BITS Pilani
Hyderabad Campus

Smart Grid for Sustainable Energy (EEE F424)

First Semester: 2022-23

Project Report

Title of the Project paper

Forecasting the Wind Generation Using a Two-Stage Network Based on Meteorological Information

Group Number: 5

Project Paper Number: 19

Submitted by Shubhan Mital

Name & ID Number of group members:

Name	**ID Number**
Shubhan Mital	2019B4A30900H
Tejasvi Chabbra	2019B3AA0636H
Raghav Gupta	2019B4A30927H
Dishank Srivastava	2020A8PS0573H

Date: December 6, 2022

TABLE OF CONTENTS

Introduction

Wind energy has numerous advantages, which is why it is one of the world's fastest-growing energy sources. Researchers are attempting to address technical and socioeconomic hurdles in order to expand wind energy's capabilities and community benefits. In this report, we are dealing with one of the papers "Forecasting the Wind Generation Using a Two-Stage Network Based on Meteorological Information."

Although wind power integration has major environmental and economic benefits, its intermittent and stochastic character causes issues in power system operation and planning. Variations in wind farm output raise regulation requirements and impair the operational efficiency of some producing units since a power system must maintain instantaneous balancing between aggregated generation and demand at all times.

The authors have tried to tackle this issue by issuing a better wind generation forecast. This complicated task of forecasting has been tackled by taking into account the time scale and the data available. The shorter time periods of milliseconds/ seconds predictions are used for wind turbine control while the larger time intervals are used for wind integration into the energy system.

The paper we are studying performs this at a 1-48 hr level depending on the demand from the electricity market. The purpose of this paper under review is to propose a practical and cost-efficient model for forecasting the generation of a wind farm. To produce the forecast with a high degree of accuracy and a minimum amount of effort to modify the input data for different wind farms. This aim was achieved by taking into account the non-stationarity and the robustness of the forecasting error.

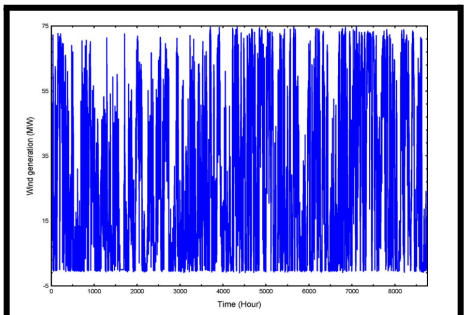


Fig. 2. Ten-minutes wind generation in the wind farm from June 1, 2004 to May 31, 2005.

The figure above taken from the paper explains the 10-minute wind generation variation and the extent to which forecasting can depend on the type of data available.

Task Description and Data Analysis

The paper under analysis uses a wind farm with the following details as mentioned in the table below:

Name	Blue Canyon I wind farm
Location	Southwestern Oklahoma, United States
Starting Date	December 2003
Nameplate Capacity	74 MW
Vicinity	Two meteorological towers within the wind farm with 45 wind turbines

The model in the paper has been established as follows:

1) Ten-minutes data from the wind farm, including the observations of aggregated wind generation of the wind farm.
2) Hourly observations of weather data from the surrounding weather station
3) Hourly meteorological forecasts at the locations of wind farms and the weather stations

Period of study used for further analysis is June 1, 2004, to May 31, 2005. The figure below is evidence supporting the non-stationarity of data

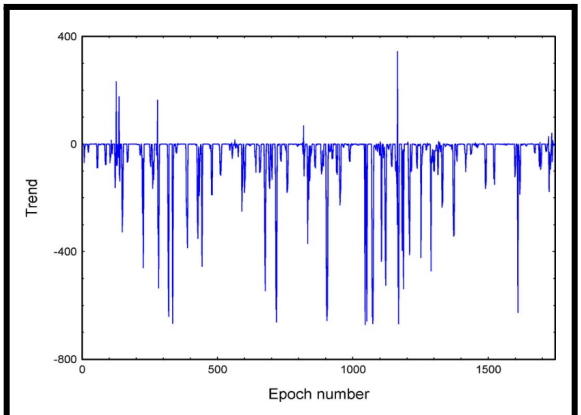


Fig. 3. Trend statistic for ten-minutes wind generation series. RAQ parameters: delay = 6; embedding dimension = 3; radius = 1.5; line definition = 10 points.

The trend statistic departs away from zero most of the time, and there exist many negative spikes which is an indication to point out the non stationarity. The relationship between the wind speed and direction is depicted by the figure below.

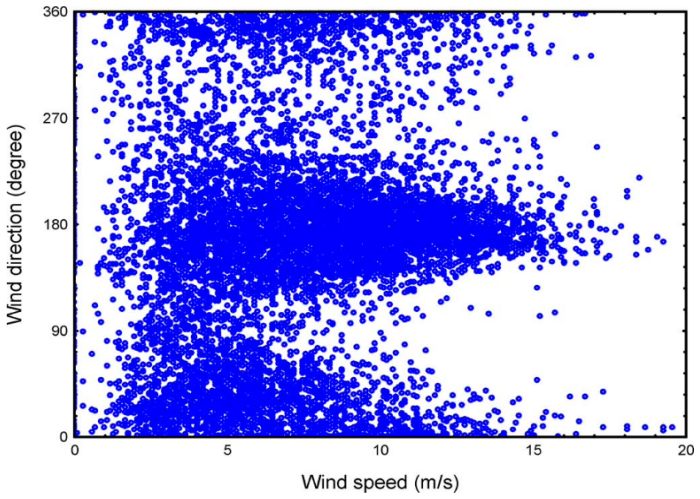


Fig. 4. Correlation between the local wind speed and the direction from June 1, 2004 to May 30, 2005. 0 or 360 is recorded as north, 90 = east, 180 = south, and 270 = west.

shows that most of the high-speed winds came from two directions primarily and those are south and northeast bands and low-speed winds existing all throughout in all directions. Wind speed is the key information for generation forecasting. The correlation calculated from the use of historical data yields a value of 0.66 which is positive and evidently visible in the plot below.

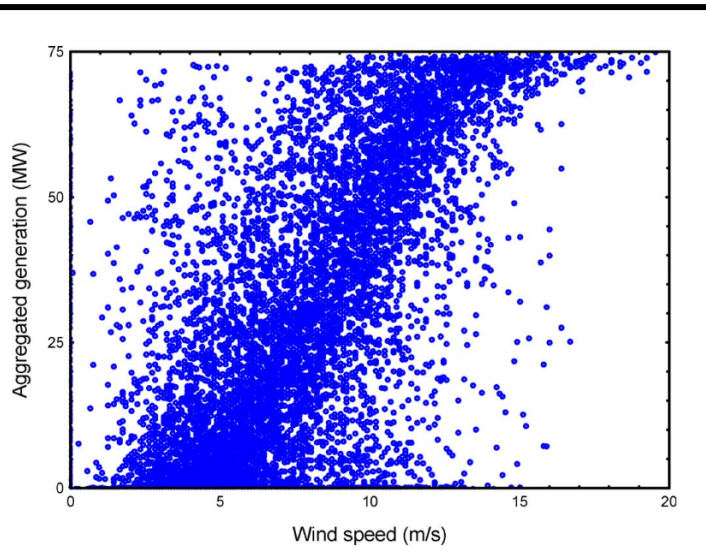


Fig. 5. Correlation between the wind speed and the aggregated wind generation at the wind farm from June 1, 2004 to May 30, 2005.

based on this analysis a forecasting system has been developed and laid down as discussed in the later sections of this report.

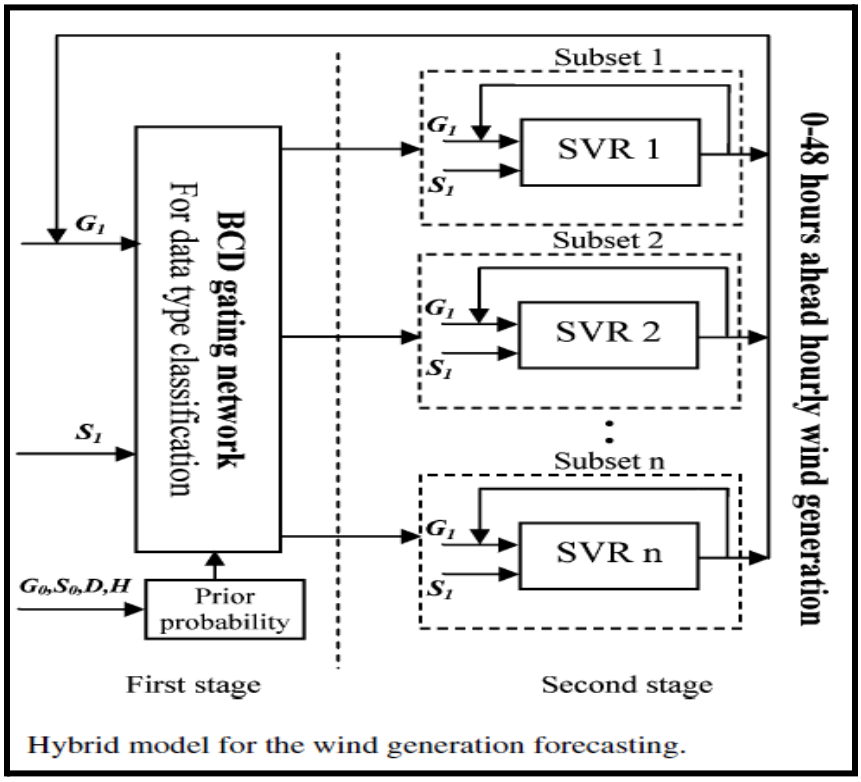
Method and Learning Algorithm Of Improved Model

A) Architecture of forecasting system

Our first task is to implement a time series-based nonlinear discrete time dynamical function that will fit the forecasting. For this, we are making use of the given function:

$$y(t+1) = f(y(t), \ldots, y(t-m+1); X)$$

Here y(t) represents hourly aggregated wind generation of the farm at time t, m is the order of the dynamical system(predetermined) and X is a vector for the control parameters of the dynamical system (eg: wind speed, direction, humidity, etc). The proposed forecasting system is based on a hybrid architecture consisting of BCD classifiers and a group of SVRs.



Hybrid model for the wind generation forecasting.

The primary job of a BCD classifier is to cluster the input dataset into several subsets with similar dynamical properties in an unsupervised manner. Since the procedure takes place in an unsupervised manner, the speed is prioritized more than anything which eventually increases the efficiency. Then comes the role of the SVRs which are responsible for filling the training data in each subset in a supervised manner. training data is filled on the basis of the stationarity and similarity between other dynamic properties within the time series. The structure as a combination of both BCD and SVR is perfectly capable of capturing the dynamics of wind generation time series. BCD identifies the set of clusters with maximum posterior and can avoid the risk of overfitting. Along with this, based on the unique theory of structure risk minimization principle, SVRs are very resistant to overfitting problems as well. Hence, they have high generalization performance on multiple time series.

The input variables for the BCD and SVR are also completely different as the input variable for the SVR includes wind generation and wind speed, whereas WG/WS, humidity, and wind direction constitute the input variables for the BCD. Due to this, the overall sensitivity of the system improves while the risk of loss of information diminishes. The overall forecasting procedure can be considered as a voting method among the BCD and the SVRs where only one SVR output is used for the final forecast.

If we wish to predict the 48h ahead of wind generation, we need to make changes in our above function as some of the values such as wind generation would be unknown. To predict the forecast in the near future, past and forecast values of the input, as well as the network's output at previous time steps are taken into use. It provides us with a new function:

$$\hat{y}(t+1) = f(\hat{y}(t), \hat{y}(t-1), \ldots, X)$$

The new function on LHS represents the one step ahead of wind generation i.e t+1 time considering t time at present. X remains the same as before.

B) Selection of input variables

Second step of the method and learning algorithm is the selection of input variables. These variables have to be selected very carefully as they usually have a huge impact on the performance of the

forecasting model. In this paper, we will focus on two different aspects of data which will help us in narrowing down and choosing the most important input variables to give as accurate results as possible before moving on to our model.

Firstly there are many exogenous variables that have a relationship with wind generation but it is not practical to use all of them as it will slow down the model considerably without much gain. Hence we choose the ones that have a significant degree of correlation to wind generation. Looking at table 1 we can see that wind direction and gust speed have a positive correlation to wind generation and hence we will choose those.

Secondly, we have meteorological data from 7 different weather stations but it's important to choose which ones are suitable for forecasting. Again looking at the cross-correlation between wind speed in these stations with generation as indicated by table 2 and setting a benchmark of 0.6, we choose stations 2 3 5.

Now we select input variables for the BCD classifier. We already have time series G_i which is a wind power generation and S_i which is wind speed. Along with that, we use average wind generation, average wind speeds, forecasted average wind direction, and humidity to find the prior probability of the BCD. Adding other factors like atmospheric pressure decreases the speed of the model without altering the performance of the system significantly and hence is ignored.

Moving on to the SVR network there are two important elements that serve as input data. First is the hourly wind power generation of the previous 6 hours that is used to create a time series style data and second is the wind speeds of the wind farms as well as the above-selected weather stations. Even this data is in form of a time series comprising the forecasted wind speed at the target hour as well the previous 5 hours.

TABLE I
CROSS-CORRELATIONS BETWEEN AGGREGATED WIND GENERATION AND DIFFERENT OBSERVATIONS WITHIN THE WIND FARM

	Wind direction	Barometric pressure	Temperature	Humidity	Gust speed
Aggregated WG	0.14	-0.02	-0.02	-0.07	0.62

TABLE IV
LIST OF INPUT DATA OF THE BCD CLASSIFIER

Input	Variable	Detail description
1-6	G_l	Wind power generation series
7-30	S_l	Wind speed series
31-40	Variables to determine prior probability	G_0: Average wind generation
		S_0: Average wind speed
		D: Forecasted wind direction
		H: Average humidity

TABLE III
LIST OF INPUT DATA OF THE SVR NETWORK

Input	Variable name	Lagged value (hours)
1-6	Hourly wind power generation (G_l)	1,2,3,4,5,6
7-12	Hourly wind speed 0 (S_l)	0,1,2,3,4,5
13-18	Hourly wind speed 2	0,1,2,3,4,5
19-24	Hourly wind speed 3	0,1,2,3,4,5
25-30	Hourly wind speed 5	0,1,2,3,4,5

The wind speed 0 means the speed within the wind farm, and accordingly number 2, 3, or 5 means weather station 2, 3, or 5.
The hour of the predication is assumed at 0, the lag 0 represents the target instant, and the 6 lagged hours means the values that were measured 6 h earlier than the hour of predication.

TABLE II
CROSS-CORRELATIONS BETWEEN THE WIND SPEEDS IN THE WEATHER STATIONS AND THE AGGREGATED GENERATION OF THE WIND FARM

Stations	1	2	3	4	5	6	7
Cross-correlation With generation	0.56	0.66	0.67	0.54	0.62	0.50	0.51

C) Learning Algorithm Part 1: The BCD classifier

Third step of the method and learning algorithm as well as the first part of the model we are using is the BCD classifier. BCD stands for Bayesian Clustering by Dynamics The basic method that is used in BCD is the clustering method which uses the principle that 2-time series are highly similar when they are generated by the same or similar stochastic process. The BCD classifier uses an auto-regression model which in simple words forecasts the variable of interest using a linear combination of past values of the concerned variable. We assume that the series follows an auto-regressive model of order p which is the value of an element of series at time t (>p) and is a linear combination of values observed in the previous steps. We can express this model as $x_j = X_j\beta_j + \varepsilon_j$. Here small x_j is a vector and X_j is (n-p) X (p+1) regression matrix. β_j is the vector of autoregressive coefficients while ε_j is a vector of uncorrelated errors which has a mean of 0 and variance of $(\sigma_j)^2$.

Now we have to cluster our data for which we use a model-based clustering procedure. A set of clusters

C_1 C_2……C_k each having m_k time series is represented by a model M_c. Each cluster is treated independently and can be described by a single autoregressive equation. Each cluster C_k can be modeled as $x_k = X_k\beta_k + \varepsilon_k$ where vector x_k and matrix X_k are defined by stacking m_k vectors x_{kj} and regression matrices $X_{kj}.x_k=(x_{k1}\ x_{k2}…..x_{kmk})^T$ and $X_k=(X_{k1}\ X_{k2}…..X_{kmk})^T$. Once we obtain all the clusters of all the different clustering models we must rank them according to their posterior probability. We do this as - the higher the posterior probability, the higher the accuracy of the predicted clustering model. We calculate this using Bayes theorem with the formula $P(M_C|x) \propto P(M_C)f(x|M_C)$. Here $P(M_c)$ is the prior probability of M_c and $f(x|M_c)$ is the marginal likelihood. This is calculated using the equation $f(x \mid M_c) = \int f(x \mid \theta_c) f(\theta_c)\, d\theta_c$.$\theta_C$ in this equation is a vector of parameters that describe the likelihood function while $f(\theta c\ )$ is the prior density. Simplifying the previous equation to remove the integral we

$$f(x \mid M_c) = \frac{\Gamma(1)}{\Gamma(1+m)} \times \prod_{k=1}^{c} \frac{\Gamma(m_k/m + m_k)}{\Gamma(m_k/m)} \times \frac{(\mathrm{RSS}_k/2)^{(q-n_k)/2}\, \Gamma\,(n_k - q)\,/2}{(2\pi)^{(q-n_k)/2} \det(X_k^T X_k)^{(1/2)}}$$

obtain. Here n_k is the dimension of vector x_k and RSS is the residual sum of squares. Only when all the models become apriori equally likely do we get our probabilistic scoring metric. However, these clusters grow exponentially and we have to use an agglomerative search strategy to merge our time series into clusters. Here we start by assuming that the m wind generation time series are all generated by different processes. Now we combine the 2 clusters and calculate the marginal likelihood of the m-1 clusters. We do this for all possible combinations and the one having the maximum is merged. However, this merging is only accepted if $f(x \mid M_m) < f(x \mid M_{m-1})$ otherwise the procedure stops. We keep repeating this procedure until no more merges are acceptable. However, this method is highly demanding computationally and takes a long time when there are too many clusters.

To reduce this a heuristic strategy based on a measure of similarity between time series is applied. In this strategy instead of the pairwise marginal likelihood, we first calculate the pairwise similarity measure for all the time series. Again the most similar pair is merged if $f(x \mid M_m) < f(x \mid M_{m-1})$. Else we look at the second most similar pair, check its acceptability, and so on. Once combined this cluster has a profile that is the average of the two combined clusters. This procedure is continued till no more acceptable mergings are found. The similarity measure which has been used in the BCD classifier is the

Euclidean distance between 2 times series $S_i=\{x_{i1}, \ldots, x_{in}\}$ and $S_j=\{x_{j1}, \ldots, x_{jn}\}$ and is calculated by the equation

$$D(S_i, S_j) = \sqrt{\sum_{t=1}^{n} (x_{it} - x_{jt})^2}.$$

D) Learning Algorithm Part 2: The SVR network

Consider the training data $(x_1,y_1),(x_2,y_2),\ldots,(x_n,y_n)$ where x_i is the input pattern and y_i associated output for x_i, This corresponds to a vector regression that solves an optimization problem:

$$\min_{\omega,b,\xi,\xi^*} \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{n}(\xi_i + \xi_i^*)$$

$$\text{subject to } y_i - (\omega^T\phi(x_i) + b) \le \varepsilon + \xi_i^*$$

$$(\omega^T\phi(x_i) + b) - y_i \le \varepsilon + \xi_i$$

$$\xi_i, \xi_i^* \ge 0, \quad i = 1, \ldots, n$$

x_i : mapped to a higher dimension space by the function Φ

ξ_i^* : slack variable of the upper training error subject to

ε - insensitive tube

$(\omega^T\phi(x_i) + b) - y_i \le \varepsilon$

Here 'C' is actually the compromise between the flatness and other losses. We have a lot of regression quality control parameters which include the width of the tube, mapping function, and cost of error C.

From the above equation, it is evidently clear that we have to put most of the data of x_i into the tube. Failing so gives rise to error ξi or ξ*i that we need to minimize in the objective function. As we know that SVR is perfectly capable of avoiding overfitting, as it minimizes the training error $C\sum_{i=1}^{n}(\xi_i + \xi_i^*)$ and regulates the term $\omega^T\omega/2$. In comparison with other least square regressions, studies show that SVR is the most general and flexible treatment for regression problems.

As we know that Φ maps xi to a high- or infinite-dimensional space, solving for ω in the same space will be a tedious task. So we make use of the theory of Lagrange.

$$\max_{\alpha_i, \alpha_i^*} -\frac{1}{2} \sum_{i,j=1}^{n} (\alpha_i - \alpha_i^*)^T Q(\alpha_j - \alpha_j^*) - \varepsilon \sum_{i=1}^{n} (\alpha_i + \alpha_i^*)$$

$$+ \sum_{i=1}^{n} (\alpha_i - \alpha_i^*)$$

$$\text{subject to, } \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) = 0$$

$$Q_{ij} = \phi(x_i)^T \phi(x_j)$$

α_i and α_i^* are Lagrange multipliers.

We apply the Kernel trick to do mapping implicitly as the inner product might be expensive to compute as φ(x) has too many elements within it. The following are typical examples of the Kernel trick. Here, γ, c0, and d are kernel parameters that help us to compute inner products in a very high dimensional space so easily even without knowing φ(x).

Implicit mapping: Kernel trick

$\phi(x_i)^T \phi(x_j) = (\gamma x_1^T x_2 + c_0)^d$: Polynomial Kernel

$\phi(x_i)^T \phi(x_j) = e^{-\gamma(x_1 - x_2)^2}$: RBF Kernel

↓

Can be calculated even without knowing Φ(x)

Numerical Experiments

Numerical experiments in this paper were conducted to check the performance of the proposed forecasting model and to provide substantial proof to validate it. We will discuss this in three sections, namely:

1. Data Collection and Implementation: where we will discuss the learning procedure of our proposed model
2. Bayesian Clustering Analysis: where we will discuss the results of clustering of our BCD and perform RQA Analysis
3. Numerical Results: where we will conclude our experiments

A) Data Collection and Implementation

To predict and validate the performance of the suggested model, two months - corresponding to June and December 2005 - have been chosen. The model was trained using data from June 1, 2004, to May 30, 2005. It is not necessary to utilize the test sets throughout the learning process because they are entirely distinct from the training sets.

The 48-hour predictions are created each morning for the generating schedule and energy markets of the following operation day, at the request of the power companies operating the wind farm.
The learning procedure of the proposed architecture is outlined as follows:

a. The complete training dataset needs to be divided into two categories: *the training set*, which is used to update the network parameters, and *the verification set*, which is used to assess performance.
b. The prior probability of the BCD classifier needs to be determined according to the control variables
c. Use Bayesian clustering by dynamics to categorize the input data type in accordance with the dynamics of the WG/WS time series.
d. Train the SVR to fit the data subset according to Lagrange's Theory in each subset of the input space
e. Forecast the wind generation using the dataset for verification and calculate the mean absolute percentage error (MAPE)
f. The final ones need to be selected from the network parameters at the minimum of the MAPE

The clustering findings may be the only thing that separates various systems. The suggested model may be simply adapted to various power systems since the BCD classifier can determine the best number of clusters as well as the ideal assignment of time series to clusters.

After the training procedure is completed, the following test process is applied to verify the proposed model:

a. Using information from the test hour and preceding hours, determine the kind of input test data using the BCD classifier.
b. Output the next hour's wind generation using the corresponding SVR network
c. Derive the 2-48 h estimates repeatedly using the network's outputs from earlier time steps and the inputs' prior values.

B) Bayesian Clustering Analysis

Here, we have briefly described the clustering results of BCD and carried out RQA analysis of the WG series in each cluster. This research illustrates the efficiency of Bayesian clustering in addition to assisting with parameter optimization throughout the training process. We have restricted ourselves from showing the analysis of the first testing dataset in the numerical experiment, as all the other datasets are on similar lines.

While conducting the numerical experiment, the whole training dataset was partitioned into nine clusters by the BCD classifier. Four of them are allotted the test data after that. The RQA analysis is utilized to look at the non-stationarity of the WG series for these four subgroups. The RQA specifications remain the same. Indicating the effect of clustering on reducing the nonstationarity of the WG series, the trend statistics for the subgroups are nearer to zero than those for the entire dataset. We did not plot the trend statistic in each cluster for the sake of simplicity.

C) Numerical Results

Finally, to compare the performance of the proposed model with the existing models, we will have to introduce some new parameters which indicate the accuracy of recall., as follows:

a. Mean Absolute Error (MAE)

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}(|y_{ai} - y_{fi}|)/P_N \times 100\%$$

y_{ai} is the actual value,
Y_{pi} is the forecast value,
P is the nameplate capacity,
and **n** is the total number of the value predicted.

b. Root Mean Square Error (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{(y_{ai} - y_{fi})}{P_N}\right)^2} \times 100\%$$

y_{ai} is the actual value,
Y_{pi} is the forecast value,
P is the nameplate capacity,
and **n** is the total number of values predicted.

Now, for comparison, simulations that are compared with Persistent (or Naive) forecasts are also conducted. The persistence algorithm, in layman's terms, uses the value at the previous time step (t-1) to predict the expected outcome at the next time step (t+1). In our case, this approach assumes that the forecast value of the wind power at the ith future time steps, Pˆper (t + i/t), is the last measured one available, P(t). The benefit gained by using the proposed model is measured as the accuracy improvement over the persistent model, given by

$$\text{Im}\, p = (\text{Err}_P - \text{Err}_m)/\text{Err}_p \times 100\%$$

where, Err is the evaluation criterion (i.e., MAE or RMSE) of the persistence
Err is the evaluation criterion of the proposed model.

The numerical results of the proposed model and the improvements over the persistent model are shown in the tables below. The persistent predicting errors are shown above the tables. The outcomes for many experimental instances, which correlate to various time scales and input variables, are given.

(a) Hour ahead predication

Persistence error: MAE=7.84 %P_n RMSE=11.93 %P_n

SVR Inputs (stations)				Errors(% of P_n)		Improvement (%)	
0	3	2	5	MAE	RMSE	MAE	RMSE
O	×	×	×	7.12	11.13	9.18	6.71
O	O	×	×	6.95	10.87	11.35	8.89
O	O	O	×	6.74	10.64	14.03	10.81
O	O	O	O	6.65	10.54	15.23	11.66

(b) 24-hour ahead predication

Persistence error: MAE=21.24 %P_n RMSE=29.84 %P_n

SVR Inputs (stations)				Errors(% of P_n)		Improvement (%)	
0	3	2	5	MAE	RMSE	MAE	RMSE
O	×	×	×	15.13	21.06	28.77	29.42
O	O	×	×	14.88	20.55	29.94	31.13
O	O	O	×	14.56	20.08	31.45	32.71
O	O	O	O	14.38	19.74	32.28	33.84

(c) 48-hour ahead predication

Persistence error: MAE=25.42 %P_n RMSE=34.81 %P_n

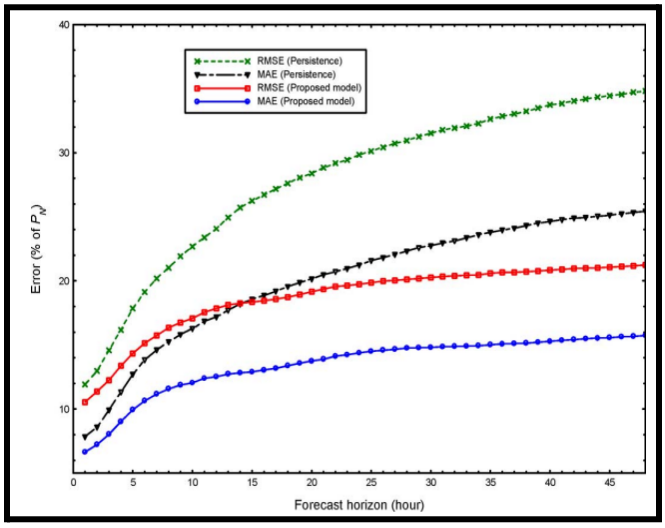
SVR Inputs (stations)				Errors(% of P_n)		Improvement (%)	
0	3	2	5	MAE	RMSE	MAE	RMSE
O	×	×	×	16.33	22.17	35.76	36.31
O	O	×	×	16.09	21.84	36.70	37.26
O	O	O	×	15.89	21.56	37.49	38.06
O	O	O	O	15.73	21.24	38.12	38.98

From the tables shown above, two major conclusions can be drawn:

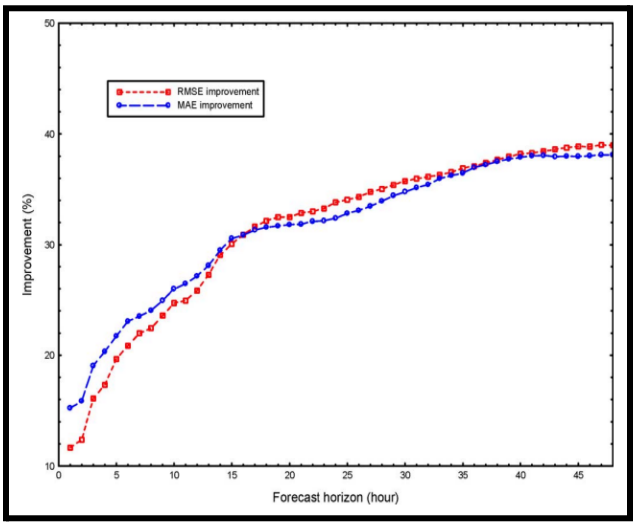
a. The suggested forecasting model surpasses the persistent forecast, even in the first time step
b. As the prediction window gets further out into the future, the inclusion of data from remote stations increases forecasting accuracy and its significance.

The first graph shown below shows the MAE and RMSE performance obtained by the proposed model, along with the performance obtained by persistence.

The second graph shown below shows the percentage improvement over the persistence obtained by the proposed model.



MAE and RMSE performance comparison



Percentage improvement over the persistence model by the proposed model

Hence, the proposed model can provide a much more accurate forecast in the whole forecast horizon. It is able to produce robust multistep ahead estimations compared with the persistent forecasts, and considerable improvement up to more than 40% over the persistence is achieved.

Conclusion

Through this paper we have developed a model which gives us a clear understanding of the projections on wind power generation for any wind farm for the next 48 hours. We have used machine learning and based our model on the BCD classifier and SVR network which gives superior accuracy as compared to a basic time series model as shown by us above.

The model has two parts namely the BCD classifier and the SVR network. After constructing the basic architecture of the model and selecting the input variables to be used based on correlation factors we move on to the classifier. In essence, the BCD classifier divides the data into small clusters and remerges these clusters based on a similarity measure i.e. Euclidean distance. This clustered data is then input into the SVR network which is used to fit our trained data within the subsets created by BCD in accordance with the dynamics of the data over the time frame.

Apart from being cost-efficient and easily implementable, the proposed model has several other benefits as well. We are able to handle the nonstationarity within the power and speed of the wind as we can quickly modify the data for different wind farms because of its strong robustness. The nonstationarity is dealt with by the clustering mechanism of the BCD and the subsequent data fitting mechanism within the corresponding subsets with the help of SVR.

Our 48 hours predictions over weather forecasting give better results with this new modified model as compared to the old persistence model which we can see by looking at RMSE and MAE values. Further, as the time duration of the forecast increases, we get more and more accurate results. So with this, we conclude that our forecasting predictions will give better results if a relatively long time frame is taken into consideration and we settle at 48 hours.

Contributions

Tejasvi Chabbra (2019B3AA0636H) - Worked on the introduction as well as the initial task description and data analysis. The initial model was also designed by him which proved that a superior ML model was needed.

Shubhan Mital (2019B4A30900H) and **Raghav Gupta(2019B4A30927H)** - Worked on the method and learning algorithm. Raghav handled the Architecture of the forecasting System and Learning Algorithm part 2: The SVR Network. Shubhan worked on the Selection of Input Variables and Learning Algorithm part 1: The BCD Classifier.

Dishank Srivastava (2020A8PS0573H) - Worked on the numerical experiments part. It consisted of Data Collection and Implementation as well as Bayesian Clustering Analysis and Numerical Results.

Shubhan Mital (2019B4A30900H) and Raghav Gupta(2019B4A30927H)- Worked on the conclusion.

All 4 team members have made equal contributions to this project report.