DOI: 10.21917/ijsc.2016.0185

LONG TERM WIND SPEED PREDICTION USING WAVELET COEFFICIENTS AND SOFT COMPUTING

Manju Khanna[1], N.K. Srinath[2] and J.K. Mendiratta[3]
[1,2]Department of Computer Science and Engineering, R.V. College of Engineering, India
E-mail: [1]manjukhanna1977@gmail.com, [2]srinathnk@rvce.edu.in
[3]Centre for Emerging Technologies, Jain University, India
E-mail: mendiratta.jk@gmail.com

Abstract

In the past researches, scholars have carried out short-term prediction for wind speed. The present work deals with long-term wind speed prediction, required for hybrid power generation design and contract planning. As the total database is quite large for long-term prediction, feature extraction of data by application of Lifting wavelet coefficients are exploited, along with soft computing techniques for time series data, which is scholastic in nature.

Keywords:

Lifting Wavelets, Soft Computing, Fuzzy Logic, Neural Network, Scholastic

1. INTRODUCTION

Due to global requirement for clean energy, a demand for long-term prediction of wind speed for power generation has increased. Power generation involves multiple power resources such as wind, solar, bio-fuel etc. change with environmental conditions. Due to these changing conditions, long-term prediction of these resources has very difficult. In the past, Andrew Kusiak et al. [1] have used a data mining approach with different algorithms and selected the one with better prediction and with minimum error between predicted and actual data. Jinbin Wen et al. [2] used lifting wavelet transform along with Support Vector Machine (SVM) approach while other research scholars applied ARMA [3] and ARIMA [4] models for the purpose. Iffat A. Gheyas et al. [5] had used generalized radial neural network, wherein the prediction has done five steps ahead of the present instant. Chang has carried out review of various methods as applied by various research scholars for wind speed and power forecasting for different durations [6].

However, all these approaches have led to only short-term predictions, which fail for long-term prediction required. Keeping in mind this gap, present work has tried for a long-term prediction of wind speed by application of soft computing and lifting wavelet transform coefficients.

1.1 WAVELETS

Time series data, with curse of dimensionality [7], suffers due to its redundancy and repeatability. Under the situation wavelets serves as a tool for features extraction and in data compression. They help in building the general data sets or functions. In addition, wavelet has the power to de-correlation of data, i.e. the data can be represented in terms of wavelet coefficients in a compact form. Wim Sweldens [8], [9], [10] introduced a technique of lifting scheme for constructing bi-orthogonal wavelets, without involving the Fourier transform.

This technique performs faster in implementation of wavelet transform by dividing the signal into high and low pass bands for sub-sampling, and by making optimal use of similarities between high and low pass filters. A wavelet $\psi_{j,m}$ is defined as dyadic translation and dilation of a $L^2(R)$ function, $\psi{:}\psi_{j,m(x)} = \psi(2^j x-m)$ is the first generation, mother wavelet.

In a more general setting, the signal without involving these operations, but consisting of properties of first generation wavelet, are named second generation wavelet and are defined by David Donoho [11] as wavelets are optimal bases for compressing, estimating and recovering functions.

Taking a representative time signal s with N samples and decomposing into lifting wavelet transform (LWT) coefficients (h_0, h_1,...,h_N) [12] such that,

$$\sum_{n=0}^{N}(-1)^n n^k h_n = 0 \tag{1}$$

with k varying from 0 to $(N-1)/2$

$$\sum_{n=0}^{N-2k} h_n h_{n+2k} = 0 \tag{2}$$

with k varying from 1 to $((N-1)/2$.

With N assuming to be an odd number, thereby providing $2^{(N-1)/2}$ solutions.

Taking $a_0 = (a_{0n})$ n being 0 to L-1, L being length of the signal s. Defining $a_1 = (a_{1n})$ and $d_1 = (d_{1n})$, as:

$$a_{(m+1)n} = \sum_{k=2n}^{2n+N} h_{k-2n} a_{mk} \tag{3}$$

$$d_{(m+1)n} = \sum_{k=(n+1-N)}^{2n+1} (-1)^k h_{2n+1-k} a_{mk} \tag{4}$$

Taking $m = 0$, produces

$$a_{on} = \sum_{k=-\infty}^{\infty} [h_{n-2k} a_{1k} + (-1)^n h_{2k+1-n} d_{1k]}) \tag{5}$$

for every $n \in Z$.

Application of Eq.(3), Eq.(4) and Eq.(5), a_0 is reconstructed from a_1 and d_1 [13], with energy of a_0 divided among a_1 and d_1, as

$$||a_0||^2 = ||a_1||^2 + ||d_1||^2 \tag{6}$$

Decomposing the sequence a_0 , as a matrix sequences d_1, d_2,..., d_m, a_m whereby the wavelet coefficients in the m^{th} row presenting detailed d_m, and a_m the coarser portion of the signal s i.e. d_i producing the highest while d_{i+1} the next lower frequency region.

As mentioned in Eq.(6) above the reduced dimension signal comprises of the total entropy of the signal while retaining the features of the signal.

1.2 SOFT COMPUTING (SC)

Soft Computing is an innovative technique in constructing computationally intelligent systems that knowledge techniques and methodologies from various sources. For instance, the neural networks recognize patterns and adapt to the changing environment, while, the fuzzy inference systems, incorporate human knowledge, perform inference and decision-making.

The integration of these two complementary approaches together with certain derivatives, i.e. free optimization techniques results in a novel approach called neuro-fuzzy and soft computing system [13].

A neural network (NN) is defined as a machine, designed to model and perform like a human brain. It can perform in a linear or non-linear mode and learn by a supervisor with training samples. Further, it is adaptive in nature thereby adapting to the environment changes. Mathematically, a neuron k is represented by [14]:

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \tag{7}$$

and

$$y_k = \varphi(u_k + b_k) \tag{8}$$

With x_1, x_2,..., x_m as the input signals; w_{k1}, w_{k2},...,w_{km} the synaptic weights of the neuron k; and u_k as a linear combiner output due to the input signals, taking b_k as the bias, $\varphi(\cdot)$ the activation function and y_k as the output signal of the neuron. The activation function is chosen depending upon the problem in hand. The neural networks can be in single or multi-layer configurations. Single layer network comprises of an input and one output layer. The multilayer feed forward network may consist of input, output and hidden layers.

The recurrent networks consist of an additional feedback loop. The feedback layer may be from either a hidden layer or an output layer. For temporal processing in neural networks, Time Delay Neural networks (TDNN) with multilayer feed-forward perceptions are most commonly used. With back propagation, the network compares the output with desired or target response at each delay instant. The output $y_i(n)$ and desired response $d_i(n)$ at time n, produce instantaneous sum of squared error

$$E(n) = \frac{1}{2}\sum_j e_j^2(n) \tag{9}$$

Index j refers to a neuron in the output layer and $e_j(n)$ the error signal defined by,

$$e_j(n) = d_j(n) - y_j(n) \tag{10}$$

With the objective of minimizing a cost function and is defined as the value of $E(n)$ computed over total time,

$$E_{total} = \sum_n E(n) \tag{11}$$

A recurrent neural networks with one or more feedback loops, maps the input to an output space. Such networks respond temporally to an externally applied input signal. These networks with single input, applied to tapped delay line memory of q units, produce an output ahead of input by one-time unit A model with present value of input $u(n)$ produce an output $y(n + 1)$ with unit delay, and are named as Nonlinear Auto-Regressive with exogenous inputs (NARX), with mathematical presentation

$$y(n + 1) = F(y(n)...y(n - q + 1), u(n)...u(n - q + 1)) \tag{12}$$

where, F being a nonlinear function of its arguments. A dynamic network with a tapped delay at the input, which is named as Focused Time-Delay Neural Network (FTDNN), without back propagation, and is well suited for time series prediction. Such a network trains faster than other dynamic networks. For multi-step ahead prediction, the network is configured in the form of a closed loop.

However, each time a neural network is trained, it results in a new solution due to different initial weights and bias values, along with different data for training, validation and for test purposes. The Fig.1 shows the architecture of a NARX network, with input $u(n)$ along with tapped delays on input side, while $y(n+1)$ is the output at time $n+1$, with tapped delays at the output side feedback, and with back propagation.

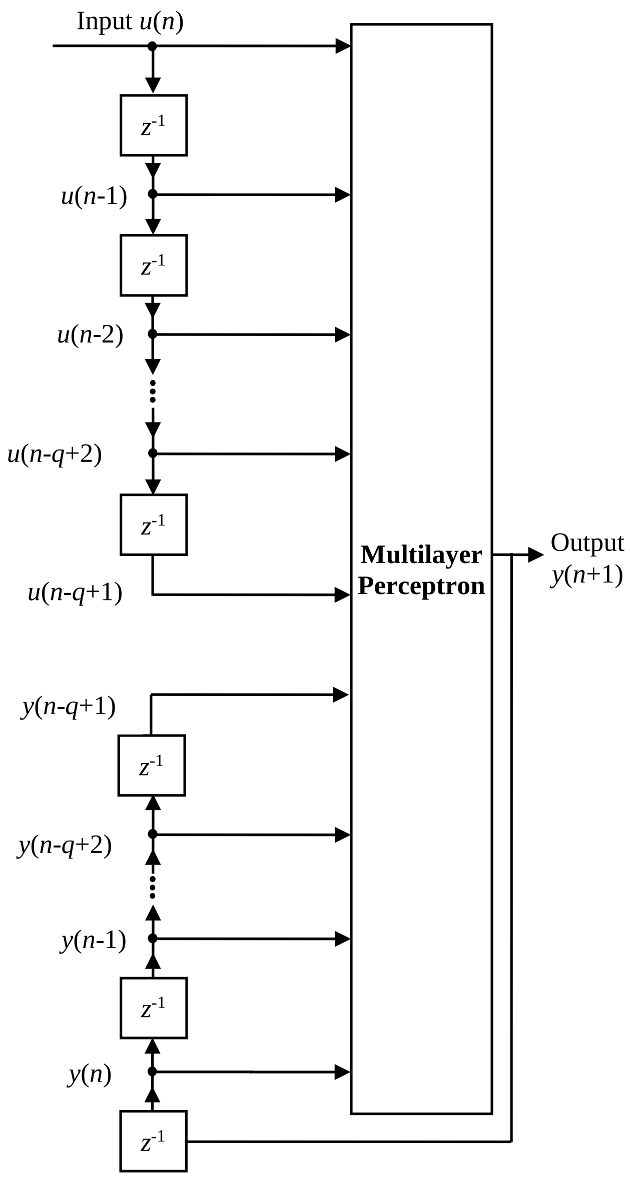

Fig.1. NARX architecture

1.3 ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM (ANFIS) [15]

It is functionally equivalent to a fuzzy inference system, with constraints as in Radial Basis Function Network (RBFN).

A first order Sugeno fuzzy model having two inputs x and y and one output z with a common set of two fuzzy if-then rules

- if x is A_1 and y is B_1, then $f_1 = p_1x + q_1y + r_1$
- if x is A_2 and y is B_2 then $f_2 = p_2x + q_2y + r_2$

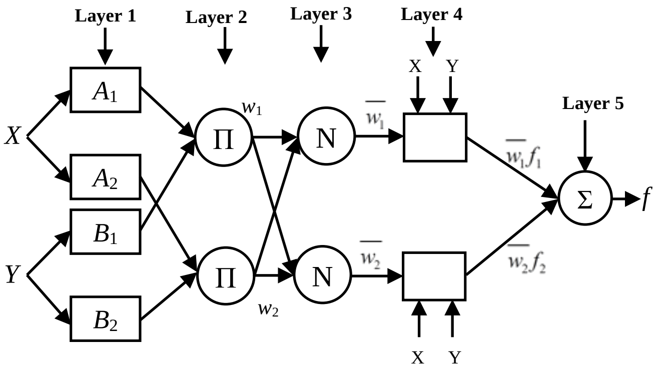

Fig.2. ANFIS architecture

The above architecture consists of similar functions with same layer, as below [15]. Layer 1 node i are adaptive nodes with a function:

$$O_{1,i} = \mu_{Ai}(x),\ \text{for } i = 1, 2 \text{ or } O_{1,i} = \mu_{Bi-2}(y),\ \text{for } i = 3, 4 \quad (13)$$

where, x or y is the input to node i. A_i or B_{i-2} are linguistic labels for the node. $O_{1,i}$ is a membership grade of a fuzzy set A, specifying the degree to which the given input x or y for the quantifier A.

In layer 2, all nodes are fixed and labelled as π, with output being product of the incoming signals

$$O_{2,i} = w_i = \mu_{Ai}(x)\mu_{Bi}(y),\ i = 1, 2 \quad (14)$$

Each output node represents the firing strength of a rule.

In layer 3, all the nodes are fixed that is denoted by N. The i^{th} node calculates the ratio of the i^{th} rule firing strength to the sum of all the rules firing strengths

$$O_{3,i} = \overline{w}_i = \frac{w_i}{w_1 + w_2},\ i = 1, 2 \quad (15)$$

In layer 4, all the i nodes are adaptive with a node function

$$O_{4,i} = \overline{w}_i f_i = \overline{w}_i (p_i x + q_i y + r_i) \quad (16)$$

where, $\overline{w}_i$ is a normalized firing strength from layer 3 and $\{p_i, q_i, r_i\}$being the parameter set of the node.

In layer 5, it has a single node labelled Σ, and computes the overall output as the summation of all the incoming signals

$$\text{Overall Output} = O_{5,i} = \sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad (17)$$

Depending upon the requirements, the layers 3 and 4 are combined, with a four-layer network. Similarly, normalization of weights is performed in the last layer.

2. EXPERIMENT

2.1 WIND SPEED PREDICTION WITH NEURAL NETWORKS

The data required for wind speed prediction for a long duration taken from NREL USA site, with a time recording interval of 10 minutes, with a reset of time after every 24 hours.

2.1.1 Modelling of NARX Neural Network:

In case of application of Lifting Wavelet transform for feature extraction and filtering purpose of the raw data different models were tried and a model with 10 hidden layers with input as data required for prediction and output layer with predicted data has been consideredfor NARX neural network. The Lavenberg-Marquardt (LM) algorithm, which is well suited for small to medium data with training, testing and validation of the model, divided into 50%, 45% and 5% respectively.

The Fig.3 and Fig.4 below show one-month prediction with and without application of Lifting wavelet transform for the month of March. As shown, there is practically no difference in the actual data and predicted values, except in the peaks of the original data reduced and filtered with Lifting Wavelet transform. In case of Fig.4, no filtering and reduction of data is carried out, which is a raw data; but with no change in predicted and actual data.

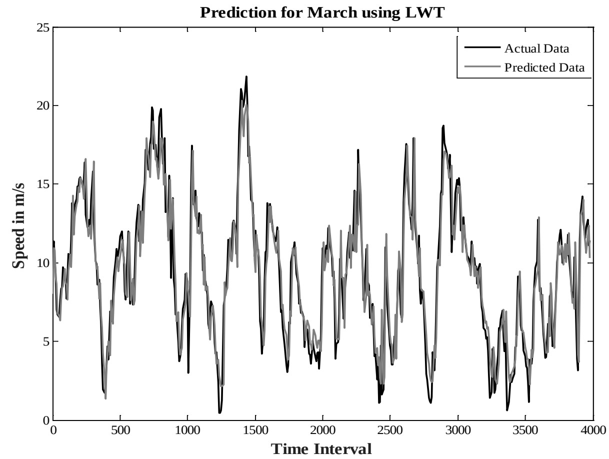

Fig.3. Wind speedPrediction for March month with LWT

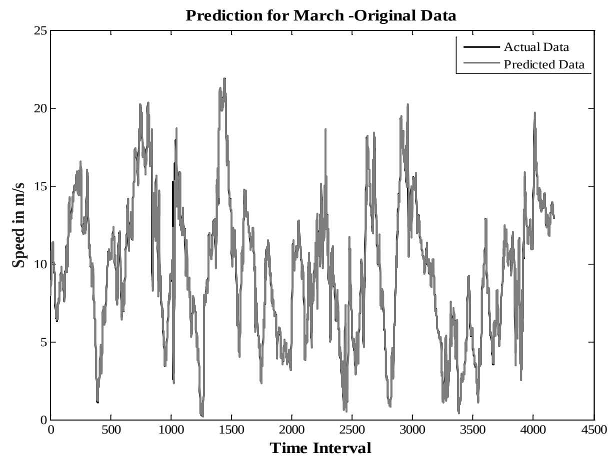

Fig.4. Wind speedPrediction for March month with actual data

Similar is the case for three months' duration prediction, as shown in Fig.5 and Fig.6 below.

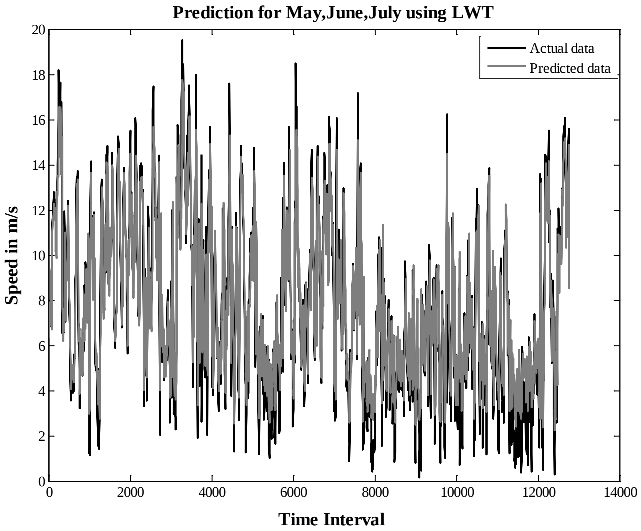

Fig.5. Wind speedPrediction for May,June and Julymonth with LWT

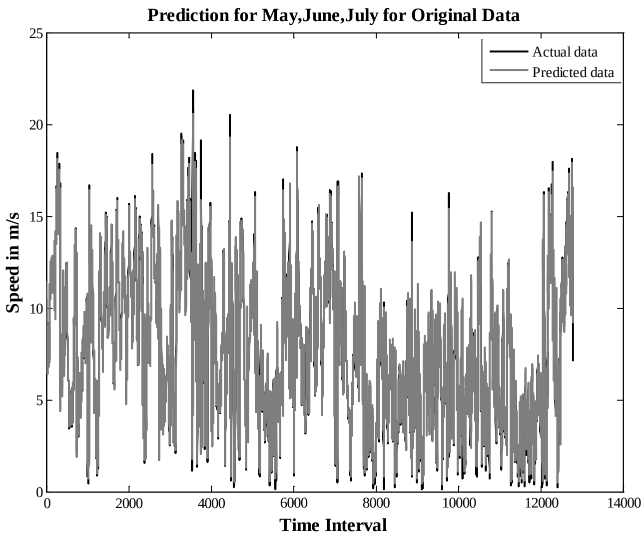

Fig.6. Wind speedPrediction for May, June and July months with actual data

As the data is recorded for more number of months, there is increase in demand on memory; thereby utility in application of wavelet transform is required for feature extraction and in reducing the data.

The Process of training of neural network involves tuning the values of weights and biases for optimising network performance and is defined as the Mean Square Error (MSE) - the average square error '*e*' between the network outputs and target outputs given by,

$$E = MSE = \frac{1}{N}\sum_{i=1}^{N}\left(e_i\right)^2 \tag{18}$$

However, as ANFIS uses Root Mean Square Error (RMSE), which is square Root of MSE, has used for NRAX results also.

Table.1. One-month Performance and RMSE error with Lifting Wavelet Transform and original data using LM algorithm

LWT data	Performance (MSE)	RMSE	Data
Training	2.3919	1.5466	50%
Testing	3.0055	1.7336	45%
Validation			5%
Original data	Performance(MSE)	RMSE	Data
Training	0.1891	0.4348	50%
Testing	0.2169	0.4657	45%
Validation			5%

Table.2. Three-month Performance and RMSE error with Lifting Wavelet Transform and original data with LM algorithm

LWT data	Performance (MSE)	RMSE	Data
Training	3.4267	1.8511	50%
Testing	4.2142	2.0528	45%
Validation			5%
Original data	Performance(MSE)	RMSE	Data
Training	0.426	0.6526	50%
Testing	0.3951	0.6285	45%
Validation			5%

The tables above with the original data reduced by one eight, the performance and RMSE error has not changed much, thus justifying use of Lifting wavelet transform for feature extraction and reducing the memory requirement accordingly. However, even a small increase of error is justified due to bell's 'curse of dimensionality', However, as mentioned above, in case of neural networks, the performance and error changes with change of training data and in change of weights and bias to the neurons, thereby giving different results with every run of algorithm. Normalization of the data is performed to reduce this effect to a certain value.

2.2 WIND SPEED PREDICTION WITH ANFIS

A comparison in performance and error in wind speed prediction with application of ANFIS (Adaptive Neuro-fuzzy Inference System). In this case, the data is divided 50% each for training and testing purpose, as compared to LM algorithm. Different Membership Functions (MFs) are tried and it is found that bell shaped MF performs better in terms of error as compared to triangle and trapezoidal MFs.

Below are the results with ANFIS prediction using Bell MF for with and without lifting wavelet transform for the month of March and for three months (i.e. May, June and July) that shows a comparison with NARX performances. Given below is the performance of Bell MF with and without LWT for one month and three month prediction of wind speed:

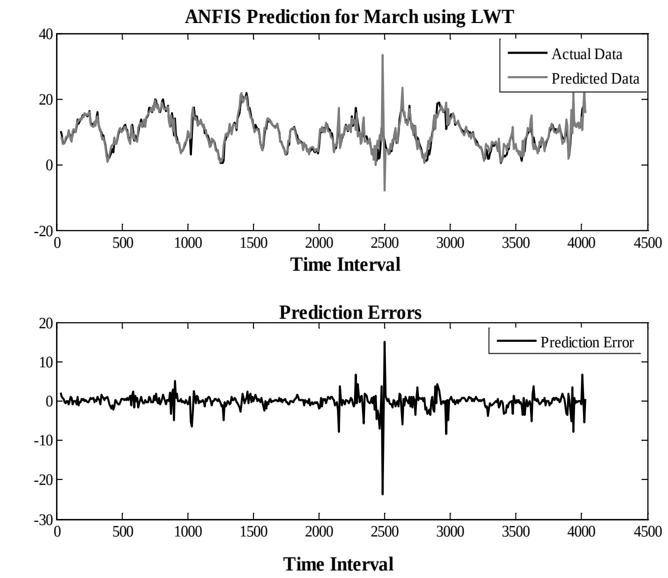

Fig.7. Wind speed Prediction for March month using LWTwith Bell MF

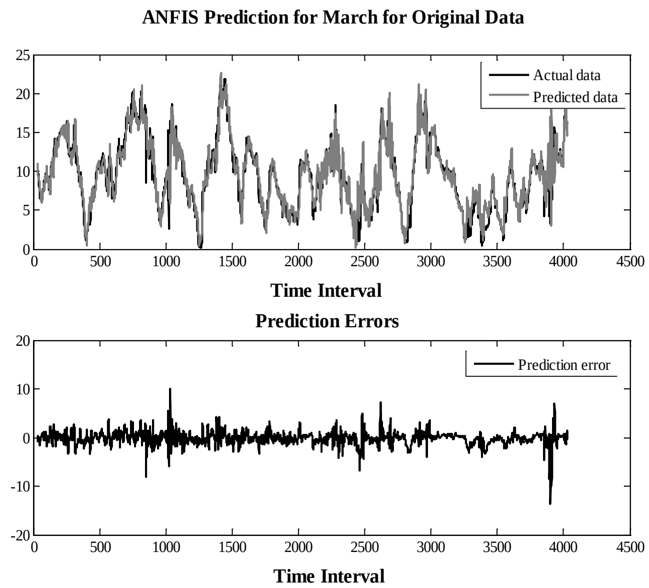

Fig.8. Wind speed Prediction for the month of March withoutLWT and with Bell MF

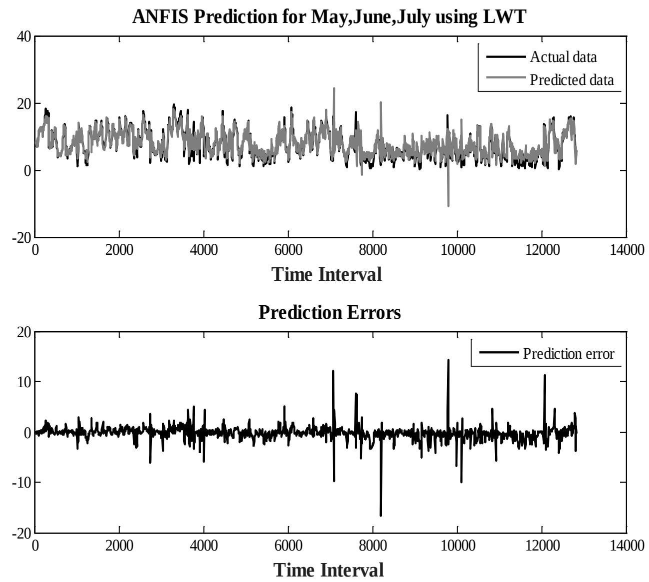

Fig.9. Wind speed Prediction for May, June and July months using LWT and with Bell MF

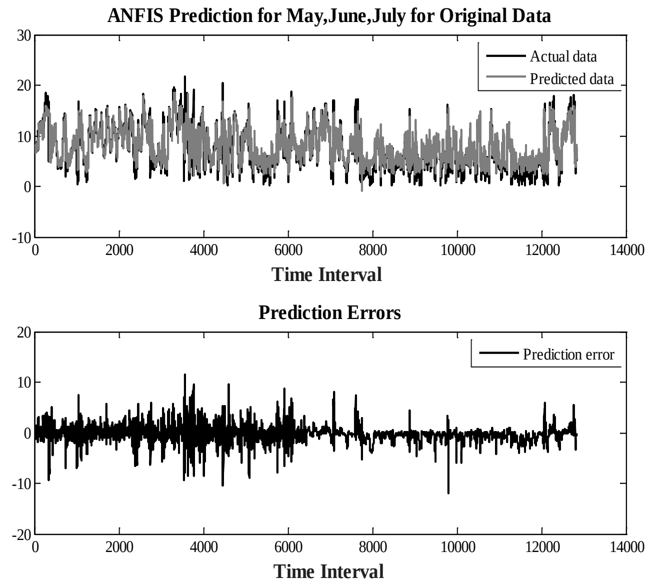

Fig.10. Wind speed Prediction for May, June and July month withoutLWT andwith Bell MF

Below is given the performance of Bell MF with and without LWT for one month and three month prediction of wind speed.

Table.3. One-month Performance and RMSE error with Lifting Wavelet Transform and original data with ANFIS algorithm and Bell MF

Bell MF with LWT	RMSE	Data Division
Training	1.1771	50%
Testing	3.0878	50%
Bell MF without LWT		
Training	1.3958	50%
Testing	2.0410	50%

Below given is the performance for ANFIS with Bell MF and with and without LWT.

Table.4. Three-month Performance and RMSE error with Lifting Wavelet Transform and original data with ANFIS algorithm and Bell MF

Bell MF with LWT	RMSE	Data Division
Training	1.7421	50%
Testing	2.5025	50%
Bell MF without LWT		
Training	1.9013	50%
Testing	2.0846	50%

The above performance with equal number of samples for NARX neural network and ANFIS algorithm with and without LWT is evaluated or comparison purposes.

The above results are inductive, that if the data reduces, the prediction error increases inverse of reduction of data, which meets the Bell's criteria of 'curse of Dimensionality' as stated above. Similarly, as the data for three months is more as compared to one-month data, either with actual or reduced, the prediction performance with more data is better as compared to less data.

The plot of errors with Lifting wavelet transform and original data, show increase in predicted data error with lifting wavelet transform speed, at the near end-point of the time, indicating accumulation of error near the end-point.

3. CONCLUSION

Application of soft computing for chaotic time series data for prediction of long time with quite accurate results, which is very much required for contractual obligations for hybrid smart grids. However, both the soft computing techniques, i.e. neural network and application of ANFIS provide comparative results, but ANFIS approach provides a consistent and better results. Further, a comparison with different MFs for ANFIS, it is concluded that bell shaped MF performs better for the wind speed data.

REFERENCES

[1] Andrew Kusiak, Haiyang Zheng and ZheSong, "Short Term Prediction of Wind Farm Power: A Data Mining Approach", *IEEE Transactions on Energy Conversion*, Vol. 24, No. 1, pp. 125-136, 2009

[2] Jinbin Wen, Xin Wang,Yihui Zheng, Lixue Li, Lidan Zhou, Gang Yao and Hongtao Chen, "Short term Wind Power Forecasting Based on Lifting Wavelet Transforms and SVM", *Proceedings of IEEE Power Engineering and Automation Conference*, pp. 1-4, 2012.

[3] J.L. Torres, A. García, M. De Blas and A. De Francisco, "Forecast of Hourly Averages Wind Speed with ARMA Models in Navarre", *Solar Energy*, Vol. 79, No. 1, pp. 65-77, 2005.

[4] Peiyuan Chen, Troels Pedersen, Birgitte Bak-Jensen and Zhe Chen, "ARIMA-based Time series model for Stochastic Wind Power Generation", *IEEE Transactions on Power Systems*, Vol. 25, No. 2, pp. 667-676, 2010.

[5] Iffat A. Gheyas and Leslie S. Smith, "A Neural Network Approach to Time Series Forecasting", *Proceedings of the World Congress on Engineering*, Vol. 2, pp. 1-5, 2009

[6] Wen-Yeau Chang, "A Literature Review of Wind Forecasting Methods", *Journal of Power and Energy Engineering*, Vol. 2, No. 4, pp. 161-168, 2014.

[7] Richard E. Bellman, "*Adaptive Control Processes: A Guided Tour*", Princeton University Press, 1961.

[8] Wim Sweldens, "The Lifting Scheme: A New Philosophy in Bi-orthogonal wavelet Constructions", Technical Report, Department of Computer Science, Katholieke Universiteit Leuven, pp. 1-12, 1996.

[9] Wim Sweldens, "The Lifting Scheme: A Construction of Second Generation Wavelets", Technical Report, Lucent Technologies, Bell Laboratories, pp. 1-42, 1991.

[10] Wim Sweldens, "The Lifting Scheme: A Custom Design Construction of Bi-orthogonal Wavelets", *Journal of Applied and Computational Harmonic Analysis*, Vol. 3, No. 2, pp. 186-200, 1996.

[11] D.L. Donoho, "Unconditional Basesare Optimal Bases for Data Compression and for Statistical Estimation", *Journal of Applied and Computational Harmonic Analysis*, Vol. 1, No. 1, pp. 100-115, 1993.

[12] Stefan Pittner and Sagar V. Kamarthi, "Feature Extraction from Wavelet Coefficients for Pattern Recognition Tasks", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, No.1, pp 83-88, 1999.

[13] S. Pittner "Dyadic Orthogonal Wavelet Bases and Related Possibilities for Optimal Analysis and Representation of One Dimensional Signals", PhD Dissertation, Vienna University of Technology, 1994.

[14] Simon Haykin, "*Neural Networks: A Comprehensive Foundation*", 2nd Edition, Prentice Hall, 2008.

[15] J.S.R. Jang, C.T. Sun and E. Mizutani, "*Neuro-Fuzzy and Soft Computing-A Computational Approach to Learning and Machine Intelligence*", 1st Edition, Pearson, 1997.