AN INNOVATIVE MODEL FOR HIGH ACCURACY WIND POWER PREDICTION

ALBU R.D., POPENȚIU-VLĂDICESCU F.,
University of Oradea, Universității no.1, Oradea,
ralbu@uoradea.ro

Abstract: This study is another version of the work described in our paper accepted at the PSAM11 & ESREL 2012 international conference where a non-linear ensemble system is used to develop a new model for predicting wind speed in short-term time scale. Regardless of the recent advancements in the re-search of prediction models, it was observed that different models have different capabilities and also no single model is appropriate under all situations. The idea behind EPS (ensemble prediction systems) is to take advantage of the features of each subsystem to confine diverse patterns that exist in the dataset. The low prediction errors demonstrate the increased prediction accuracy.

Keywords: ensemble**,** wind-power, short-term, prediction, forecast, recurrent neural networks, wind energy.

1. INTRODUCTION

Wind power prediction is of vast importance for the safety and stabilization of grids (Moyano et al. 2009). The most intricate problem now is to improve the prediction accuracy. BP (Backward Propagation) neural networks have been used extensively in wind power prediction but results have shown slow convergence rate (Gaofeng et al. 2008).

In the technical literature, we can find two major approaches to forecast wind power:

1. Physical methods: Require many physical considerations to gain the best prediction precision. The input variables will be physical or meteorology information. They present advantages in long-term prediction

2. Statistical methods: Aspire at finding a relationship between the on-line measured power data. They will use the historical data of the wind farm. They present advantages in short-term prediction.

Ensembles are prediction techniques used to produce a representative sample of a dynamic system possible future state (Raj Kiran et al. 2011). Sometimes the EPS may use different models for different members, or different formulations of a model.

The multiple simulations show two main sources of uncertainty in prediction models

- Errors introduced by chaos or sensitive dependence on the initial conditions.
- Errors introduced because of imperfections in the model.

This article is structured as follows: we focus on the proposed EPS and its architecture. The next section contains numerical results from a real world case study, particularly our EPS prediction results. We tested the proposed EPS using data sets collected from the ANM (National Meteorology Administration) website. In the last section, we raise some interesting conclusions.

2. THE PROPOSED ENSEMBLE PREDICTION SYSTEM

The time scales we use in this short-term prediction solution are in the order of some days for the forecast horizon and from minutes to hours for the time-step. For the purpose of time series prediction, an ensemble can be considered to be a general nonlinear mapping between a subset of the past time series and the future time series values. The proposed EPS is presented in Fig. 1. In the following sections we will briefly describe each prediction model. For more details about their architecture and function see the referred papers.

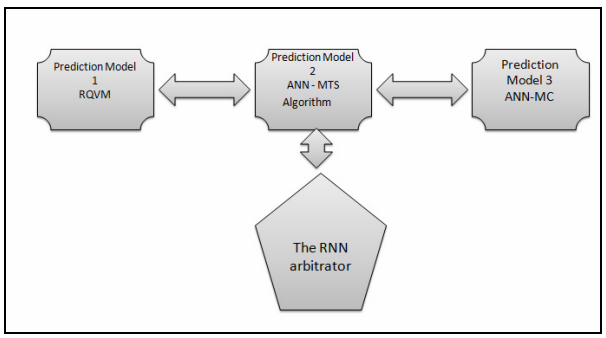


Figure 1. The EPS

2.1. The RQVM

The RQVM (Recurrent Quadratic Volterra Model) is similar to the one proposed by Duehee Lee in 2011. His paper shows a way to use the recurrent quadratic Volterra system to predict the wind power. The RQVM is a second-order polynomial equation that uses the output data as feedback recursively, after passing a time-delay

filter. The Volterra system is extracted from the weights of the Recurrent Neural Network. In order to build Volterra kernels from the combination of weights, the activation function is approximated to the high order polynomial function using the Lagrangian interpolation. The memory of the Volterra system is identified using the PACF Partial Autocorrelation Function. The Volterra system can analyze order and memory and it captures the output power patterns that can be used for short-term prediction. For simplicity, it is assumed that the system is causal and homogeneous. The truncated recurrent Volterra system *V (P, M)* with the finite order of *P* and the finite memory of *M* is defined as.

$$x(n) = h_0 + \sum_{k_1=1}^{M} h_1(k_1)x(n-k_1) + \sum_{k_1=1}^{M}\sum_{k_2=1}^{M} h_2(k_1,k_2)x(n-k_1)x(n-k_2) + \sum_{k_1=1}^{M}\cdots\sum_{k_p=1}^{M} h_p(k_1,\ldots,k_p)x(n-k_1)\ldots x(n-k_p) + e(t)$$

$$n = M+1, M+2, \ldots \quad (1)$$

Where *h0* is the constant, which is zero in this case, and hi (*k1, ... ki*) is the set of Pth order Volterra kernel coefficients. Consecutively, to reduce the computational complexity, the kernels should be assumed as:

$$h_p(k_1, \ldots k_p) = 0 \text{ if } \quad k_1 > \ldots > k_p \quad (2)$$

The kernels hi are assumed to be a symmetric function with respect to all permutations of the indices $k_i, \ldots, k_P$. As a result, one kernel per each permutation is enough to describe the Volterra system. The other kernels become zero. The input signal is $X_{n-1} = \{x(n-1), x(n-2), \ldots, x(n-M)\}$, and the output signal is the x(n). All signals and kernels are real numbers. Since it is difficult to categorize errors in the truncated Volterra model as natural system errors or higher order term errors, the author assumed higher order terms are absorbed into the Volterra system errors. The number of Volterra kernels of each order can be calculated by using the Combination with a Repetition. The equation is as follows:

$$\frac{(M+P)!}{P!(M)!} \quad (3)$$

The order is limited to the second order, and the memory is limited to 20. The RNN in this model has three layers: one input layer, one hidden layer, and one output layer. The input layer receives as many input neurons as the number of memory, and input neurons are in the tap-delayed form. One hidden layer has an arbitrary number of neurons which is heuristically decided by the number of Volterra kernels. Hidden layers receive"net input" which is the sum of the input values that are multiplied by their corresponding weights. Hidden neurons use the tangent hyperbolic function *tanh* as the activation function. The output layer has only one neuron and uses a linear activation function.

2.2. ANN Prediction Model Based on MTS Algorithm

As the second model in our ensemble system we used a model analogous with the one described by Shuang Han et al. 2011. They have put forward a wind power prediction model build with a BP neural network optimized by Tabu search algorithm with memory function. The basic principle of ANN (Artificial Neural Network) based on MTS (Multiple Tabu Search) algorithm is to optimize neural network's connection weights using TS algorithm which has memory function. Achieve the global optimal solution using the global search capability of TS algorithm and thus avoid getting into local minimal.

Suppose the error function of some BP network is: $f = f(W_h, W_o, \Theta_h, \Theta_o)$ where:

- W_h, are connection weights between input layer and hidden layer.
- W_o, are connection weights between hidden layer and output layer.
- Θ_h, is the threshold value of hidden layer neurons.
- Θ_o is the output layer neurons.

The optimization for network is the process of solving *min (f (Wh, Wo, Θh, Θo))*. For the expediency of representation, symbol Δ is used to denote vector $(W_h, W_o, \Theta_h, \Theta_o)$.

The following is the procedure of optimizing neural network with TS algorithm:

1. Initialize Δ; endow every component of Δ with a little random number denoted $\Delta_{initial}$.
2. Δ_{best} denotes the optimal solution and Δ_{now} denotes the current solution. $\Delta_{best} = \Delta_{initial} = \Delta_{now}$. The vector Δ_{now} is storied in the Tabu table.
3. Produce a neighbourhood solution Δ_{new} of $\Delta_{initial}$ and calculate f (Δ_{new}) and f (Δ_{best}).
4. If f (Δ_{new}) has not varied for many times continuously n stop the algorithm and output the result, else go on with the next step.
5. If f (Δ_{new}) < f (Δ_{best}) then $\Delta_{best} = \Delta_{new}$. The vector Δ_{new} enters Tabu table and the memory point in table backward in turn. If f (Δ_{new}) >= f (Δ_{best}) we need to judge if Δ_{new} is within some memory point's given neighbourhood. If it is a neighbourhood solution vector Δ_{new} is reconstructed. If it isn't then $\Delta_{now} = \Delta_{new}$ and update the Tabu table at the same time.
6. Produce a neighbourhood solution Δ_{new} of Δ_{now} and go to step (4).
7. The optimized weight vectors and threshold value vectors are obtained when the training finished.

ISSN 2067-5534 © 2012 JSE

The inputs of the BP network are: wind speed, wind direction, air temperature, air pressure, relative humidity. Less neurons in the hidden layer result in worse prediction precision. On the contrary, over fitting may occur with too many hidden neurons and this result in poor prediction precision and long training time. The common method is to adjust it in training process. The optimal number is corresponding to the minimal error.

2.3 Artificial neural network–Markov chain model

In the model proposed by S.A. Pourmousavi Kani et al. 2011, artificial neural network (ANN) and Markov chain (MC) are used to develop a new ANN–MC model for predicting wind speed in very short-term time scale. In this study, the short-term patterns in wind speed data are captured by ANN and the long-term patterns are considered utilizing the MC approach and four neighbourhood indices. The proposed model consists of two ANNs. The first one, ANN-1, is used for short-term wind speed prediction. ANN-1 is a multi-layered perceptron (MLP) that consists of one input layer, one hidden layer, and one output layer that has only one neuron. This step is called primary prediction by the author. This network has 10 inputs fed with actual wind speeds corresponding to times *t* to *t-10*. The training data consists of 30 sets with 10 measured wind speeds in each set. After the primary prediction, TPs (transition probabilities) for predicted values, other four indices and primary prediction outputs are fed as input variables to a second ANN (ANN-2). Finally, the constructed model is used for different time horizon predictions. Regarding the Markov process, the probability of the given condition in the given moment may be deduced from information about the previous conditions. The order of the chain gives the number of time steps in the past influencing the probability distribution of the present state, which can be greater than one. Many natural processes are considered as Markov processes. Actually, the TPM (transition probabilities matrix) is a tool for describing the MC activities. Each element of the matrix represents probability of moving from a specific condition to a next state. TPM is formed by 600 preceding wind speed data and the calculated matrix is used for primary predicted values. Initially they calculate the Markov state for primary predicted values, the outputs of ANN-1, for one step ahead. Then, according to TPM, the probability of predicted value in the next step is calculated. This process is executed for all primary predictions. For longer prediction horizon, transition probabilities for two or three steps ahead are necessary. In these cases, the above TPM is multiplied according to the number of time steps in the future. It is observed that there is a logical relation between the states for predicted values in comparison with the states of actual values.

The relations between the primary prediction results and the coefficients obtained from MC are difficult to be established. Since ANNs can encode complex and non-linear associations, the ANN-2 is used to discover the relations between the primary prediction values and the obtained probabilities. The inputs of the ANN-2 are: the transition probability of the predicted values' state, transition probabilities toward two next states (FNIs) and transition probabilities toward two backward states (BNIs). The main purpose of ANN-2 is to achieve higher accuracy of prediction in contrast with primary predicted values. Since the ANN-2 has six input variables and one output variable, number of neurons in each layer should be in the range of variables. The best structure for the ANN-2 with the least MAPE (Mean absolute percentage error) is determined by the author as 3, 0 and 1 neurons for input, hidden and output layers, respectively, with 10 training vectors and the learning rate of 0.01–0.05.

2.4 The arbitrator

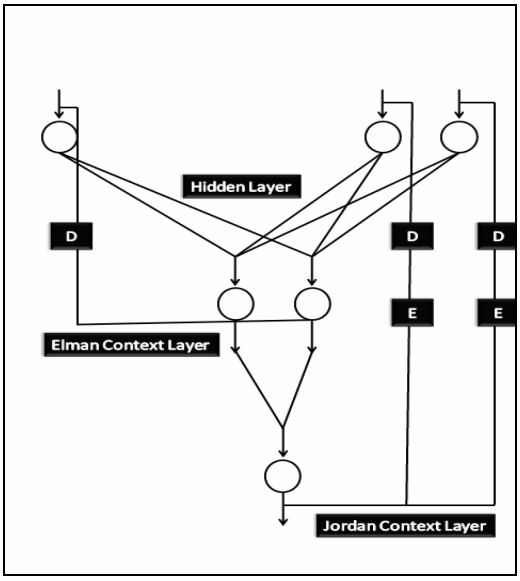


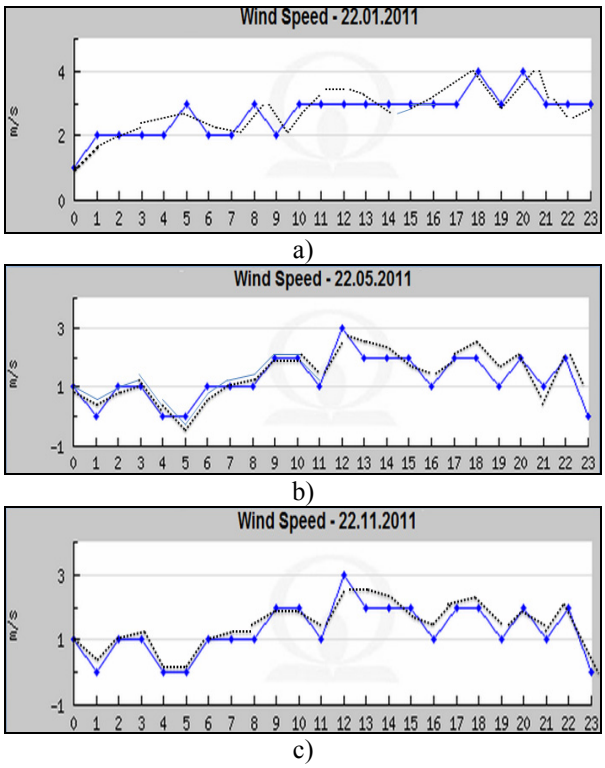
Figure 2. The arbitrator

The ensemble uses the output obtained from the individual constituents as inputs to it and the data is processed according to the design of the arbitrator. As the arbitrator in our approach we use the following RNN shown in (Fig. 2).

The RNN has two context layers: the Elman context layer and the Jordan context layer, both with some differences from the original Elman and Jordan recurrent neural networks. The Elman context layer differs from the original Elman RNN because the two context neurons obtain inputs from the output of the hidden layer after a delay of one time unit. In Elman context layer the number of neurons must match the number of neurons in hidden layer. A Jordna RNN has a number of neuron in context layer that matches the number of neurons in output layer. Another difference is that in a Jordan network the output is used to feed the context layer through a non-weighted connection and the context layer is going to feed to the hidden layer just as in Elman networks.

3. PREDICTION RESULTS

We tested the proposed EPS using data sets collected from the ANM (National Meteorology Administration) website.



a)

b)

c)

Figure 3. Prediction results a) a winter day b) a summer day c) a fallen day

We randomly selected three days, one from each season, and predict the wind speed at each hour for these days. Our EPS results are illustrated in (Fig. 3). The solid line represents the measured wind speed and the dotted line indicates the predicted values. The wind speeds were measured in Tulcea and collected from the ANM web site. To predict *W (d, h),* the wind speed at hour h of day d, we train the RNN with only last two values: *W (d-1, h), W (d-2, h*); it has been reported that every data point in a time series is only strongly dependent on the immediate past two values. The training is complete when we provide as inputs all wind speed values, for a number of *n* epochs. One epoch is finished when the entire training set is exposed to the RNN. The number of epochs is the number of steps of the training process, it is a dynamic value; we set it high and let it stop according to the validation set. The initial learning rate is 0.001, results in good coarse training quickly. For better performance, we used a schedule of 0.0005 for two epochs, followed by 0.0002 for the next three, 0.0001 for the next three, 0.00005 for the next four, and 0.00001 thereafter. The learning rate is decreased by 79.4% of its value after every epoch. In order to implant fixed points into recurrent systems, the backpropagation technique is used. In fixed-point learning, the first action is the forward propagation of the activations. This procedure repeated for a certain number of times will induce the relaxation period. This has to be repeated until the network attains its own dynamic. After the net become stable, an error can be computed at the output. Then, the error is propagated backwards through the network. The error at each output can be multiplied by the relaxed activation for updating the weights. We have to select the relaxation time both in the forward and backpropagation phases.

In (Fig. 4) is presented a comparison between the prediction accuracy of the ensemble subsystems and the EPS, at different time-horizons. As we can see in the, we obtained very good prediction results proven by a very low average error rate. The secret is the joint usage of Neuro Solutions features and our innovative EPS architecture.

The prediction error of a model is classically defined as the difference between the measured and the predicted value. A horizon dependent model error *e (t + k|t)* is given by:

$$e\,(t+k|t) = v\,(t+k) - vp\,(t+k!t) \qquad (4)$$

Where *v (t + k)* is the measured wind speed at time *t + k*, and *vp (t+k!t)* is the wind speed predicted for time *t + k* and computed at time *t*. The evaluation criterion we used is the MAPE defined as:

$$MAPE\,(k) = \frac{1}{N}\sum_{t=1}^{N}\left(\left|\frac{e(t+k\,|\,t)}{v(t+k)}\right| X100\right) \qquad (5)$$

Also, *k* and *N* represent the prediction horizon and number of prediction respectively. In this study for each hour we have done 30 predictions, so N=30.

ISSN 2067-5534 © 2012 JSE

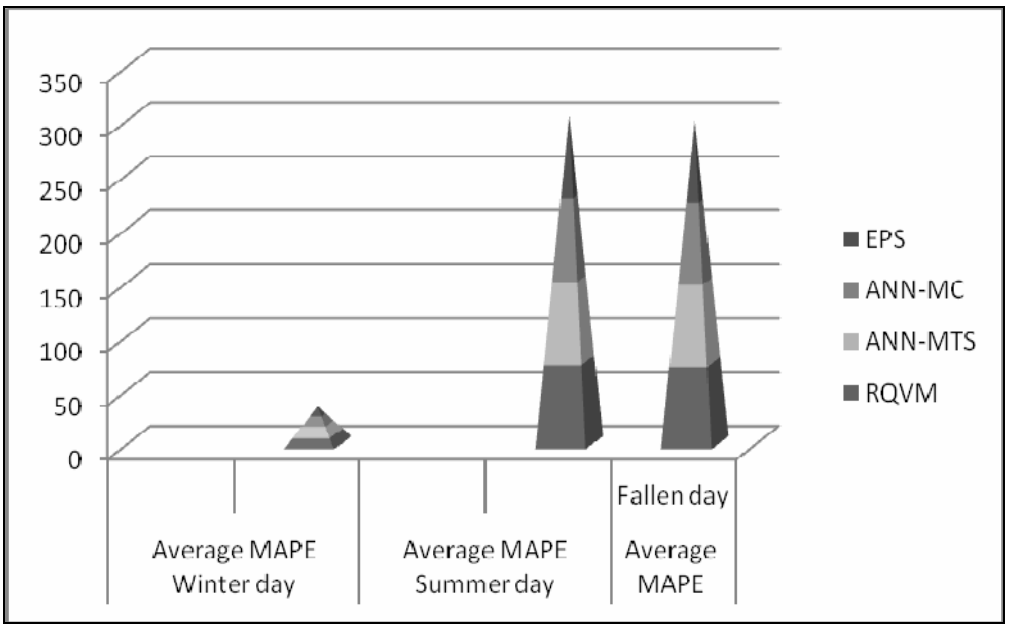


Figure 4. MAPE at different time-horizons for each ensemble subsystem

Table 1. Best prediction errors and MAPE

Hour	Winter day 22.01.2011	Summer day 22.06.2011	Fallen day 22.10.2011
0	+0.01 / 1.35	-0.01/ 1.2	+0.01/ 1.12
1	-0.23 / 11.5	+0.4 / 403	+0.4 /403
2	-0.02 / 1.06	-0.13 / 13.13	+0.02 / 2.2
3	+0.25 / 12.5	-0.003 / 0.35	+0.1 / 98
4	+0.79 / 39.5	+0.17 / 167	+0.02 / 0.2
5	-0.65 / 21.6	-0.35 / 305	+0.02 / 0.22
6	-0.24 / 12	-0.15 / 147	+0.01 /1.55
7	+0.06 / 2.96	-0.13 / 132	+0.1 / 107
8	-0.02 / 0.66	+0.1 / 9.69	+0.2 / 203
9	+0.34 / 17	-0.3 / 15	-0.01 / 0.55
10	-0.19 / 6.33	+0.002 / 0.1	-0.01 /0.55
11	+0.01 / 0.33	+0.3 / 31	+0.5 / 508
12	+0.19 / 6.35	-0.32 / 11.28	-0.6 /22
13	+0.12 / 4,04	+0.85 / 42.6	+0.7 / 36
14	-0.02 / 0.77	+0.32 / 18	+0.4 / 19.9
15	-0.01 / 0.33	-0.01 / 0.52	-0.1 / 5.6
16	+0.13 / 4.43	+0.42 / 43	+0.65 /66
17	+0.8 /28.65	-0.001 / 0.06	+0.18 /9
18	+0.03 / 0.75	+0.62 / 32	+0.21 / 10.52
19	-0.1 / 3.35	+0.79 / 80.5	+0.32 / 33
20	-0.21 / 5.52	+0.001 / 0.05	-0.03 / 1.53
21	+0.05 / 1.66	-0.23 / 23.2	+0.2 /20.1
22	-0.21 / 7.22	+0.03 / 1.5	+0.16 / 8
23	+0.01 / 0.39	+0.34 / 339	+0.21 /230

Table 2. Average MAPE for each ensemble subsystem

Model	Average MAPE Winter day	Average MAPE Summer day	Average MAPE Fallen day
RQVM	8.73	76.18	74.57
ANN-MTS	8.91	75.92	74.98
ANN-MC	8.12	76.09	74.68
EPS	7.92	75.64	74.46

4. CONCLUSIONS AND FUTURE WORK

A repeatedly scenario appears when new prediction results are presented: the new model is argued to have higher predictive accuracy than do other prediction models. This situation shares the issue that predictive accuracies are being calculated and compared in different test sets. The prediction model may have inherent accuracy, but the accuracy as measured will vary across test sets. This makes it impossible to define the accuracy of a prediction model independent of the test set to which it is applied. Basically, no real-world prediction model can foresee perfectly. There are four major reasons for this.

1. The prediction model may not have one or more required variables: we simply do not know everything necessary to predict with perfect accuracy.
2. Measurement error can reduce accuracy: a poor data collection phase introduces noise in the data set and damage predictive accuracy.
3. The prediction model may not fit as well as it could: a predictor might be forced to have a linear outcome when a nonlinear result would have fit the data better and would permit more precise predictions.
4. The effective sample size may be unsatisfactory to approximate the prediction model coefficients as accurately as possible.

Consequently, the new prediction models must be compared to existing models on the same data sets if we want to judge whether progress has been made or not.

ACKNOWLEDGEMENTS

This work was partially supported by the strategic grant POSDRU/CPP107/DMI1.5/S/80272, Project ID80272 (2010), co-financed by the European Social Fund-Investing in People, within the Sectorial Operational Program Human Resources Development 2007-2013.

REFERENCES

[1]. Henrik Madsena, Răzvan-Daniel Albu, Ioan Felea, Florin Popenţiu-Vlădicescu, (2012) "A New Ensemble Model for Short Term Wind Power Prediction", accepted at PSAM11 & ESREL 2012 international conference.

[2]. Barbounis T. and Theocharis J., (2007) Locally recurrent neural networks for wind speed prediction using spatial correlation, *Information Sciences,* vol. 177, no. 24, pp. 5775 – 5797.

[3]. Costa A., Crespo A., Navarro N., Lizcano G., Madsen H. and Feitosa E., (2008) A review on the young history of the wind power short-term prediction, *Renewable and Sustainable Energy Reviews*, vol. 12, no. 6, pp. 1725-1744, June 2008.

[4]. Duehee L., (2011), Short-Term Prediction of Wind Farm Output Using the Recurrent Quadratic Volterra Model, *IEEE Trans. Reliability*, vol. 58, no. 1, pp. 106–117.

[5]. Gaofeng F., Weisheng W., Chun L., Huizhu D. (2008) Wind power prediction based on Artificial Neural Network, *Proceedings of the CSEE*, vol. 28, no. 34, pp. 118-123, Dec. 2008.

[6]. Gebraeel N., & Pan, J., (2009), Prognostic degradation models for computing and updating residual life distributions in a time-varying environment, *IEEE Trans. Reliability*, vol. 57, no. 4, pp. 539–550.

[7]. Montgomery D. C. and Murat K. (2008), Introduction to Time Series Analysis and Forecasting, D. J. B., N. A. C., and G. M. F., *Eds. Wiley series in probability and statistics.*

[8]. Moyano C, F., Pecas L. J. A. (2009) An optimization approach for wind turbine commitment and dispatch in a wind park. *Electric Power Syst Res,* 79(1):71–9.

[9]. Negnevitsky M., Mandal P., and Srivastava A., (2009) An overview of forecasting problems and techniques in power systems, in *Power Energy Society General Meeting*, PES '09. IEEE, 2009, pp. 1 –4.

 ISSN 2067-5534 © 2012 JSE