**A**

**MINOR PROJECT REPORT ON**

**WIND POWER FORECASTING**

**USING MACHINE LEARNING AND STATISTICAL TOOLS**

*Submitted in partial fulfillment for the award of the degree*

*Of*

**BACHELOR OF TECHNOLOGY**

*In*

**INSTRUMENTATION AND CONTROL ENGINEERING**

*Submitted by*

Neha Gupta(03411503014) RachnaPathak(03611503014)

*Mentored by*

**Mr. Poras Khetarpal**



**DEPARTMENT OF INSTRUMENTATION AND CONTROL ENGIEERING,**

**BHARATI VIDYAPEETH’S COLLEGE OF ENGINEERING(AFFILIATED TO GURU GOBIND SINGH INDRAPRASTH UNIVERSTY,DELHI) NEW DELHI -110053**

**NOVEMBER 2017**

Table of contents

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Particulars** | **Page No.** |
| 1. | Introduction and Basic Concepts | *4* |
| 2. | Project abstract | *16* |
| 3. | Literature Review | *17* |
| 4. | Introduction to tsfresh | *25* |
| 5. | Data preprocessing | *26* |
| 6. | Project Implementation | *27* |
| 7. | Code | *34* |
| 8. | Result | *36* |
| 9. | Conclusion | *37* |
| 10. | Bibliography | *38* |

**List of Figures Page No.**

Fig 1: Wind Power Forecasting Insights 6

Fig 2: Supervised Learning 7

Fig 3: Unsupervised Learning 8

Fig 4: Semi Supervised Learing 8

Fig 5: Linear SVM Hyper plane 10

Fig 6:Non linear SVM 10

Fig 7: Logistic Regression 11

Fig 8: KNN Functioning 12

Fig 9: Machine Learning Ensemble Methods 13

Fig 10: Global Renewable capacity for power generation 15

Fig 11: Yearly wind energy production in Germany 18

Fig 12: Training and selection of an appropriate machine learning model 20

Fig 13: Example of NWP by German weather service 21

Fig 14: Deployment of NWP in WPFS Ver 1.0 with five basic modules 21

Fig 15 Support Vector Regression Illustration 23

Fig 16: Generalized model of Functional Expansion Forecaster 24

Fig 17: Spatio Temporal Clustering 24

Fig 18: Sample temporal series 25

Fig 19: Preprocessed Data 27

Fig 20: Training the model based on Adaboost Algorithm 28

Fig 21: Predicted wind velocities for next 100 points 30

Fig 22: RBFN Architecture 31

Fig 23: RBF Neuron activation values for different values of beta 32

Fig 24: Response of RBF NN on testing data 33

Fig 25: Codes for RBFN and Markov Chain Implementation 35

Fig 26: RBFN Regression 36

Fig 27: Wind velocity generated for the next 16 points 36

Fog 28: Generated potential for next 16 points 37

**INTRODUCTION**

Motivation

Since the industrial revolution in the first half of the 19th century, demand for energy to power high-tech societies and lifestyles has increased exponentially. That demand has, to date, largely been met by burning fossil fuels, a by-product of which is the emission of carbon dioxide into the atmosphere. At present, the daily release of over 100 million tones of this invisible gas goes largely unnoticed, while the lives of many have never been more comfortable thanks to the abundance of on-demand energy and derived products.

In 2011 cumulate anthropogenic CO2 emissions reached over 2000Gt, half of which has been emitted since 1970. The concentration of CO2 in the Earth’s atmosphere is increasing, and recently passed 400ppm (parts per million), well above the 1850 level of 285ppm and the estimated safe upper limit of 350ppm. Emissions of CO2 and other greenhouse gases are driving global climate change, the effects of which are beginning to be felt around the world. Quoting from the 2014 Intergovernmental Panel on Climate Change synthesis report. “Climate change will amplify existing risks and create new risks for natural and human systems. Risks are unevenly distributed and are generally greater for disadvantaged people and communities in countries at all levels of development.”

Those risks include increased frequency and duration of extreme weather events, ocean acidifi- cation, sea level rise, and increased/decreased precipitation depending on region. Large fractions of animal and plant species face extinction due to climate change. Food and water security are at risk without significant adaptation. Urban areas face increased risks to people, assets, economies and ecosystems, including risks from heat stress, storms and extreme precipitation, inland and coastal flooding, landslides, air pollution, drought, water scarcity, sea level rise and storm surges.

Rural areas are expected to xiii experience major impacts on water availability and supply, food security, infrastructure and agricultural incomes, including shifts in the production areas of food and non-food crops around the world. The need for action has never been more apparent.

In 1988 the UN and World Meteorological Organization established the Intergovernmental Panel on Climate Change to assess scientific information on all aspects of climate change and its impacts in order to formulate a realistic response.

This lead to the adoption of the Kyoto Protocol in 1997, which set various targets for developed countries to reduce emissions. However, the 2009 UN climate summit in Copenhagen failed to produce any legally binding targets for global emission control and by the end of the first phase of the Kyoto Protocol in 2012, many large emitters had failed to ratify or removed themselves from the treaty, including the US, Canada, Russia and Japan. While many developed nations now have domestic emission targets, including the US and China, hopes for global commitment to curb greenhouse gas emissions rest with the 2015 climate summit in Paris later this year. The European Union is one of the few original signatories of the Kyoto Protocol which has legally binding emissions reduction targets at present. The block has targets to reduce emissions by 20% compared to 1990 levels and to be generating 20% of electricity from renewables by 2020.

At present in the UK power generation accounts for around one quarter of greenhouse gas emissions. De-carbonising the UK power sector over the next three and a half decades will require a huge reduction in fossil fuel use and a large increase in renewable energy generation in combination with other low-carbon energy sources and energy efficiency measures.

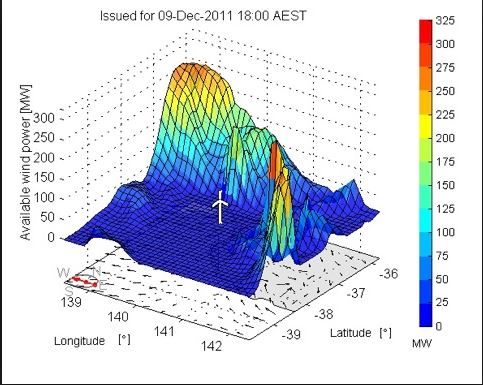


Figure1: Wind power forecasting insights

Basic Concepts

**Machine Learning**

Machine Learning is a sub field of computer science that according to Arthur Samuel in 1959, gives “computers the ability to learn without being explicitly programmed”.

Machine Learning has evolved from the study of pattern recognition and computational learning theory in artificial intelligence, and is one of the most worked upon topics today.

**Supervised learning**

In this, the input data, or the training data has a known label or result such as blink/non-blink. A model is prepared through a training process in which it is required to make predictions and is corrected when those predictions are wrong. The training process continues until the model achieves a desired level of accuracy on the training data. Examples of such problems are classification and regression. Examples of such problems are classification and regression. Example algorithms include Support Vector Machines and the Back Propagation Neural Network.

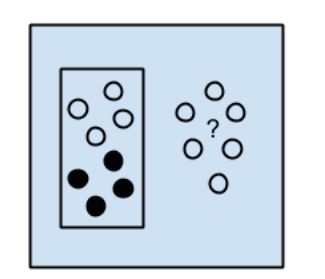


Fig 2: Supervised Learning

**Unsupervised Learning**

Unsupervised learning is the task of inferring a function to describe hidden structure from “unlabeled” data. In this, the input data is not labeled and does not have a known result. A model is prepared by deducing structures present in the input data. This may be to extract general rules. It may be through a mathematical process to systematically reduce redundancy, or it may be to organize data by similarity. Examples of such problems are clustering, dimensionality reduction, etc. Example algorithms include: k-Means.

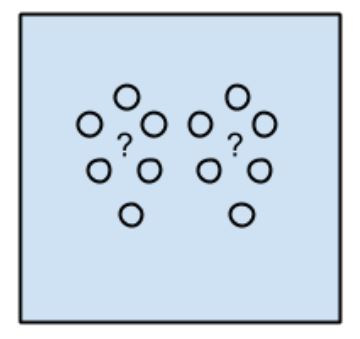


Fig3: Unsupervised learning

**Semi-Supervised Learning**

Semi-supervised learning is a class of supervised learning tasks and techniques that also makes use of unlabeled data for training – typically a small amount of labeled data. In other words, the input data is a mixture of labeled and unlabeled examples. Research has found that unlabeled data, when used in conjunction with a small amount of labeled data, can produce considerable improvement in learning accuracy. The acquisition of labeled data involves a great deal of human involvement and thus may be a costly affair. Compared to this, acquisition of unlabeled data is relatively inexpensive. But in cases where cost/budget is limited, semi-supervised learning can come in very useful and practically applicable. There is a desired prediction problem but the model must learn the structures to organize the data as well as make predictions. Examples of such problems are classification and regression. Example algorithms are extensions to other flexible methods that make assumptions about how to model the unlabeled data.

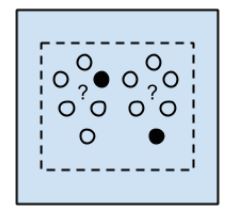


Fig 4: Semi Supervised Learning

**Linear regression**

Linear regression is an approach for modeling the relationship between a dependent variable y and one or more independent variables, also known as explanatory variables, denoted by X. The case of one explanatory variable is called simple linear regression. For more than one explanatory variable, the process is called multiple linear regression. Linear regression has various uses: If the goal is prediction, or forecasting, or error reduction, linear regression can be used to fit a predictive model to an observed data set of y and X values. After developing such a model, if an additional value of X is then given without its accompanying value of y, the fitted model can be used to make a prediction of the value of y. This approach has immense applications in developing machine learning models and solving statistical problems.

**Support Vector Machine**

In machine learning, support vector machines are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marled as belonging to one or the other categories, and SVM training algorithm builds a model that assigns any new training examples to one category or another, making it a non-probabilistic binary linear classifier.

In simple words, SVM is a supervised machine-learning model for classification or regression problems where the dataset teaches the SVM about the classes to which data belongs, so that SVM can classify any new data. It works by classifying the data into different classes: by finding a line (hyper plane), which separates the training data set into classes.

As there are many such linear hyper planes, SVM algorithm tries to maximize the distance between the various classes that are involved and this is referred as margin maximization. If the line that maximizes the distance between the classes is identified, the probability to generalize unseen / new data - is increased. In addition to performing linear classification, SVMs can also efficiently perform non-linear classifications, using what is known as the kernel trick; which is basically implicit mapping of their inputs into higher dimensional feature spaces. SVMs come in very handy, since classifying data is a very important task and also a very crucial requirement for building many machine learning systems.

SVM’s can be classified into two categories:

• **Linear SVMs** In a linear SVM, the classifier separates the training data by a hyper plane.

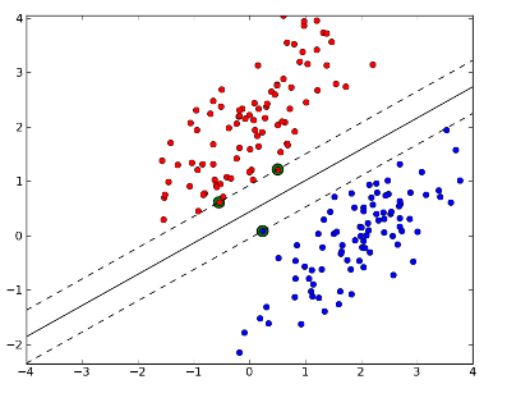


Fig 5: Linear SVM Hyper plane

The data for training is a set of points (vectors) xj along with their categories yj. For some dimension d, the xj ∊ Rd, and the yj = ±1. The equation of a hyper plane can be represented as:

f(x)=x′β+b=0 Where β ∊ Rd and b is a real number.

• **Non-Linear SVMs** In a non-linear SVM, it is not possible to separate the training data using a hyper plane. For example, if the training data for Face detection consists of group of images that are faces and another group of images that are not faces, under such conditions, the training data is very complex, hence it is impossible to find a representation for every feature vector.

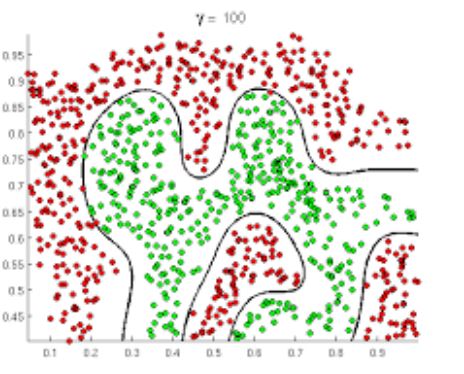


Fig 6: Non Linear SVM

**Logistic Regression**

Logistic Regression machine-learning algorithm is basically for classification tasks and not regression problems. In this, basically the dependent variable is categorical. For example, consider a case of a binary dependent variable, which can have only two values, say ‘0’ and ‘1’, which represent outcomes such as blink / non-blink, or pass / fail, etc. This algorithm applies a logistic function to a linear combination of features to predict the outcome of a dependent variable based on predictor variables.

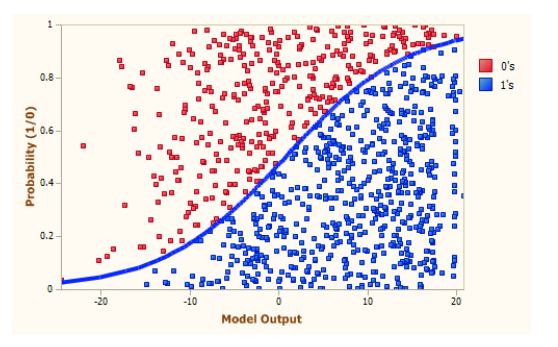


Fig 7 : Logistic Regression

Based on the nature of categorical response, logistic regression is classified into 3 types:

• **Binary Logistic Regression** – The most commonly used logistic regression when the categorical response has 2 possible outcomes i.e. either yes or not. Example: Predicting whether there is an eye blink or not.

• **Multi-nominal Logistic Regression** – Response has 3 or more possible outcomes with no particular ordering. Example: predicting what kind of search engine (Yahoo, Bing, Google, and MSN) is used by majority of Indian citizens.

• **Ordinal Logistic Regression** – Categorical response has 3 or more possible outcomes with natural ordering. Example: How a customer rates the service and quality of food at a restaurant based on a scale of 1 to 10.

**k-Nearest Neighbors:**

In pattern recognition, the k nearest neighbors algorithm is a nonparametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space.

kNN is a simple algorithm that stores all the available data and classifies new data based on a measure of similarity. k-NN has been used in statistical estimation and pattern recognition since a very long time. Algorithm: A ‘case’ is classified by a majority vote of it’s neighbors, with the case being assigned to the class most common amongst it’s K nearest neighbors, which is measured by a distance function.

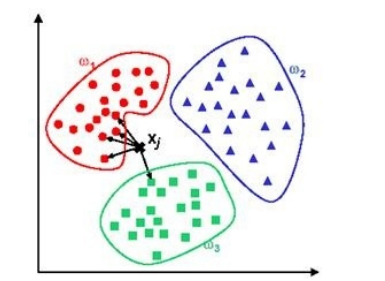


Fig 8: KNN Functioning

**Clustering**

Clustering is an unsupervised learning task. The objective is to divide a set of objects, represented by inputs {x1, x2, . . . , xn}, into a set of disjoint clusters {{x1,1, x1,2, . . . , x1,n1 }, {x2,1, x2,2, . . . , x2,n2 }, . . ., {x3,1, x3,2, . . . , x3,n3 }}, that contain objects similar to each other in some sense. Typically, similarity between two objects is defined by Euclidean distance or Manhattan distance.

**Ensemble methods**

**Ensemble** methods use multiple models to combine them into one stronger model. Experience shows that it is common for individual algorithms to be outperformed by combinations ofmodels and heterogeneous combinations are of special interest ([BelKor07]). As mentioned by [Die00] there are essentially three reasons why ensembles of models perform better than individual models: statistical, computational and representational.

**Statistical**

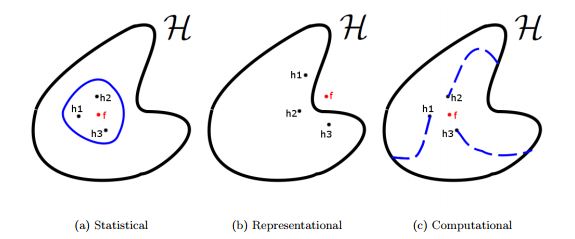
One way to look at a learning algorithm is to view it as searching a hypothesis space H to find the best one. Without enough data the algorithm may find a few different hypotheses in H that give the same accuracy on the training (or validation) data. By constructing an ensemble, the algorithm can find a point that is, in a certain sense, an average of the members of the ensemble and thus reduces a risk of choosing the wrong classifier.

**Representational**

In many cases the true function f cannot be represented within hypothesis space H. By combining classifiers into an ensemble it may be possible to expand a set of representable functions. Even though some algorithms, like neural networks (that actually are universal approximators as shown in [Hor91]), can, in theory, express a lot of functions, it is important to bear in mind that due to the finite amount of training data, they will effectively explore a finite number of hypotheses and will stop when the model fits the training data well enough.

**Computational**

Even when there is enough data, an algorithm performing local search for the best hypothesis may get stuck in local optima. That is the case for neural networks for example. Therefore, starting from different points and combining obtained models in an ensemble may lead to a model that is closer to the true hypothesis.

Fig9:Machine Learning Ensemble Methods

**The Power System**

Many of the world’s power systems were developed over the past century: initially to power electric lighting, then to transmit electricity from a few large power stations to individual cities or industrial complexes, and later becoming increasingly interconnected eventually providing a reliable power supply on national and even continental scales. Unlike resources that can be easily stored, electricity supply must meet demand in real time. If more power is produced than consumed, the frequency of the AC power system increases, and vice versa. Even a small change in frequency is enough to damage synchronous machines and other equipment. Other limits on voltage, line and transformer capacity, reactive power and phase angle are imposed for similar reasons. Modern power systems are highly controlled and include protection systems to maintain safe operation and protect equipment.

Today, power system operators act in conjunction with electricity markets to provide secure and economic supply. Electricity networks form natural monopolies which were traditionally operated by vertically integrated public companies that generated, transmitted and distributed electricity to consumers. However, the liberalization of electricity markets, beginning in the UK in the 1980s, has seen the vertical disintegration and privatisation of the electricity industry, and the creation of new markets for energy, ancillary services and capacity.

During the same period, the mix of generation technologies began to change. Electricity generation has been dominated by large synchronous machines, driven by thermal power stations since the early 20th century, and power systems have been designed to accommodate them: high voltage transmission systems carry power form large power stations to load centres, where it is distributed to customers at a lower voltage. However, the beginning of the 21st century has seen the rapid growth of renewable electricity generation in developed countries motivated by the threat of climate change

Weather dependent renewable generation such as wind and solar power are variable and often spread over large geographical areas, connecting to power systems at the distribution level. The rise in so-called distributed generation poses a challenge to power systems that were built for large synchronous machines connected close to load centres.

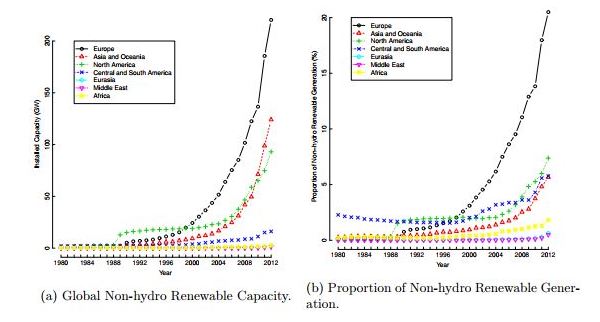


Figure 10:Global Renewable capacity for power generation

Electricity markets and power system operators are having to adapt to ever increasing penetration of variable generation, and one key component of that transition is forecasting variable generation.

**Wind Power Forecasting**

Given that the wind, and therefore the power generated by wind turbines, is variable, and that electricity supply must meet demand in real time, the need for wind power forecasts is clear [10]. Furthermore, since the day-to-day running of today’s liberalised energy industry is marketised, all participants in the industry are exposed to the effects of increasing the penetration of variable generation. With financial penalties for over- /under-delivering on generation, and repercussions for electricity price and balancing costs, forecasting is critical to economic operations, as well as technical . Wind farm developers and operators also have an interest in future production to minimise lost energy capture when performing maintenance, protect assets against extreme weather events and identify locations with an abundant wind resource.

**PROJECT ABSTRACT**

Wind is one of the important renewable energy resources. It is also a key component in the distributed energy generation for the future smart grids. Depleting fossil fuel resources is a major challenge faced

by people all over the world. Consequently, there is a major thrust on investigating and developing clean and renewable energy sources such as solar, wind, hydro, geothermal and biofuels. Wind energy is emerging as a viable solution to the world owing to its wide availability and ease of conversion.

Substitution of significant part of present power generation by conventional sources by wind energy is is being planned by nations and is being implemented in progressive stages.

**Accurate forecasting of wind power generation is quite an important as well as challenging task for the system operators and market participants due to its high uncertainty**.

Itis essential to quantify uncertainties associated with wind powergeneration forecasts for their efficient application in optimalmanagement of wind farms and integration into power systems.

We are going to propose a model using statistical tools and Machine Learning techniques that would be used for **short term forecasting of wind power generation**, which is a fundamental requirement for market operators and other concerned entities to mitigate the problems like power balance, system frequency maintenance, power quality issues, scheduling and planning complexities and efficiently utilize this clean and freely available source of energy.

The main drawback with the existing statistical models is that they require a historical dataset for forecasting. This is a hurdle for newly started farms due to the lack of historical data set. This model will be composed of self-adaptive Artificial Neural Networks to effectively forecast the wind power without using the historical data. The algorithms will converge after few days of operation. Neural network models such as Multi-Layer Perceptron (MLP), Radial Basis Functions (RBF), and Support Vector Machine (SVM) etc. have better prediction capabilities at the expense of high computational complexity due to the presence of hidden layer.

**LITERATURE REVIEW**

**Power Systems**

To optimally utilise variable renewable generation, such as wind power, power systems and the way they are operated are changing: transmission networks must connect distant renewable generation to load centres, distribution networks must accommodate small and medium scale generation, and operators must consider the stochastic nature of this new variable generation when performing scheduling tasks. Many decisions relating to power system operation are increasingly informed by forecasts on a variety of temporal and spatial scales, and the upper limit on the level of variable generation that can be accommodated by a given power system will ultimately be set by the skill of these forecasts, and their users. A recent survey of US power system operators identifies the growing importance of forecasting for reliable grid operations, with one of the key findings being that “wind power forecast[ing] is the most important pre-requisite for successfully integrating wind energy into power systems”. More specifically, integrating wind power and maintaining security of supply requires careful management of transmission constraints and scheduling of conventional generation, which to be done most efficiently, requires accurate intra-day and day-ahead forecasts. Furthermore, it is well established that moving to a probabilistic approach is of as much benefit as moving from naive to advanced point forecasting

**Electricity Markets**

Electricity markets were designed for dealing with mainly dispatchable generation and fairly predictable demand allowing for extensive forward contracting accompanied by a real-time mechanism to facilitate power system operation. As recently as a decade ago their future evolution in many developed countries was expected to remain in this paradigm, as demonstrated by a 2005 paper describing US electricity markets and their future evolution that includes no mention of the potential role of renewable generation . Meanwhile, Denmark was learning how to operate liberalised power markets with high volumes of wind power, occasionally approaching 100% instantaneous penetration . Today, with many governments committed to reducing CO2 emissions, electricity markets in developed countries are increasingly having to operate with, and plan for, high renewable energy penetration. Participants in existing markets rely on forecasts to make optimal trading decisions, while new market structures are being proposed to address some of the failings of markets designed for conventional generation.

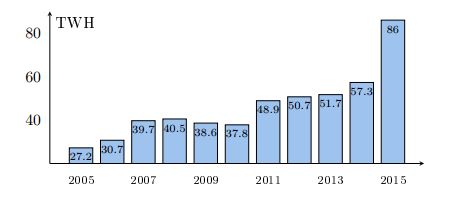


Figure11: Yearly wind energy production in Germany

**Operations and Maintenance**

Finally, maintenance costs contribute a significant portion to the cost of energy over the lifetime of a wind farm. Onshore, operations and maintenance (O&M) costs typically make up around 5% of the cost of energy, whereas offshore the figure can be much higher, from 20% to 30% or more, depending on the distance of the farm from shore. Wind forecasts allow non-essential maintenance to be scheduled to minimize lost energy capture onshore, and are essential for scheduling maintenance offshore where safety constraints on vessel operation and crew transfer are very restrictive.

**Objective**

It is the objective of this research to develop new prediction techniques for application to short- and wind power forecasting. Forecasts on this time scale are typically made using recent measurements as an input to a statistical model. Numerous such models are described in the literature, each with its own merits. However, spatial techniques, where measurements made at multiple locations are used as inputs, are underdeveloped and have many attractive benefits. Capturing spatial correlation has been shown to improve forecast skill in small scale studies.

Spatial models may also be built to directly forecast power production removing the need to model wind farm power curves. This is investigated in conjunction with a method for producing very-short-term forecasts with a much higher spatial dimension, a problem facing power systems operators with very high wind penetration.

**State-of-the-Art in Wind Power Forecasting**

The history of wind power forecasting can be traced back to the late 1970s when it was identified as a key requirement for operating large scale wind power plants . A good example of early work is by Brown et al. who used wind speed forecasts and a wind turbine power curve to produce forecasts, published in 1984∗ . Brown identified the need to understand how wind might contribute to future ‘multisource’ energy networks, and recognised at this early stage that “once a wind power generator is supplying power to an energy system, a method of forecasting wind power a few hours in advance is required to ensure efficient utilization of the power.” Over the following 30 years research activity in this area has expanded, most significantly since the early 2000s, as wind power has been adopted around the world.

Wind power forecasting is regarded as a high priority research area that is expected to reduce energy and power system running costs, and improve power system reliability. The International Energy Agency highlights advances in forecasting in its 2013 technology roadmap using Spain as an example where forecast errors from 1 to 48 hours ahead have reduced significantly between 2008 and 2013. However, it goes on to stress the importance of further research and development (R&D) in short-term forecasting saying: “Improving the accuracy of short-term wind forecast is needed for the operation of wind power plants, especially for electricity markets and the power system.”

As a result, both academic and commercial institutions are investing in forecasting R&D and the state-of-the-art is advancing rapidly. Energy forecasting more generally has grown into a broad and fast moving research area featured in many international conferences and publications. In 2012, point wind power and load forecasting challenges comprised the first Global Energy Forecasting Competition , and attracted a large number of entries. The 2014 competition expanded to include solar power and electricity price forecasting, and probabilistic forecasts.

The European Commission funded research project ANEMOS.plus—“Advanced Tools for the Management of Electricity Grids with Large-Scale Wind Generation” in partnership with the SafeWind project produced a broad literature review of the state-of-the-art in short term prediction of wind power in 2011, containing some 386 references, which serves as a starting point for this review.

**Prediction techniques** can be divided in two groups, forecasting based on numerical weather prediction (NWP) on the one hand and predictions based on the historical time series on the other hand [22,29]. In both categories a wide variety of different techniques and hybrid approaches can be found. A review of methods for wind power forecasting is given by Foley. A review for forecasting of both wind speed and generated power was written by Lei et al. [73]. Soman et al. [102] give a comprehensive overview of techniques with emphasis on the use for different horizons. Wang et al. [49] provide a classification of various wind power forecasting methods.

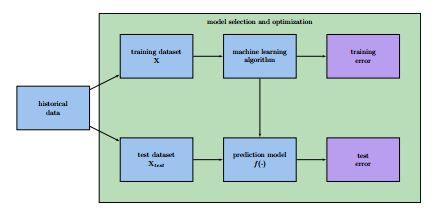


Figure12: Training and selection of an appropriate machine learning model

**Numerical Weather Prediction**

NWP models are based on physical computations describing the state of the atmosphere, including values of radiation, turbulence, and pressure. Besides the laws of physics, typically Navier-Stokes-Equations are employed, which are used to describe the motion of viscous liquids. Weather prediction can be addressed with global or regional models of different resolution. For dealing with larger resolution and better representations of the atmospheric processes, NWP models are usually computed using supercomputers at the weather services or in research institutes. The forecasts are not produced for one particular purpose, but rather for different use cases in industry and science. The output of the model is not only wind speed but the state of the atmosphere for a given place and time, resulting in a coarse grid of forecasts. Based on the grid, with distance between the grid points of few kilometers, the wind speed at the location of the target turbine is computed. The speed value is transformed into a power value by applying the power curve of the turbine.

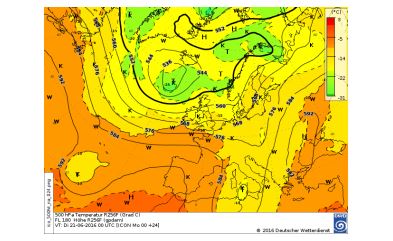
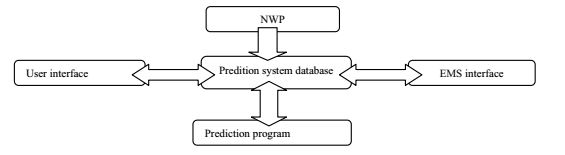


Fig13: Example for NWP by the German Weather Service

Figure14:Deployment of NWP in WPFS Ver 1.0

Although these models are widely applied, there are some drawbacks. The consumer of the forecasts is dependent on the weather services providing the forecasts. The time scales that can be used are always fixed and the forecasts are only available at a certain point of time. Because of the chaotic nature of the atmosphere, it is a very challenging task to give a good prediction with a physical model and therefore, for short forecast horizons other approaches like statistical learning yield superior results. Besides these approaches, computational fluid dynamics (CFD) gained attention in the recent past.

In the field of numerical weather forecasts, it is common to use ensemble models

On the one hand, forecasts can be improved. On the other hand, not only a deterministic prediction is desired but rather an prediction interval with uncertainty information, which is obtained by probabilistic methods. Several NWP models are initialized with different values and combined to an ensemble forecast. In the field of NWP models, this approach is especially reasonable because of the difficulty to obtain the accurate values of the current atmospheric state. Further, small changes in the initial states yield large deviations in the forecast outcome. The forecasting using ensembles was one of the most important novelties of the recent years in the field of NWP. The weather services offer ensemble forecasts and their use is considered as state of the art.

**Statistical Learning**

Statistical learning refers to a vast set of tools for understanding data. These tools can be classified as supervised or unsupervised. Broadly speaking, supervised statistical learning involves building a statistical model for predicting, or estimating, an output based on one or more inputs. It has been shown that machine learning methods are well-suited to the domain of wind speed and wind power prediction. Especially when spatiotemporal information is available machine learning models can yield feasible prediction performance.

**Support Vector Techniques**

The SVM technique is very successful for classification and regression tasks. In particular, good generalization abilities can be achieved. For short-term wind power prediction, Kramer and Gieseke successfully applied the SVR algorithm. Here, a loss function parameter study is conducted State of the Art and analyses of the prediction on both grid point and park levels suggest that SVR is very well-suited when -loss6 is employed. Mohandes employ support vector machines for wind speed prediction, too. The performance of the SVM prediction is compared to the multilayer perceptron (MLP) neural networks. The SVM model outperforms the MLP in most cases. However, the experiments are only conducted on mean daily wind speed data from Madina city, Saudi Arabia and give no insights for short-term prediction horizons. Another SVR approach to short-term wind power forecasting is given by Zhang. They describe a framework based on grid search and a multi scale SVR. A comparison with an MLP demonstrates that the SVR approach is “robust, precise, and effective”. Salcedo-Sanz employ SVR for the reconstruction of wind speed measurements from neighboring turbines. In the majority of cases, the SVR model works better than an MLP.

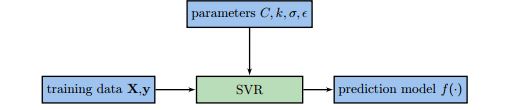


Figure15: Support Vector Regression Process Illustration

**Artificial Neural Networks**

Wind speed forecasting tasks can be handled by using back-propagation neural networks. Different variants of backpropagation and in particular Resilient Propagation are compared by Stubbemann .

Artificial Neural Networks (ANN) are good candidate models for establishing a non-linear relationship between inputs and outputs for which no perfect mathematical model is available. Neural network models such as Multi-Layer Perceptron (MLP), Radial Basis Functions (RBF) and Support Vector Machine (SVM) etc. have better prediction capabilities at the expense of high computational complexity due to the presence of hidden layer. To reduce this complexity, the authors used models like, Functional Link ANN (FLANN), Legendre Neural Network (LeNN) and Chebyshev Neural Network (ChNN) which do not contain any hidden layer. In general, these models are single-layer ANN possessing higher rate of

convergence and lesser computational load than those of a Multi Layered Perceptron structure.

All the models mentioned in this paper receive the wind velocity (m/s) and i rection for three consecutive hours and predict the wind power for the next hour. During the next hour, when actual wind power value is measured from the wind farm and compared against the predicted value. The error value is used to train the models to achieve accurate prediction.

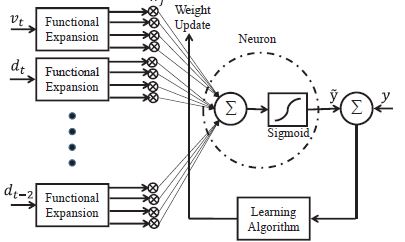


Figure 16: Generalized model of Functional Expansion Forecaster

**Spatio-Temporal Techniques**

In contrast to numerical weather predictions, machine learning methods usually only make use of the time series data itself, i.e., power or speed measurements. A training dataset consists of historical measurements. When performing a forecast, the objective is to predict the measurement after a forecast horizon ∆t., e.g., in half an hour. The input patterns consist of µ past time steps, which is called the feature window. It can be seen that nearby turbines show similar speeds at the same time and it exists some correlation between the time series of them. If one wants to give a power output prediction for a certain turbine based on its past time series measurements used as patterns, including the measurements of turbines in the vicinity of a few kilometers into the patterns greatly helps to reduce the prediction error.

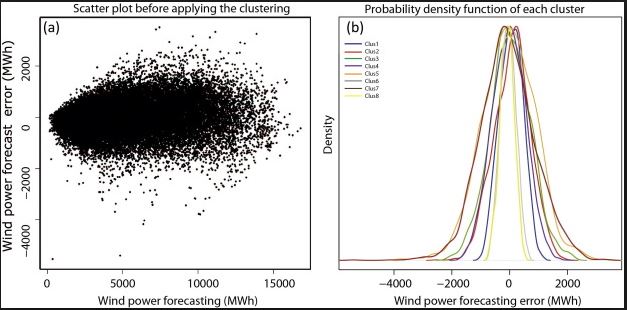
****

Figure17: Spatio Temporal Clustering

**INTRODUCTION TO tsfresh**

To preprocess the data, we have used a python package known as tsfresh. It automatically calculates a large number of time series characteristics, the so called features. Further the package contains methods to evaluate the explaining power and importance of such characteristics for regression or classification tasks.

**Concept**

tsfresh is used to to extract characteristics from time series. Let’s assume you recorded the ambient temperature around your computer over one day as the following time series:

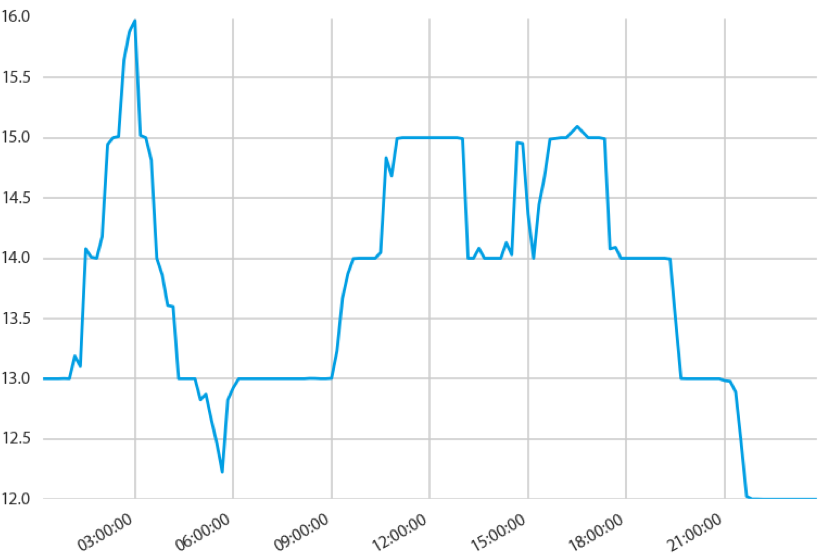
****

Figure18: Sample Temporal series

**DATA PREPROCESSING**

**Feature Calculation**

tsfresh enforces a strict naming of the created features, which you have to follow whenever you create new feature calculators. This is due to the **[tsfresh.feature\_extraction.settings.from\_columns()](http://tsfresh.readthedocs.io/en/latest/api/tsfresh.feature_extraction.html" \l "tsfresh.feature_extraction.settings.from_columns" \o "tsfresh.feature_extraction.settings.from_columns)**method which needs to deduce the following information from the feature name

* the time series that was used to calculate the feature
* the feature calculator method that was used to derive the feature
* all parameters that have been used to calculate the feature (optional)

Hence, to enable the **[tsfresh.feature\_extraction.settings.from\_columns()](http://tsfresh.readthedocs.io/en/latest/api/tsfresh.feature_extraction.html" \l "tsfresh.feature_extraction.settings.from_columns" \o "tsfresh.feature_extraction.settings.from_columns)** to deduce all the necessary conditions, the features will be named in the following format

{time\_series\_name}\_\_{feature\_name}\_\_{parameter name 1}\_{parameter value 1}\_\_[..]\_\_{parameter name k}\_{parameter value k}

(Here we assumed that {feature\_name} has k parameters).

**Time Series Forecasting**

Features that are extracted with *tsfresh* can be used for many different tasks, such as time series classification, compression or forecasting. This section explains how one can use the features for time series forecasting tasks.

The “sort” column of a DataFrame in the supported [Data Formats](http://tsfresh.readthedocs.io/en/latest/text/data_formats.html#data-formats-label) gives a sequential state to the individual measurements. In the case of time series this can be the *time* dimension while in the case of spectra the order is given by the *wavelength* or *frequency* dimensions. We can exploit this sequence to generate more input data out of a single time series, by *rolling* over the data.

**PROJECT IMPLEMENTATION**

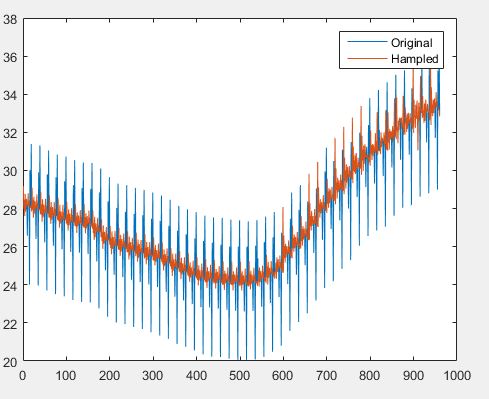
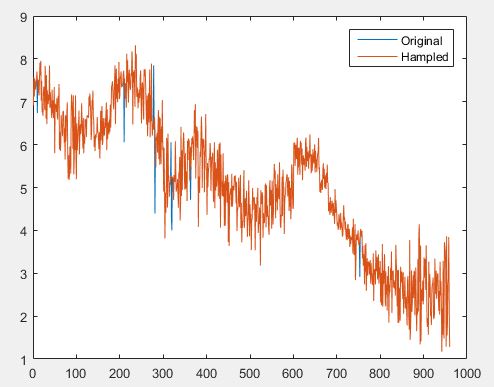
** **

Figure 19: Preprocessed Data

The data was acquired from Karanataka Grenko Power Plant. It cosistes the values of temperature, wind velocity, wind’s direction amd corresponding voltage generated by the turbines. Data points are 15 minutes apart. The data was found to have outliers and missing values. It was treated for these anamolies using MATLAB.

Using TSfresh. more than thousand features were extracted Taking these features as base, Adaboost algorithm was applied and model was trained for predicting future values.

**ADABOOST ALGORITHM**

**AdaBoost**, short for "Adaptive [Boosting](https://en.wikipedia.org/wiki/Boosting_(meta-algorithm))", is a [machine learning](https://en.wikipedia.org/wiki/Machine_learning) [meta-algorithm](https://en.wikipedia.org/wiki/Meta-algorithm) formulated by [Yoav Freund](https://en.wikipedia.org/wiki/Yoav_Freund" \o "Yoav Freund) and [Robert Schapire](https://en.wikipedia.org/wiki/Robert_Schapire) who won the [Gödel Prize](https://en.wikipedia.org/wiki/G%C3%B6del_Prize) in 2003 for their work. It can be used in conjunction with many other types of learning algorithms to improve their performance. The output of the other learning algorithms ('weak learners') is combined into a weighted sum that represents the final output of the boosted classifier. AdaBoost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers. AdaBoost is sensitive to noisy data and [outliers](https://en.wikipedia.org/wiki/Outlier). In some problems it can be less susceptible to the [overfitting](https://en.wikipedia.org/wiki/Overfitting_(machine_learning)" \o "Overfitting (machine learning)) problem than other learning algorithms. The individual learners can be weak, but as long as the performance of each one is slightly better than random guessing (e.g., their error rate is smaller than 0.5 for binary classification), the final model can be proven to converge to a strong learner.

Every learning algorithm will tend to suit some problem types better than others, and will typically have many different parameters and configurations to be adjusted before achieving optimal performance on a dataset, AdaBoost (with [decision trees](https://en.wikipedia.org/wiki/Decision_tree_learning) as the weak learners) is often referred to as the best out-of-the-box classifier.[[1]](https://en.wikipedia.org/wiki/AdaBoost#cite_note-1)[[2]](https://en.wikipedia.org/wiki/AdaBoost#cite_note-2) When used with decision tree learning, information gathered at each stage of the AdaBoost algorithm about the relative 'hardness' of each training sample is fed into the tree growing algorithm such that later trees tend to focus on harder-to-classify examples.

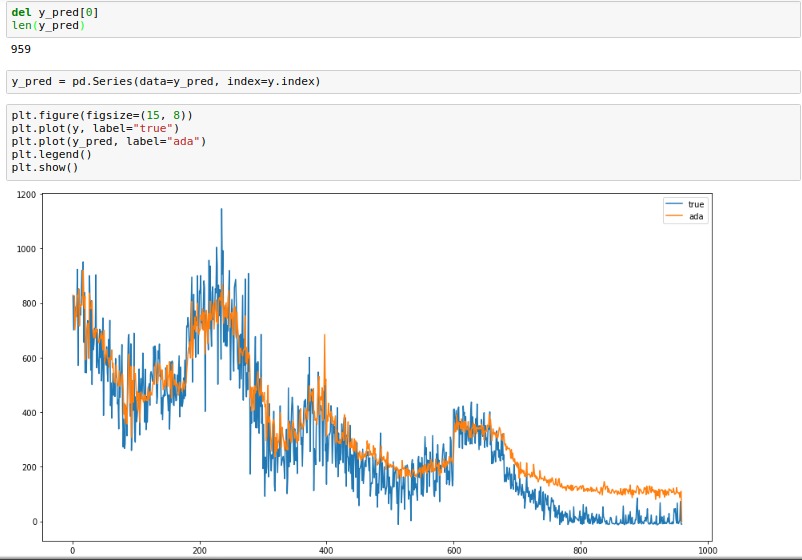


Figure 20: Training the model based on Adaboost algorithm

**MARKOV CHAINS**

The determination of important statistical models for wind speed time series at different time periods is a huge point of interest for the wind power industry, in particular for optimal control of the wind turbines. Also important topics are: establishing of a sending schedule or programming the wind energy, designing and evaluating wind energy and so on.

Time series data for wind speed can be obtained from different sources, such as weather forecasting stations or from wind turbines itself that are equipped with anemometers.

Time series data set of wind speed may be noted as 𝑣𝑡,𝑡=0,1,2,⋯,𝑇 where 𝑣𝑡 is the wind speed (discreet) at the time 𝑡, 𝑡 starting at 0 and ending in 𝑇. The total number of discrete values of the wind speed for a time series is 𝑇+1.

Wind speed measurements are continuous values, but to be able to fit into an application of Markov chains, 𝑣𝑡 is discreet and can take a finite number of possible states.

Let us suppose that continuous wind speeds can be discretized in 𝑟 states, 𝑆𝑖,=0,1,2,⋯,𝑟. For example, 𝑆1 state can be assigned for values of wind speed between 0 and 1 m/s.

The method in which wind speeds are divided in discreet values of 𝑟 states, is highly dependent on the target application and other factors such as changes in wind speed or specifications of the turbine. In the literature often is used to make this division the range of speeds 𝑣𝑐𝑢𝑡−𝑖𝑛 and 𝑣𝑟𝑎𝑡𝑒𝑑.

The reason for this range is as follows: a turbine generates zero power as long as wind speed is between 0 and the cut-in speed. When the wind speed is higher than the rated speed, then the turbine will generate a set power (relatively constant), unless the wind is too strong and the wind speed exceeds the cut-out speed, moment that is equivalent with the turbine shutdown in order to protect it from potential damage.

Based on the startup speed and the normal speed of the wind we could define the state 𝑆1 as being wind speeds that range from 0 to 𝑣𝑐𝑢𝑡−𝑖𝑛, meaning 0≤𝑣≤𝑣𝑐𝑢𝑡−𝑖𝑛. 𝑆𝑟 could be defined as a state comprising wind speed values greater or equal to 𝑣𝑟𝑎𝑡𝑒𝑑. Thus 𝑆2 and 𝑆𝑟−1 could be defined based on different classification schemes of wind speed.

A simple way would be addressing speed range, Δ𝑣. Let Δ𝑣=1 𝑚/𝑠 be a range, then 𝑆2 is the state containing the following values 𝑣𝑐𝑢𝑡−𝑖𝑛≤𝑣≤𝑣𝑐𝑢𝑡−𝑖𝑛+1. The states from 𝑆3 to 𝑆𝑟−1 could be defined in the same manner. In this case, the number of possible states is given by the interval Δ𝑣, 𝑣𝑐𝑢𝑡−𝑖𝑛 and 𝑣𝑟𝑎𝑡𝑒𝑑.

**Procedure to simulate wind speed**

The first step is to calculate the one step transition matrix. This matrix is a Markov chain so the sum of the probabilities of a single line (of the matrix) is equal to 1. Thus, an initial state is set. The first state, of no wind, 𝑆0 can be considered. Using a random uniform number, the following state of the wind speed can be determined. If 𝑆0 is obtained, then first we check if the wind speed is 0. If the wind speed is not 0, then a random number from the interval of states 𝑆0 is used in order to generate a value. If the highest state is found then a distribution of a gamma parameter is used to calculate the wind speed. For intermediate states, a value is generated of a uniform

The states of the Markov chain are defined as:

 state 1: 𝑣∈(0,1]𝑚/𝑠;

 state 2: 𝑣∈(1,2]𝑚/𝑠;

 state 3: 𝑣∈(2,3]𝑚/𝑠;

 state 4: 𝑣∈(3,4]𝑚/𝑠;

 state 5: 𝑣∈(4 5]𝑚/𝑠;

 state 6: 𝑣∈(5,6]𝑚/𝑠;

 state 7: 𝑣∈(6,7]𝑚/𝑠;

 state 8: 𝑣∈(7,8]𝑚/𝑠;

 state 9: 𝑣∈(8,9]𝑚/𝑠;

The lowest recorded speed: 1.169000

The highest recorded speed: 8.319000

Mean speed: 5.061765

The estimated matrix of transition of order 1 is:

0 1.0000 0 0 0 0 0

0.2000 0.6000 0.2000 0 0 0 0

0 0.4000 0.4000 0.2000 0 0 0

0 0 0.0714 0.7143 0.1429 0.0714 0

0 0 0.1250 0.2500 0.2500 0.3750 0

0 0 0 0.1111 0.4444 0.3333 0.1111

0 0 0 0 0 0.2500 0.7500

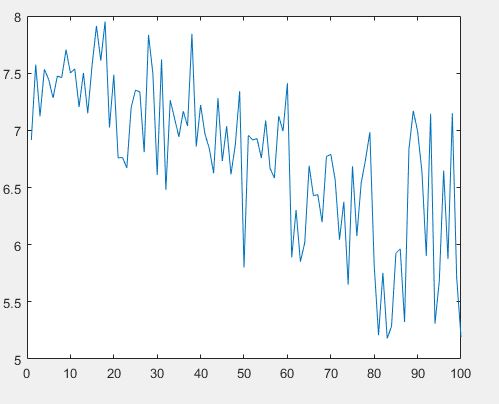
****

FIGURE21: Predicted wind velocities for next 100 data points

**RADIAL BASIS FUNCTION NEURAL NETWORK**

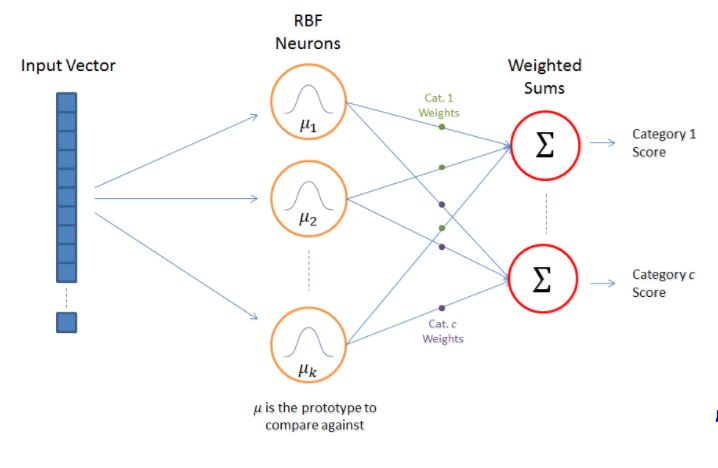
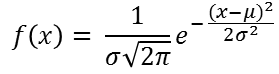
Each neuron in an MLP takes the weighted sum of its input values. That is, each input value is multiplied by a coefficient, and the results are all summed together. A single MLP neuron is a simple linear classifier, but complex non-linear classifiers can be built by combining these neurons into a network. RBFN approach is more intuitive than the MLP. An RBFN performs classification by measuring the input’s similarity to examples from the training set. Each RBFN neuron stores a “prototype”, which is just one of the examples from the training set. When we want to classify a new input, each neuron computes the Euclidean distance between the input and its prototype.****

Figure 22: RBFN Architecture

RBF Neuron Activation Function

Each RBF neuron computes a measure of the similarity between the input and its prototype vector (taken from the training set). Input vectors which are more similar to the prototype return a result closer to 1. There are different possible choices of similarity functions, but the most popular is based on the Gaussian. Below is the equation for a Gaussian with a one-dimensional input.

[](http://chrisjmccormick.files.wordpress.com/2013/08/gaussian.png)

Where x is the input, mu is the mean, and sigma is the standard deviation. This produces the familiar bell curve shown below, which is centered at the mean, mu (in the below plot the mean is 5 and sigma is 1).

The RBF neuron activation function is slightly different, and is typically written as:

[Activation_Equation](http://chrisjmccormick.files.wordpress.com/2013/08/activation_equation.png)

In the Gaussian distribution, mu refers to the mean of the distribution. Here, it is the prototype vector which is at the center of the bell curve.

For the activation function, phi, we aren’t directly interested in the value of the standard deviation, sigma, so we make a couple simplifying modifications.

The first change is that we’ve removed the outer coefficient, 1 / (sigma \* sqrt(2 \* pi)). This term normally controls the height of the Gaussian. Here, though, it is redundant with the weights applied by the output nodes. During training, the output nodes will *learn* the correct coefficient or “weight” to apply to the neuron’s response.

The second change is that we’ve replaced the inner coefficient, 1 / (2 \* sigma^2), with a single parameter ‘beta’. This beta coefficient controls the width of the bell curve. Again, in this context, we don’t care about the value of sigma, we just care that there’s some coefficient which is controlling the width of the bell curve. So we simplify the equation by replacing the term with a single variable.

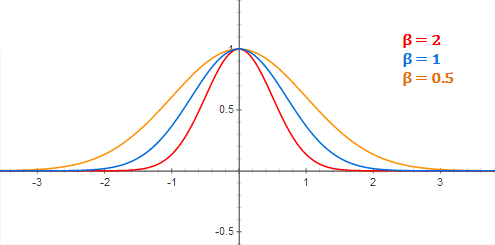
[](http://chrisjmccormick.files.wordpress.com/2013/08/diff_variances_plot.png)

Figure 23:RBF Neuron activation for different values of beta

There is also a slight change in notation here when we apply the equation to n-dimensional vectors. The double bar notation in the activation equation indicates that we are taking the Euclidean distance between x and mu, and squaring the result. For the 1-dimensional Gaussian, this simplifies to just (x - mu)^2.

It’s important to note that the underlying metric here for evaluating the similarity between an input vector and a prototype is the Euclidean distance between the two vectors.

Also, each RBF neuron will produce its largest response when the input is equal to the prototype vector. This allows to take it as a measure of similarity, and sum the results from all of the RBF neurons.

As we move out from the prototype vector, the response falls off exponentially. Recall from the RBFN architecture illustration that the output node for each category takes the weighted sum of *every* RBF neuron in the network–in other words, every neuron in the network will have some influence over the classification decision. The exponential fall off of the activation function, however, means that the neurons whose prototypes are far from the input vector will actually contribute very little to the result.

We trained the RBF Neural Network on the whole dataset and a regression was performed between velocity and potential generated values. Consequently a relationship was found between them.

The new values of wind velocity were then fed to the trained model and potential generated was forecasted.

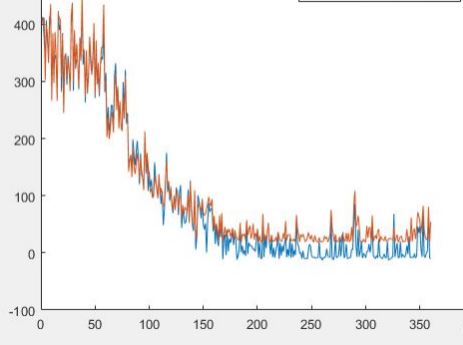
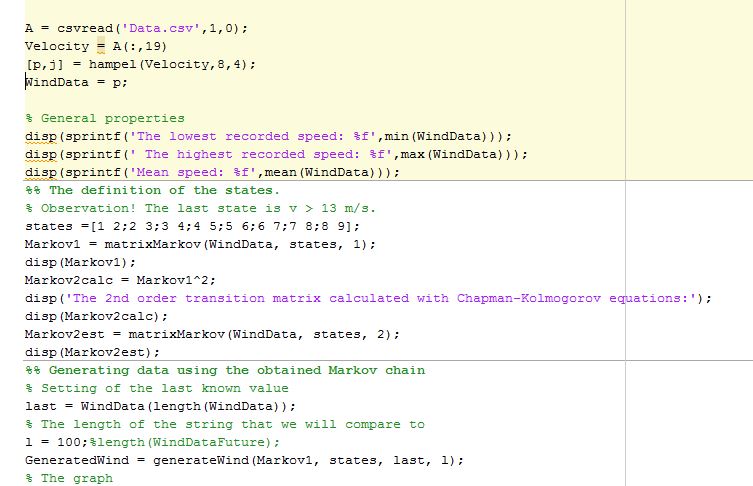
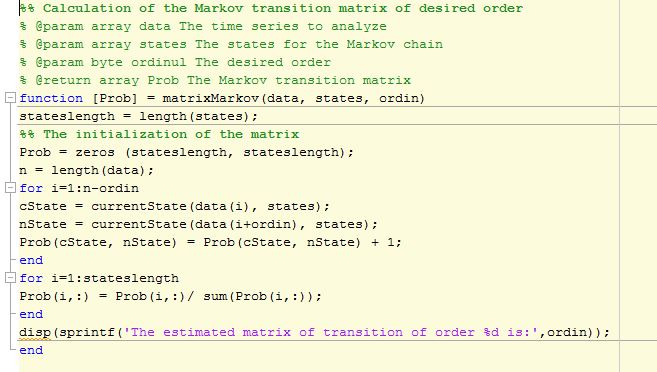
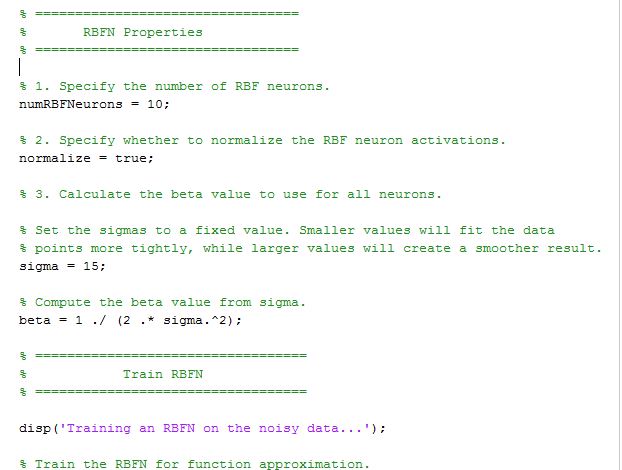


Figure 24: Response of RBF NN on testing data (Potential Values)

**CODE**

****





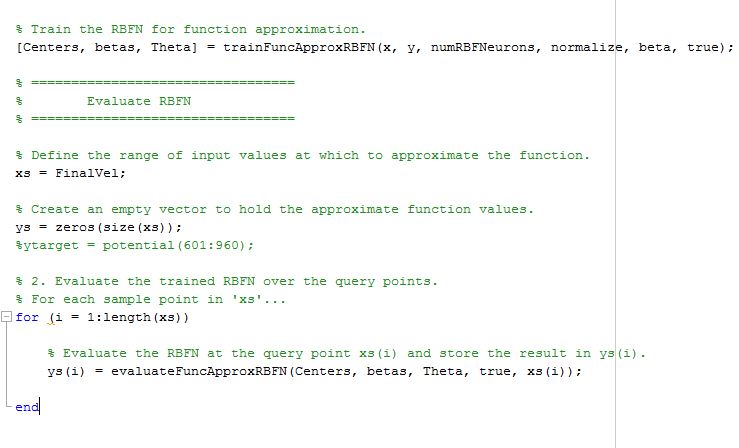


Figure 25: Codes for Markov chain and RBFNN implementation

**RESULT**

Accuracy attained in training Adaboost algorithm :

Accuracy attained in training RBFNN : 91.55%

Relationship obtained between Velocity and Potential Generated:

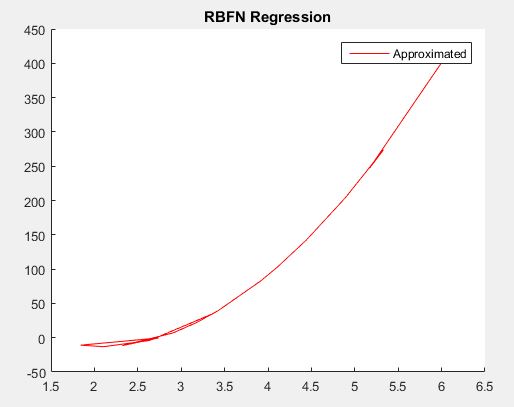


Figure 26 :RBFN Regression X axis: Velocity Y axis: Potential

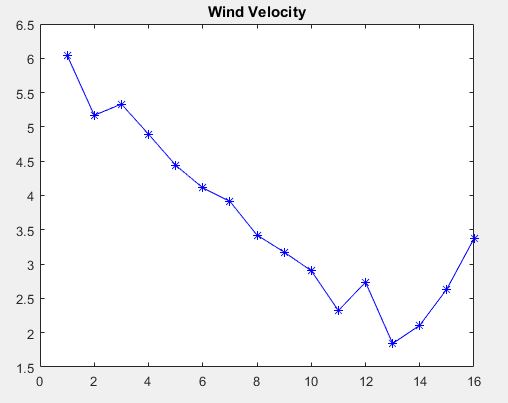


Figure 27: Wind velocity Generated for the next 16 points

**CONCLUSION**

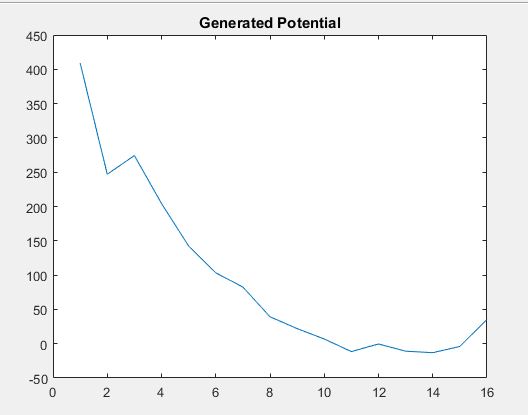


Figure 28: Generated Potential for the next 16 points

Statistical and Machine Learning tools were employed to forecast the wind potential generated. Results are in accordance with the general trend.

Other factors involved in predicting the potential values generated by the turbine viz temperature and direction of wind can also be included to train the model.

**BIBLIOGRAPHY**

1. Ziming Zhu, Doudou Zhou, and Zhong Fan “Short term forecast of wind power generation based on SVM with pattern matching”
2. A.V. Durga Ganesh Reddy and L. Tarun Varma “Wind Power Forecasting Without Using Historical Data”
3. Jethro Dowell “Spatio-temporal Prediction of Wind Fields”
4. Justin Phillip Heinermann M.sc. “Wind Power Prediction with Machine Learning Ensembles”
5. Hiroyuki Mori and Soichiro Okura “Application of S-Transform-based Artificial Neural Network to Wind Speed Forecasting”
6. Sreenu Sreekumar, Kailash Chand Sharma and Rohit Bhakar, SMIEEE “Optimized Support Vector Regression Models for Short Term Solar Radiation Forecasting in Smart Environment”
7. Nitin Anand Shrivastava, Abbas Khosravi and B .K.Panigrah “Prediction Interval Estimation for Wind Farm Power Generation Forecasts Using Support Vector Machine”
8. [Online tsfresh source webpage] <http://tsfresh.readthedocs.io/en/latest/index.html>
9. Ahmad, S. / Wan Hussin, W. M. A. / Bawadi, M. A. / Mohd. Sanusi, S. A.: “Analysis of Wind Speed Variations and Estimation of Weibull Parameters for Wind Power Generation in Malaysia”. *Working Paper, Universiti Sains Malaysia*, http://eprints.usm.my/id/eprint/10959, Accessed 16 May 2015, 2009.
10. Aksoy, H. / Fuat Toprak, Z. / Aytek, A. / Erdem Ünal, N.: “Stochastic generation ofhourly mean wind speed data”. *Renewable Energy* 29(14), 2004, pp. 2111–2131.
11. Song, Z. / Geng, X. / Kusiak, A. / Xu, C.: “Mining Markov chain transition matrix from wind speed time series data”. *Expert Systems with Applications* 38(8), 2011, pp. 10229–10239.