Carnegie Mellon University

Assessing efficacy of machine learning methods to predict power output of wind energy in order to assist economic dispatch

18618: Smart Grids & Future Electric Energy Systems

Divjot Sinha (dsinha)
Narendra Jhabakh (njhabakh)
Varun Deshpande (varund)

Contents

Contents i
1. Introduction 1
2. Dataset Description 2
3. Data Analysis Methodology 3
3.1 Data Selection 3
3.2 Data Processing 3
4. Analysis 4
4.1 Machine learning models 4
4.1.1 Support Vector Machines 4
4.1.2 Artificial Neural Networks 5
4.2 Metrics for comparison 6
4.3 Cases & Methodology 6
4.3.1 Case definitions 6
4.3.2 Methodology 7
5. Results and Inferences 7
5.1 Tabulated results 7
5.2 Inferences and discussion 7
5.3 Graphs from results 8
6. Future work 10
6.1 Model Accuracy 10
6.2 Economic Dispatch 10
7. Software and Packages 11
8. References 12

1. Introduction

Economic Dispatch (ED) refers to the short-term determination of power output so as to meet the system load at a minimum cost keeping in view transmission and operational constraints. The goal of economic dispatch is to find optimal and reliable generation dispatch. The prevalent use of alternative sources of energy like solar and wind in today's power generation systems makes the ED problem even more interesting by adding uncertainty and intermittency issues to an already complex situation. Previously, the ED problem only took into account power produced by conventional energy power generators that used fossil fuels. With emphasis on zero emissions, the power generation scenario is rapidly evolving and wind energy in specific is experiencing exponential growth world over. Global warming and climate change have compelled power companies to diversify and generate electricity using wind especially in countries like USA, Denmark, Spain, Germany and India. In recent years the cost of wind power generation has come down substantially with main capital cost being the initial wind turbine installation. The wind farm land is also being used for agricultural purposes making it an extremely lucrative business in developing countries

The main issues involving incorporation of wind power generation into the ED scenario are intermittency, wind speed prediction and power fluctuation in view of security and reliability essential of an electric energy power system. To effectively use wind energy for power systems it is essential to be able to predict wind speed/ generation over a period of time.

Our approach involves use of machine learning techniques, specifically support vector machines and artificial neural nets to predict wind speed and forecast wind power generation.

2. Dataset Description

A typical 2MW to 5MW wind turbine has more than 150 sensors which capture data relating to wind speed, wind direction, ambient temperature, power output, energy generation, vibrations in the tower, temperatures of different components including nacelle, generator, gearbox and so on. When installing and commissioning a wind farm for utility scale application, turbine manufacturer's setup a full scale SCADA (Supervisory control and data acquisition) system on the farm in order to collect this sensor data and store it for later analysis of performance, health and to make budgetary and operations decisions

The data collected over SCADA is sent over communication network to the farm control room & database of the operations and maintenance entity which in many cases is the turbine manufacturer itself.

The data collected and stored is generally of three types:

- Analog & counter data
  Includes analog variables like temperature, vibration, power and counters like machine downtime, grid downtime, machine fault time, energy generation and so on
- State level data
  Includes flags, and machine operation states depicting whether the turbine is in the 'Active, 'Pause', 'Waiting for wind', 'Maintenance stop' and so on.
- Alarms
  Includes specific alarms for faults relating to blade angles, voltage and current droops, gearbox oil slumps and so on.

All three types of data are accompanied by the turbine's identification number and the timestamp. In case of alarms the exact time stamp gets recorded where as for the analog and counter data, the sensor is polled by the SCADA system at specific time intervals (generally 10 or 15 minutes) and is stored on a server in a SQL database or equivalent

For the sake of this project, we contacted wind and solar analytics startup in India –"Algo Engines" - http://algoengines.com/

They agreed to give us limited access to their SQL server where we were able to get access to three years of data for one particular wind farm in India.

Apart from the fact that the wind farm whose data is being chosen is from India, for confidentiality sake the company would not share the name of the power producer, the location of the farm and so on.

3. Data Analysis Methodology

3.1 Data Selection

Out of the information available to us we chose to work with one year of data and selected data from November of 2013 to October of 2014. We utilized the columns: Turbine ID, Time stamp, Number of active turbines on the farm, Average farm wind speed, Ambient temperature.

These timestamps were at a resolution of 10 minute average values.

3.2 Data Processing

PCA (Principle component analysis) has been done by numerous studies to realize that 8 hour inputs of average hourly wind speed data can best be utilized to predict one or three hours ahead.
The raw data available in SQL database was in chronological 10 minute interval format and we utilized a specific SQL query with the following processing to prepare a dataset fit for training the machine learning models.

- Average out 10 minute data for each hour.
- Use 10 minute average data for hourly average only if number of active turbines for that 10 minute average value is more than half. This is called a valid slot.
- Calculate the hourly average only if more than 3 out of 6 slots are valid slots.
- Arrange 8 consecutive hour average values in order. Call them H1 to H8.
- At the end of the H8 attach the 11^{th} hour average value. Call this P3 for predict 3^{rd} hour ahead.
- Each new row of the dataset will have H1 which starts with the value next H2 of the previous row. The data is therefore left shifted in each subsequent row.
- If any of the H values are missing because they didn't meet the above criteria, all the rows that would contain those H values have been eliminated. This has been done in order to give clean contiguous data to the machine learning algorithm.
- Ambient temperature information has also been appended to the dataset.
- From the timestamp available of the original data, the month number has also been extracted and appended to the above dataset.

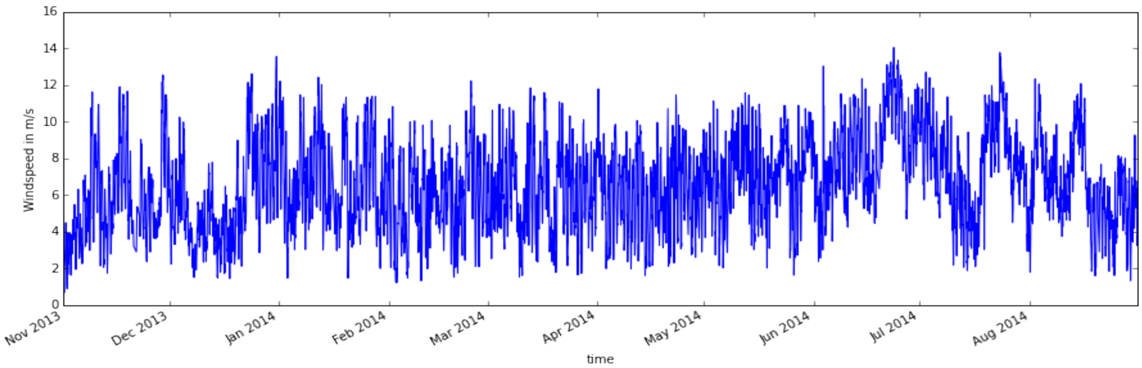


Figure 1: Variation of wind speed over time

4. Analysis

In this study we have attempted to train and use machine-learning models in order to do short term wind speed forecasting. By applying predicted wind speed on the site specific power curve ,we can get predictions for power output of the farm. The power curve data for the site was not provided to us out of confidentiality reasons and therefore we have limited ourselves to studying wind speed predictions.

4.1 Machine learning models

Two popular machine learning techniques have been considered for short term wind speed forecasting: Support Vector Machines and Artificial Neural Networks. Following sections delve deeper into the application of these methods.

4.1.1 Support Vector Machines

Support vector machines (SVMs) are powerful classification/prediction methods in which linear and non-linear decision boundaries can be implemented. Instead of utilizing the entire data set for classification, support vectors i.e. vectors to data points closest to the decision boundary are chosen and only these are used. A hyperplane is constructed at the boundary and the distance of the hyperplane between the support vectors (known as the margin) is minimized during training. This minimization can be optimized by converting the data points to a higher dimension where a linear division exists between the support vectors of the different classes which are used for training.

The minimization step to find the weights w defining the hyperplane dividing the training vectors x_i (ith training set) can be analytically written as:

$$\min_{(w,b,\zeta)} \frac{1}{2} w^T w + C \sum_{i=1}^{l} \zeta_i$$
$$\text{subject to } y_i(w^T z_i + b) \geq 1 - \zeta_i$$
$$\zeta \geq 0, i = 1, 2, ...l$$

where l is the number of training sets, C the penalty term, $\zeta_i$is the error term for the ith training set, and $z_i = \varphi(x_i)$, where φ is a function transforming the features x_i into a higher dimensional space. This formulation is known as the "primal form". Now, the kernel function is defined as $K = \varphi(x)^T \varphi(x)$

This abstraction is useful when x_i is transformed into an infinite-dimensional feature space as in the case of an Radial basis function (RBF) kernel.

However, it is often beneficial to transform the primal formulation what is known as the "dual form" using Lagrange multipliers, which can be evaluated using quadratic programming. The dual form can be written as:

$$\min_{(\alpha)} \frac{1}{2} \alpha^T Q \alpha - e^T \alpha$$
$$\text{subject to } 0 \leq \alpha_i \leq C, i = 1, 2, ...l$$
$$y^T \alpha = 0$$

Where e is a vector of ones, Q is l*l Semi-definite matrix, where $Q_{(i,j)} = y_i y_j K(x_i, x_j)$. The weights for the vectors are thus calculated as $\sum_i^l \quad \alpha_i y_i \phi(x_i)$. The Gaussian RBF(Radial Basis Function) kernel can be written as:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{exp(||\boldsymbol{x}_i - \boldsymbol{x}_j\ ||^2)}{2\sigma^2}$$

This transforms x_i to an infinite dimension space. The prediction for the future 3rd hour using SVM is as shown below under the results section.

4.1.2 Artificial Neural Networks

Artificial Neural Networks (ANNs) are a family of modeling methods used in Machine learning for predicting non-linear decision surfaces. The methodology followed is something very similar to how neurons work. The aim is to estimate or approximate functions f that map X to Y. Artificial neural networks are generally presented as systems of interconnected neurons which exchange information between each other. The links between neurons have numeric weights that are tuned based on experience. This is a strong feature of neural that make them adaptive to inputs and capable of learning.

Each unit of the network gives an output as:

$$unit\ output\ = \frac{1}{1 + exp(w_0 + \sum_{i=1}^{N} \quad w_i x_i)}$$

where w_is are the weights associated and x_is are the input parameters. The weights are initially assigned random values and they are optimised using back propagation. Where the actual outputs are compared with the unit outputs. The error is decreased and weights are optimized using Gradient Descent:

$$\Delta w_i = -\eta \nabla E$$
$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i}$$

Here E is the error term and is given as $E\ = 0.5 * \sum_{i=1}^{N} \quad (t_d - o_d)^2$
The weights are updated after every iteration of optimization by a Δw_i. Here $\eta$is the step size for gradient descent. $t_d$is the predicted observation and $o_d$is the actual observed data point. A more detailed explanation of the implementation is shown here [1]. The results obtained by ANNs are shown under the results section.

4.2 Metrics for comparison

Two popular prediction metrics RMSE & MAPE have been chosen for this analysis and results for both models have been presented through these metrics.

RMSE: Root mean square error
The square root of the mean/average of the square of all of the error in the dataset.

MAPE: Mean absolute percentage error
The mean of the percentages of all the absolute errors in the dataset.

4.3 Cases & Methodology

4.3.1 Case definitions

Three different cases of training data have been selected to be fed as input to the machine learning algorithms and the outputs of each of them in terms of the defined metrics have been tabulated (in section 5.1, along with graphs of predictions versus actual values in section 5.3)

Case 1: 8 hour average wind speeds
Case 2: 8 hour average wind speeds + ambient temperature of latest hour

Case 3: 8 hour average wind speeds + ambient temperature of latest hour + month

4.3.2 Methodology

Given below is the general algorithm followed in the program code for all of the above cases:

- The dataset was loaded into a data frame variable.
- Based on the case, relevant data was extracted from the dataset and segregated into train and test datasets.
- 80% of data was allotted to training and the remaining was reserved for testing.
- The train data was fed as input to the machine learning model along with configuration parameters [2]
    - For SVM: nu = 0.2, epsilon =0.3, Kernal = RBFdot
    - For ANN: hidden layers: 5, iterations: 5000
- The trained model was used to predict the output variable of the test dataset
- Predicted value of the output variable was plotted against the true value
- RMSE and MAPE were calculated on the test data actual vs predicted values

5. Results and Inferences

5.1 Tabulated results

Case	Case descriptions	SVM		ANN	
		RMSE	**MAPE %**	**RMSE**	**MAPE %**
Case 1	8 hour average inputs	1.498	20.45	1.51	20.71
Case 2	8 hour average inputs + ambient temp	1.46	19.87	1.67	23.92
Case 3	8 hour average inputs + ambient temp + month	1.42	20.35	1.79	25.08

5.2 Inferences and discussion

From the above table we can see that in general the Support vector machine has a better accuracy in terms of RMSE and MAPE values when compared to Artificial Neural Networks

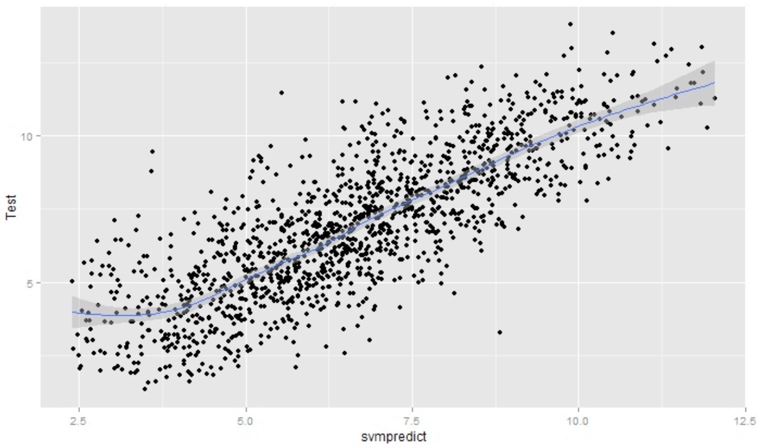
On adding ambient temperature data we see that the Support vector machine has been able to improve its accuracy by a small percentage but adding month data along with temperature has degraded its prediction performance.

In case of artificial neural networks, the model has a similar accuracy to SVM in case 1 but when provided with temperature and month related data we can see that its performance degrades.

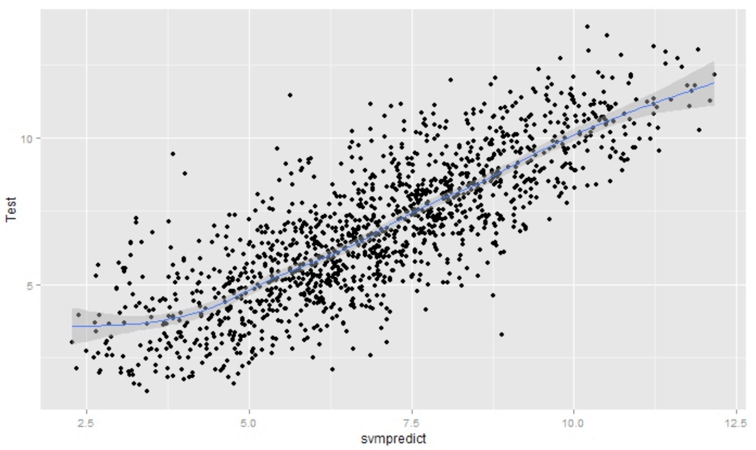
We can infer that SVM is a suitable technique for short term wind forecasting and can be utilized on a farm level scale for prediction of wind speed with a 20% accuracy margin. Further, along with site specific power curve or learned power curves of individual wind turbines it is possible to predict farm power to a better short term accuracy and this information can be valuable for economic dispatch. In Section 6.1 we have discussed how to improve the accuracy of both models.

5.3 Graphs from results

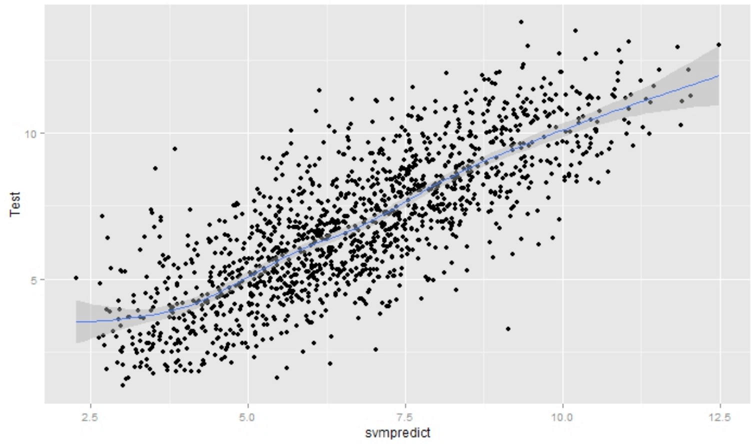
Graph 1: Predict vs Actual – SVM case 1



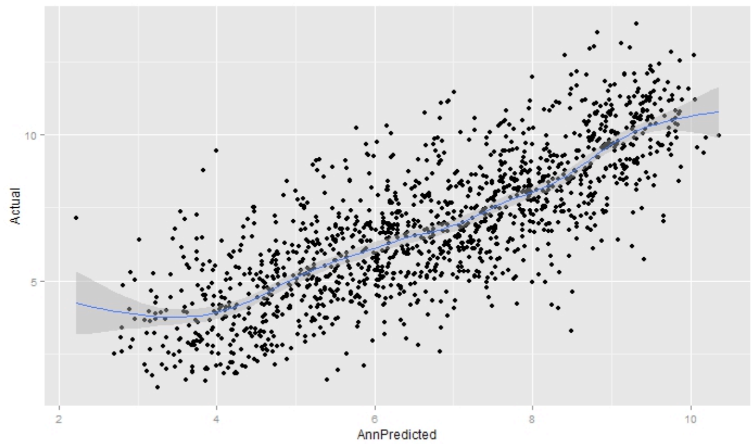
Graph 1: Predict vs Actual – SVM case 2



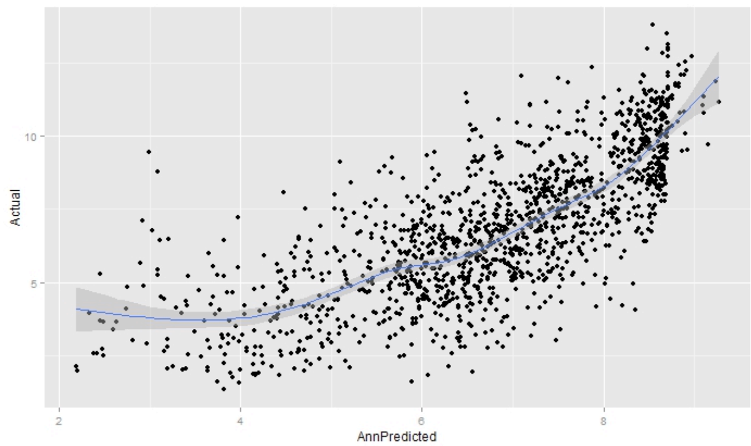
Graph 1: Predict vs Actual – SVM case 3



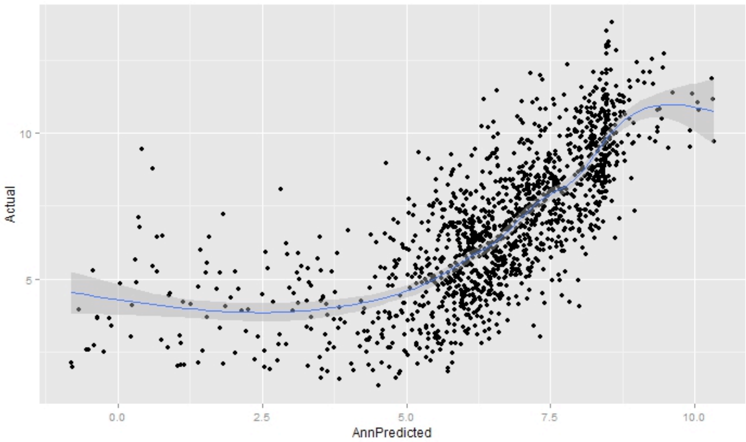
Graph 1: Predict vs Actual – ANN case 1



Graph 1: Predict vs Actual – ANN case 2



Graph 1: Predict vs Actual – ANN case 3



6. Future work

6.1 Model Accuracy

The accuracy of prediction depends on the data, the classifier and its parameters. As seen in the table above addition of ambient temperature data improves accuracy in terms of MAPE and RMSE for Support Vector Machine. Also in the case of Support vector machines, there is a trade-off between C and the variance. Both of them are inversely proportional. Having a high bias term would mean that you have lower variance and vice versa. Greater variance is helpful when the size of the training set is small.
Accuracy of ANNs can be increased by increasing the number of iterations required for optimization. This may prove to be computationally costly, as it takes a considerable amount of time to reach the optimum values while training ANNs.

Another method of improving accuracy for these classifiers which need optimization would be to add an error term with weights corresponding to each error. We have not implemented these methods and leave it for future work and scope.

6.2 Economic Dispatch

The previous sections aim at giving better future predictions for wind power generation. Once we have a good prediction with reliable accuracy, we can dispatch the necessary power to offset the load by traditional power generation plants. A simple yet effective method of optimising the dispatch of power can be done using some assumptions. These assumptions constitute the Static Economic dispatch problem, wherein the goal is to reduce the marginal cost of generation per power generating equipment given the demand. We assume only electric power generation for the economic dispatch model and not something as complex as a Co-generation model. Thus the model as explained by Hetzer *et al.* [3]. can be written as :

$$min\,\{\sum_i^N \quad C_i(P_i) + \sum_i^M \quad C_{wi}(W_i)\}$$

Where $C_i$the cost function of the ith generator producing is $P_i$amount of Power and $C_{wi}$is the cost function of the ith wind generator producing $W_i$amount of Wind Energy.
Here the Cost functions can be either assumed to be linear or as higher order polynomial functions. In the case of linear cost functions:

$$C_i(P_i) \;= a_iP_i$$

And in case of non-linear cost functions (assuming quadratic, but can be higher order as well):

$$C_i(P_i) \;= a_iP_i^2 + b_i \;P_i + c_i$$

As explained in [3] the cost functions pertaining to ordinary generation are quadratic as shown above whereas in the case of wind it is linear.

In this report we have not analysed Economic dispatch and leave it for future work as we do not have the required parameters for other generation including capacity, demand and costs. On availability of this data, we can find out optimum costs and generation schemes given a particular region with its demand.

7. Software and Packages

Analytics software: R version 3.2.3
GUI: R-Studio version 0.99.473
Packages used:
Kernlab (For SVM) [4]
RSNNS (For ANN) [5]
RCPP (supporting package) [6]
GGplot2 (For plotting) [7]

8. References

[1] Nielsen, Michael. "CHAPTER 2." Neural Networks and Deep Learning. Creative Commons Attribution, n.d. Web. 17 Dec. 2015.
[2] Zeng, Jianwu. "Support Vector Machine-based Short-term Wind Power Forecasting." IEEE Xplore. N.p., Mar. 2011. Web. 17 Dec. 2015.
[3] Hetzer, John, David C. Yu, and Kalu Bhattarai. "An Economic Dispatch Model Incorporating Wind Power." *IEEE Transactions on Energy Conversion IEEE Trans. On Energy Conversion* 23.2 (2008): 603-11.
[4] Karatzoglou, Alexandros, and Alexander J. Smola. "CRAN - Package Kernlab." CRAN - Package Kernlab. N.p., 5 Aug. 2015. Web. 17 Dec. 2015.
[5] Bergmeir, Christoph, and José M. Benítez. "CRAN - Package RSNNS."*CRAN - Package RSNNS*. N.p., 12 June 2015. Web. 17 Dec. 2015.
[6] Eddelbuettel, Dirk. "CRAN - Package Rcpp." *CRAN - Package Rcpp*. N.p., n.d. Web. 17 Dec. 2015.
[7] Wickham, Hadley. "CRAN - Package Ggplot2." *CRAN - Package Ggplot2*. N.p., 17 Mar. 2015. Web. 17 Dec. 2015.