for Out of sample
 % Estimation of WAPE of Out of sample
Perf_Out= sum(abs(y_out-c))/sum(c);
%end of neural network
% the obtained WAPE value is copied to the matrix with zeros
Matrix_Out(n, 1) = Perf_In;
Matrix_Out(n, 2) = Perf_Out;
end
```

```
% Sensitivity analysis
[row,column]=size(p);
% Assigning Matrix_Sens with zeros
Matrix_sens=zeros(row,column);
% For loop for number of inputs
for input=1:55
sum=0;
% For loop for partial derivation and Calculating Sum (total output of neural net)
for i=1:n
syms in;
X=in*A(i,input)+B(i,1);
% Mathematical representation of Neural Network
Y=tansig(X)*C(i,1);
% Partial derivative of Y w.r.t "in"
f=diff(Y,in);
%sum of all the partial derivation value with respect to one input variable
sum=sum+f ;
end
% for loop for calculating and copying the t value to Matrix Sens
for ui=1:row
in=p(ui,input);
%in = table2array(S)
% Substituting input samples into the derivative
t= vpa(subs(sum));
% converting 't' to double
u=double(t);
% Copying the value to Matrix_sens
Matrix_sens(ui, input) = u;
end
end
```

10.3 Appendix 3 R STUDIO Code for SVM

R Studio code for Support Vector Machine – Regression

```
# clearning the command window
cat("\014")
# installation of the package "e1071"
install.packages("e1071")
#loading the package "e1071"
library(e1071)
# setting the seed for running the model
set.seed(10)
# Training SVM model using the function svm()
mod<-svm(PROD_TOT ~., data = in_, kernel = "linear")
# Tuning of the model to obtain the best
tune.mod = tune(svm,PROD_TOT ~.,data = in_, kernel = "linear",ranges =
list(cost=c(0.001:1),gamma=c(0.001:1)))
summary(tune.mod)
#best model is tuned and assigned to the "best model"
bestmodel=tune.mod$best.model
summary(bestmodel)
# Prediction of In sample and Out-of-Sample values and writing the result in the .CSV format
write.csv(predict(bestmodel,in_),"SVM_in.csv")
write.csv(predict(bestmodel,out),"sVMo1.csv")
```

10.4 Appendix 4 R STUDIO Code for K-NN Algorithm

R Studio code for K-Nearest Neighbour - Regression

```
# KNN Algorithm using caret library
# clearning the command window
cat("\014")
# setting the seed for running the model
set.seed(100)
# Instaling the package "caret"
install.packages("caret")
#loading the package "caret"
library(caret)
# initializing O as the target variable, where in_ is the in sample data frame
myvars<-c("PROD_TOT")
O <- in_[myvars]
# initializing I as the input variables
myvars1<-
c("EST_TOT","D1_PROD1","D1_PROD2","D1_PROD3","D1_PROD4","D1_PROD5","D1
_PROD6","D1_PROD7","D1_PROD8","D1_PROD9","D1_PROD10","EST1","EST2","EST
3","EST4","EST5","EST6","EST7","EST8","EST9","EST10","D2_PROD1","D2_PROD2","
D2_PROD3","D2_PROD4","D2_PROD5","D2_PROD6","D2_PROD7","D2_PROD8","D2_
PROD9","D2_PROD10","D1_EST1","D1_EST2","D1_EST3","D1_EST4","D1_EST5","D1
_EST6","D1_EST7","D1_EST8","D1_EST9","D1_EST10","D2_EST1","D2_EST2","D2_E
ST3","D2_EST4","D2_EST5","D2_EST6","D2_EST7","D2_EST8","D2_EST9","D2_EST1
0", "D1_PROD_TOT", "D1_EST_TOT",  "D2_PROD_TOT",        "D2_EST_TOT")
I<-in_[myvars1]
# For running the KNN technique in R the out variable should be numeric variable
y <- as.numeric(unlist(O))
# Training of KNN using knnreg() function from the library(caret)
fit <- knnreg(I,y, k =15)
# Initializing Test as the input variables of the Out-of-sample
Test<-out[myvars1]
# Prediction of In sample and Out-of-Sample values and writing the result in the .CSV format
write.csv(predict(fit, I),"knnin15.csv")
write.csv(predict(fit, Test),"knnout15.csv")
```