



Missouri University of Science and Technology  
Scholars' Mine

---

Electrical and Computer Engineering Faculty  
Research & Creative Works

Electrical and Computer Engineering

---

01 Jan 2014

## Hidden Markov Model with Information Criteria Clustering and Extreme Learning Machine Regression for Wind Forecasting


Dao Lam

Shuhui Li

Donald C. Wunsch

*Missouri University of Science and Technology*, [dwunsch@mst.edu](mailto:dwunsch@mst.edu)

Follow this and additional works at: [https://scholarsmine.mst.edu/ele\\_comeng\\_facwork](https://scholarsmine.mst.edu/ele_comeng_facwork)

 Part of the [Electrical and Computer Engineering Commons](#), and the [Numerical Analysis and Scientific Computing Commons](#)

---

### Recommended Citation

D. Lam et al., "Hidden Markov Model with Information Criteria Clustering and Extreme Learning Machine Regression for Wind Forecasting," *Journal of Computer Science and Cybernetics*, vol. 30, no. 4, pp. 361-376, Vietnam Academy of Science and Technology, Jan 2014.

The definitive version is available at <https://doi.org/10.15625/1813-9663/30/4/5510>

This Article - Journal is brought to you for free and open access by Scholars' Mine. It has been accepted for inclusion in Electrical and Computer Engineering Faculty Research & Creative Works by an authorized administrator of Scholars' Mine. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact [scholarsmine@mst.edu](mailto:scholarsmine@mst.edu).

# HIDDEN MARKOV MODEL WITH INFORMATION CRITERIA CLUSTERING AND EXTREME LEARNING MACHINE REGRESSION FOR WIND FORECASTING

DAO LAM<sup>1</sup>, SHUHUI LI<sup>2</sup>, AND DONALD WUNSCH<sup>1</sup>

<sup>1</sup>*Department of Electrical & Computer Engineering, Missouri University of Science & Technology; dlmg4, dwunsch@mst.edu*

<sup>2</sup>*Department of Electrical & Computer Engineering, The University of Alabama; sli@eng.ua.edu*

**Abstract.** This paper proposes a procedural pipeline for wind forecasting based on clustering and regression. First, the data are clustered into groups sharing similar dynamic properties. Then, data in the same cluster are used to train the neural network that predicts wind speed. For clustering, a hidden Markov model (HMM) and the modified Bayesian information criteria (BIC) are incorporated in a new method of clustering time series data. To forecast wind, a new method for wind time series data forecasting is developed based on the extreme learning machine (ELM). The clustering results improve the accuracy of the proposed method of wind forecasting. Experiments on a real dataset collected from various locations confirm the method's accuracy and capacity in the handling of a large amount of data.

**Keywords.** Clustering, ELM, forecast, HMM, time series data.

## 1. INTRODUCTION

The importance of time series data has established its analysis as a major research focus in many areas where such data appear. These data continue to accumulate, causing the computational requirement to increase continuously and rapidly. The percentage of wind power making up the nation's total electrical power supply has increased quickly. Wind power is, however, known for its variability [1]. Better forecasting of wind time series is helpful to operate windmills and to integrate wind power into the grid [2, 3].

The simplest method of wind forecasting is the persistence method, where the wind speed at time ' $t + \Delta t$ ' is predicted to be the same speed at time ' $t$ '. This method is often considered a classical benchmark. Such a prediction is of course both trivial and useless, but for some systems with high variability it is challenging to provide a meaningful forecast that outperforms this simple approach. Another more useful example of a classical approach is the Box-Cox transform [4], which typically is used to approximate the wind time series to Gaussian marginal distribution before using the autoregressive-moving-average (ARMA) model to fit the transformed series. However, ARMA models are often outperformed by neural network based methods [5], [6], which represent the approach mentioned in this paper.

The forecasting of time series data using neural networks has been researched on widely [7, 8] due to the ability of neural networks to learn the relationship between inputs and outputs non-statistically and their lack of a requirement for any predefined mathematical models. Many wind

forecasting methods have used this approach, including [9, 10]. However, training the network takes a long time due to slow convergence. The most popular training method is backpropagation, but it is known to be slow in training, additionally, its wind forecasting performance, in general, has not been as successful as other applications of backpropagation [8]. Radial basis function (RBF) trains faster but with high error and can not handle a large amount of data due to the memory requirement for each of the training samples. The adaptive neuro-fuzzy interface system (ANFIS) predictor [11] is a fuzzy logic and neural network approach that improves on the persistence method but is still limited in terms of speed when working with large data sets.

A more successful clustering approach is the hidden Markov switching model. In [12], hidden Markov switching gamma models were used to model the wind in combination with additional information. Such approaches, however, have not used clustering techniques to group the data to the same model. Recently, [1] proposed a two-step solution for wind power generation. First, mean square mapping optimization was used to predict wind power, and then adaptive critic design was used to mitigate wind power fluctuations.

Wind speed trends change over time. Therefore, to understand the nature of wind currents, a stochastic model must be built for wind time series. Several approaches have been used in times series data analysis, the most popular of which is the hidden Markov model (HMM) [12]. However, HMM parameter estimation is known to be computationally expensive, and with such a large sequence of National Oceanic & Atmospheric Administration (NOAA) data used to model the wind, the current approaches remain unable to accomplish such estimation.

The goal of this paper is to present an effective solution for forecasting the wind time series, which is achieved by first clustering the time series data using HMM, and then using the clustering results in the extreme learning machine predictor. Therefore, this paper makes valuable contributions. From the clustering perspective, a novel method of clustering time series data is proposed that uses HMM with modified information criteria (MIC) to identify the wind time series clusters sharing the same dynamics. The paper offers the following new features to clustering using HMM: first, it provides a mechanism for handling sequential data that are simultaneously continuous and discrete; second, it proposes a method that probabilistically determines the HMM size and partition to best support clustering; and third, it makes use of the power of the Hidden Markov Model ToolKit (HTK) [13] engine, an open-source speech processing toolkit provided by Cambridge University, to induce the HMM from the time series. One of the primary advantages of the presented method compared to others is its ability to handle a large amount of time series data by leveraging HTK for HMM clustering and the extreme learning machine (ELM) to obtain the analytic solution when training the neural network. Moreover, to forecast wind, a new method for wind time series data forecasting is developed herein based on ELM. The clustering results improve the accuracy of the proposed wind forecasting method.

The paper is organized as follows. Sec. 2. provides a brief review of ELM, model selection and related work. Then, the proposed framework for wind forecasting is presented in Sec. 3.. Next, in Sec. 4., the experiment is demonstrated on real data to confirm the success of the clustering approach in clustering. Sec. 5. details the performance of the approach during different seasons and forecast horizons. Finally, Sec. 6. concludes the paper with for future work.

## 2. BACKGROUND AND RELATED WORK

### 2.1. Model Selection

From probabilistic mixture model-based selection, it is known that model selection involves finding the optimum number of mixtures by optimizing some criterion. In model-based clustering, mathematical models represent a cluster’s structure, and model parameters are chosen that best fit the data to the models. Several criteria have been investigated in the literature, including the Akaike information criterion (AIC) [14] and the Bayesian information criterion (BIC) [15]. In general, no criterion is superior to any other, and criteria selection remains data-dependent.

In HMM clustering, BIC is often used for model selection, e.g., [15–17]. The basic definition of a BIC measure given a model  $\lambda$  and data  $X$  is [15]:

$$BIC = \log\{P(X|\lambda, \hat{\theta})\} - \frac{d}{2}\log(L) \tag{1}$$

where  $d$  is the number of independent parameters to be estimated in the model.  $L$  is the number of data objects, and  $\hat{\theta}$  is the parameter estimation of the model  $\lambda$ .

Similarly, the AIC measure [14] is given as:

$$AIC = \log\{P(X|\lambda, \hat{\theta})\} - d \tag{2}$$

Choosing parameters that maximize the criteria allows the best-fitting model to be selected. In both equations,  $\log\{P(X|\lambda, \hat{\theta})\}$ , which is the data likelihood, increases as the model becomes bigger and more complicated, whereas the second term, which is the penalty term, favors simple, small models. For extended series such as wind data, computing  $\log\{P(X|\lambda, \hat{\theta})\}$  often requires a lot of time. This challenge is met by using the HTK Toolbox in this paper.

A comparison of (1) and (2) reveals a difference in the penalty term. Moreover various forms of BIC measures have been applied successfully in many clustering applications [18].

In addition to the problem of defining the model, HMM clustering also faces the problem of cluster validity, as do other clustering techniques [19]. In the model selection, some existing criteria, techniques, and indices can facilitate the selection of the best number of clusters. This paper follows Bayesian information criteria, which uses the best clustering mixture probability:

$$P(X|\lambda) = \prod_{i=1}^L \sum_{k=1}^K P_k * P(x_i|\lambda_k) \tag{3}$$

where  $X$  is the dataset,  $\lambda_k$  is the model of cluster  $k$ ,  $x_i$  is the  $i^{th}$  data point in dataset  $X$ ,  $P_k$  is the likelihood of  $x_i$  in cluster  $k$ , and  $L$  and  $K$  are the number of data points and clusters, respectively.

### 2.2. Extreme learning machine (ELM)

ELM is a feed forward single hidden layer neural network that can approximate any nonlinear function and provide very accurate regression [20, 21]. The most advantageous feature of ELM, however, is the way it is trained. Unlike other neural networks that take hours, or even days to train because of their slow convergence, ELM input weights can be initialized randomly, and ELM output weights can be determined analytically by a pseudo inverse matrix operation [21].

Let  $\mathbf{X} \in R^{n \times N} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]$  be  $N$  data used to train the ELM. To take the bias value of the neuron,  $\mathbf{X}$  is transformed into  $\hat{\mathbf{X}}$  by adding a row vector of all 1s, i.e.  $\hat{\mathbf{X}} = \begin{bmatrix} \mathbf{X} \\ \mathbf{1} \end{bmatrix}$ .

Denote the expected output of the ELM  $\mathbf{T} \in R^{k \times N} = [\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_N]$ .

Denote  $\mathbf{W}_i \in R^{N_H \times n}$  and  $\mathbf{W}_o \in R^{k \times N_H}$  as the input weight matrix and output weight matrix of ELM where  $N_H$  is the number of neurons in the hidden layer.

Doing so yields

$$\mathbf{H} = g(\mathbf{W}_i * \hat{\mathbf{X}}) \quad (4)$$

where  $\mathbf{H} \in R^{N_H \times N}$  is the hidden layer output matrix of ELM and  $g$  is the nonlinear activation function of the neuron.

Once  $\mathbf{H}$  is obtained, the output of the output layer can be calculated

$$\mathbf{O} = g_2(\mathbf{W}_o * \mathbf{H}) = \mathbf{W}_o * \mathbf{H} \quad (5)$$

(5) occurs because the output node activation function is linear.

For training purposes  $\mathbf{O}$  should be as close to  $\mathbf{T}$  as possible, i.e.  $\|\mathbf{O} - \mathbf{T}\| = 0$ .

ELM theory [21] states that to achieve  $\|\mathbf{O} - \mathbf{T}\| = 0$ ,  $\mathbf{W}_i$  can be initialized with random value and  $\mathbf{W}_o$  computed as

$$\mathbf{W}_o = pinv(\mathbf{H}) * \mathbf{T} \quad (6)$$

where  $pinv(\mathbf{H})$  represents the generalized inverse of a matrix.

Once training is complete, ELM can be used for the purpose of regression or classification.

### 2.3. Related work

The HMM was first developed for speech processing [22], resulting in the two most successful HMM speech engines, HTK [13] and Sphinx [23]. Since then, HMMs have been applied extensively in numerous research studies and applications, including those involving handwriting, DNA, gestures, and computer vision.

In the HMM clustering literature, sequences are considered to be generated from a mixture of HMMs. The earliest work was presented by [24], in which a mixture of HMMs was regarded as a composite HMM. A new metric distance was devised between sequences using the log likelihood and clustered using hierarchical clustering.

Reference [25] extended this work to apply to electrocardiogram (ECG) data using a technique in which observations followed an auto-regressive model rather than a Gaussian mixture. Similarly, in [26] the log likelihood between the sequence and the model was used as the feature vector for the sequence.

To better choose the correct model and number of clusters for HMM clustering, [16] used the BIC. Their approach was not tested on real data and would require some modifications for practical application, as seen in Sec. 2.1.. The method used in this paper, while similar to theirs, has advantages. HTK is used to learn HMM parameters and handle time series with multiple features.

To date, wind forecasting approaches have assumed continuous HMMs, but in practice, a wind time series feature vector is simultaneously discrete (for wind direction) and continuous (for wind speed). The method proposed in this paper is able to handle this problem successfully.

### 3. WIND TIME SERIES FORECASTING USING HMM CLUSTERING AND ELM PREDICTION

This section presents a novel framework for wind time series forecasting. The basic idea is to incorporate data available from different locations in order to achieve better prediction. The framework first clusters the wind time series into groups of similar patterns and then uses data in the same group to train an ELM to improve the prediction result.

#### 3.1. HMM clustering using modified information criteria

Clustering, often known as unsupervised learning, allows objects possessing similar features to be grouped together. This paper presents a new method for clustering wind time series data. Each time series is modeled by an HMM, and clustering is based on the similarity between those models. The algorithm is given in Fig. 1.

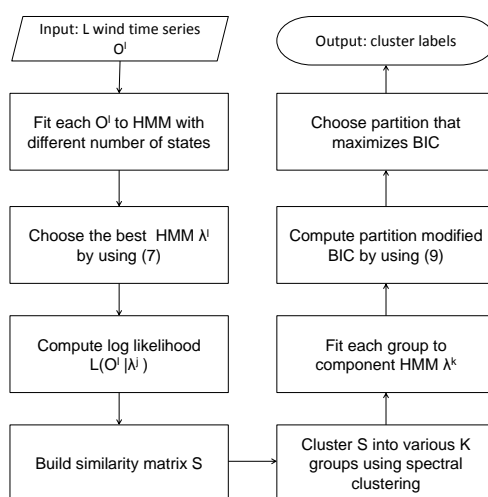


Figure 1: Flow chart of time series clustering using MIC HMM. This process removes most of the non-local data but keeps any non-local data that fall into the same cluster as the local data

In the first step, the algorithm searches for the best model for each sequence. Each sequence essentially consists of subsequences, each of which is regarded as a sample in HTK. The HMM is learned using the HTK toolbox. HInit is randomly initialized and the model is later refined by HRest.

The log likelihood of the sequence provided by each model is used to compute the BIC measurement from (7). In this paper, BIC is modified to better work with data from a discrete HMM with numerous observations:

$$MIC = \log\{P(X|\lambda, \hat{\theta})\} - \alpha d \tag{7}$$

where  $\alpha$  is the adjusted coefficient, which will be defined in Sec. 4.. The typical value of  $\alpha$  for a discrete HMM is 0.2.

The number of independent parameters in a discrete HMM is calculated using (8):

$$d = Q^2 + QM - Q - 1 \quad (8)$$

where  $Q$  represents the number of states and  $M$  represents the number of observations.

After the best model for each sequence is found, the log likelihood  $L(O^i|\lambda^j)$  is computed as the distance from sequence  $i$  to sequence  $j$ . The drawback of this step is the cost,  $O(N^2)$ , but for a small system, this is acceptable. This log likelihood then is used as a similarity between the two sequences  $i$  and  $j$ . Unfortunately, this likelihood is not symmetric and therefore not applicable for this clustering algorithm unless some transform is undergone. Reference [25] suggests several transforms, but for the current approach, the sum of two likelihoods produces satisfactory results.

The next steps involves finding the best partition by scanning the number of partitions from  $K_{min}$  to  $K_{max}$ . At each  $K$ , a spectral clustering algorithm [27] is used to achieve the partition. Each cluster found using spectral clustering then is modeled by an HMM using the same initialization as in step 1. Next,  $K$  component HMMs representing sub-systems are obtained. Finally, (9) is used to compute the BIC criteria, which are used to measure the quality of the configuration.

$$BIC = \sum_{i=1}^L \log \sum_{k=1}^K P_k * P(O_i|\lambda_k) - \beta(K + \sum_{k=1}^K d_k) \quad (9)$$

where  $P_k$  is the likelihood of data given cluster  $k$ . The membership is assumed crisp, so  $P_k = 1$  if sequence  $O_i$  is in cluster  $k$ , and  $P_k = 0$  otherwise.

Note that in (9), the first term generally increases as  $K$  increases because smaller clusters generally result in better HMMs; therefore, the sum of the log likelihood will be higher. On the other hand, the second term increases linearly with  $K$ . Therefore, BIC will reach the peak at some  $K_{best}$  and then decrease after that.

$\beta$  is an important factor to be defined. In this experiment,  $\beta$  depends on both  $\alpha$  and the number  $S$  of sub-sequences in each sequence:

$$\beta = \frac{\alpha}{1 + \log(S)} \quad (10)$$

### 3.2. Prediction using ELM

The prediction of wind speed  $a$  steps ahead is based on past samples from the target wind farms and the changes in wind speed at nearby wind farms caused by a meteorological events [28]. For the purpose of wind speed prediction, a specific ELM design is used. The structure of this ELM is described in Fig. 2 as follows: i) The number of ELM prediction inputs is based on the number  $n$  of past samples of wind speed used. When predicting with clustering information,  $m$  past samples from every other time series are appended to the  $n$  past samples of the considered time series. ii) The number  $F$  of hidden nodes is defined. Researchers [21] have claimed that the performance of ELM does not depend on the number of hidden nodes if it is large enough. Therefore, in Sec. 5. a small number of hidden nodes are defined but still retain good performance. And iii) Only one output neuron is used for the forecast result.

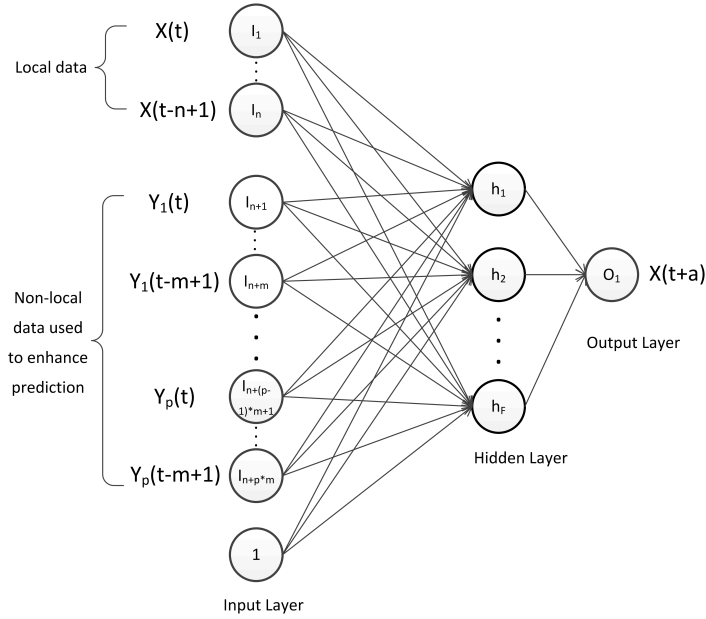


Figure 2: ELM predictor: Input layer serves as a tap delay line, hidden layer has nonlinear activation and linear output layer neuron estimates the predicted value [21]. In this figure,  $X(t)$  represents the local time series and  $Y_i(t)$ ,  $i = 1..p$  are all the time series data from non-local sites clustered together with local data via the process depicted in Fig. 1. Thus,  $i$  is the index of the  $i^{th}$  non-local site in the chosen cluster; and  $p$  is the number of sites that belong to the same cluster as the local site;  $n$  is the number of past samples of local time series used as ELM inputs;  $m$  is the number of past samples from every other time series in the same cluster;  $F$  is the number of hidden nodes; and  $a$  is the number of steps ahead into the future to be predicted.

#### 4. EXPERIMENTAL DESIGN

To verify the proposed approach, the real wind time series is clustered to define its dynamic properties. The work is completed in MATLAB and HTK. This combination is very efficient because MATLAB provides fast programming and analysis of the result, while HTK provides an engine with which to process the huge HMM computation. The experiment is conducted on a high-performance workstation running Ubuntu with Intel Xeon 6 cores 2.4 GHz CPU, 16 GB RAM. To speed up HMM parameter learning, the MATLAB parallel processing toolbox is utilized.

Before embarking upon a detailed description of the dataset, some important terms are clarified as follows. A site is a geographical location where data are collected. An observation is a measurement of a wind feature at a specific time and location. An observation has some features, including speed and direction. A sequence or time series is the collection of observations at a location over time. A subsequence is a sequence whose observation is limited to one year. In this paper, each site had 35 subsequences corresponding to each year. A cluster is a group of sites that share a similar HMM model.

The wind data set was obtained from the NOAA. The wind time series dataset used in this paper was collected from 15 sites around Vichy, MO, such as St. Louis, MO, Chicago, IL and Denver, CO. Their locations are mapped in Fig. 5.

Data were collected from 1973 to 2010, and measurements were supposed to have been taken



every hour. In total, the number of time samples for each sequence is a few hundred thousand. These are not averaged together by site or time. The duration of data collection is longer than that in many similar wind data studies, but it is appropriate because the clusters vary slowly. Thus, such long periods can also be used in the presented approach. There are 11 temporal features for each observation, and wind speed and wind direction are being the two most important. While the distribution of wind direction is fairly regular from 10 to 360 degrees, the wind speed distribution appears inconsistent, especially during gusts when the top speed is much higher than the average.

With such a long sequence of hundreds of thousands of observations, calculating the model for each sequence using an EM algorithm would be time consuming. This problem was addressed by dividing the wind time series at each location into smaller sequences according to each year. The sites with incomplete data for the entire period from 1973-2010 were not considered. In all, there were 35 sub-sequences for each location. From a speech processing perspective, each sub-sequence is regarded as a sample of a word or a sentence that can be used to train an HMM corresponding to that word or sentence.

Over years of observation for data collection, many entries in the sequences, both for wind speed and wind direction, were missed. In those instances, the missing entry was populated with the nearest available observation. Moreover, the time series were resampled into paces of one hour by averaging the observations that fell between the two consecutive hours.

The temporal features in the sequences also posed some problems. While in reality, wind direction and speed values are continuous, in the data set, wind direction was recorded as a discrete value in multiples of 10s. This has caused continuous Gaussian mixture HMM approaches, such as in [24,25], with failure to represent the sequence.

A discrete HMM is used by discretizing the wind speed using a histogram method. Wind speed histogram bins with speeds less than 2m/s are combined with adjacent bins, and those with speeds greater than 30m/s are stored in the same bin. Combining the two discrete wind speed and direction bins yields the observation symbol, with each value corresponding to an observation in the sequence. The dataset contains 2597 different observations.

Following the proposed framework, to find the best model for each sequence, the number of states in the HMM varies between 1, 5, 10, 15, 20, 25, and 30 and the HInit from HTK is run to compute the HMM and the log likelihood of each sequence for each model.

The result of the likelihood is depicted in Fig. 3(a). As illustrated, the log likelihood increases with the size of the model. The BIC is computed using 7 for each model. The value of  $\alpha$  changes from .1 to 1 and the value that BIC shows a clear peak is chosen. The result is plotted in Fig. 3b with  $\alpha = .2$ .

As indicated in Fig. 3(b), when the number of states is small, the log likelihood is dominant in the BIC, which explains the initial increase in the BIC. However, as the number of states increases, the penalty for complex configuration becomes dominant in the BIC, resulting in BIC reductions. For all of the time series, the BIC peaks after a certain number of states. Fig. 3(b) shows that this peak occurs at  $Q = 10$  for all sequences except for sequence 11. Therefore, an HMM with a size of 10 is used in later steps as the model for deciding the optimum number of clusters.

The dataset itself is very large, but the number of sequences is fairly small (15). Therefore, the number of clusters should be between  $K_{\min} = 2$  and  $K_{\max} = 7$ . Fig. 4 depicts the result of the BIC measurement for different partitions.

Fig. 4 indicates that the BIC is highest when  $K = 2$  and decreases as  $K$  increases. Therefore, the ideal number of clusters is 2. The clustering result appears in Fig. 5, where sites in the same

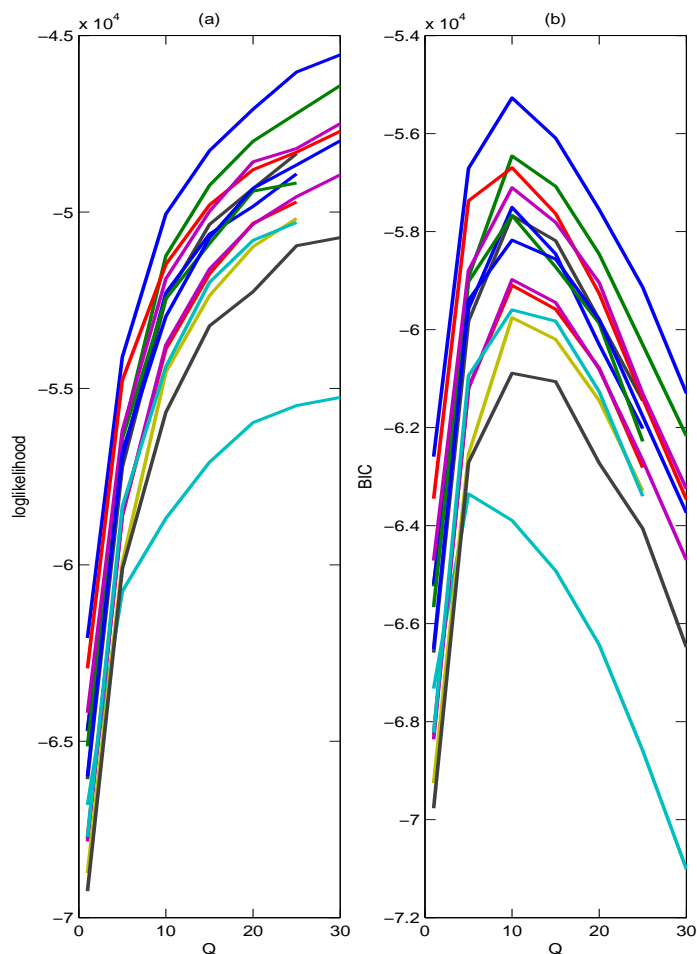


Figure 3: a) Log likelihood and b) BIC of the time sequences with the best estimating HMM plotted vs. the number of states in the HMM. Note that log likelihood increases with the number of states but BIC peaks at an intermediate number of states. Each time series is represented by different color (best viewed in color).

cluster have the same shape.

The HMM parameter learning complexity is  $O(Tx|Q|^2)$ , where  $T$  is the length of the sequence and  $Q$  is number of states. With this large dataset, the average time required to learn the best HMM for each site is 40 minutes, and the total time required to learn the best partition is 3 hours. This is still better than other methods, in which such learning is infeasible due to the large memory footprint. Moreover, the learning only has to be run once.

The sites on the right of the map belong to the same group and share the same dynamic wind properties. The site of particular interest in this work is Vichy, MO because the original goal of obtaining this dataset is to use the data from surrounding locations to help predict the wind pattern nearby. This clustering result shows that data from other sites in this group can be used to facilitate wind prediction in Vichy, MO as discussed in next section.

The clustering results can be leveraged for wind forecasting. This paper only forecasts the wind speed, but the method can be extended easily to other wind features, such as wind direction. Many

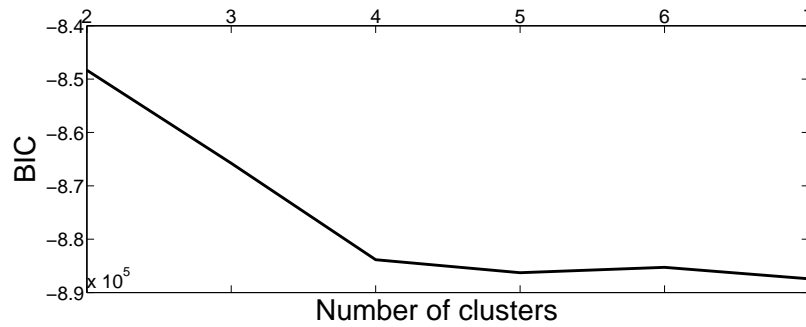


Figure 4: Cluster validity using BIC. After clustering into  $k$  clusters using spectral clustering, the partition BIC calculated using (9) was used as a cluster validity index. Higher index indicates better partitioning.



Figure 5: Result of clustering wind location: sites in the pin-shape correspond to the first cluster; sites in the balloon-shape correspond to the second cluster.

of the available wind forecasting approaches only use data collected from a single location. Making use of available data collected from sites belonging to the same clusters can enhance the prediction results, as shown in this section.

A single step ahead scheme is used for prediction, which means the wind speed is predicted one hour ahead. Once the predicted value is actually observed, the observed value was used as a feature in the next prediction. The wind time series is arranged into the input matrix as discussed in section 3.2.. Before proceeding with the prediction, the data is normalized to the range of  $[-1,1]$  as required by ELM.

## 5. FORECAST RESULTS AND DISCUSSION

Four experiments demonstrate the superiority of wind forecasting with clustering over forecasting without clustering and other forecasting methods. Among the 15 sites with available data, Vichy, MO is chosen as the location of the experiment and considered to be local, while the other 8 sites (data from Denver, CO excluded because it is too far from Vichy, MO) in the same cluster as Vichy,

MO is considered to be a group. All of the data are divided into seasons. 20% of the final year (432 hours, with data taken equally from each season) is used for testing. The rest of the data from all of previous years constitutes the training set (see Table 1). Other parameter settings are listed in Table 1. All ELMs have sigmoid activation function in the hidden layers. All reported results are averaged after 10 runs of ELM regression to take into account the randomness of input weight initialization.

Exp.	$F$	$n$	$m$	Train	Test
Fig. 6	10, 100, 1000	1..100	0	Vichy, MO data, autumn 1975..2008+ 80% of 2009	Vichy, MO, autumn 20% of 2009
	100	1..100	1..20	Group data, autumn 1975..2009 (80% of 2009 for local data)	
Table 2	100	50	0	Vichy, MO data, autumn 1975..2008+ 80% of 2009	Vichy, MO, autumn 20% of 2009
	100	30	4	Group data, autumn 1975..2009 (80% of 2009 for local data)	
Table 3	100	50	0	Vichy, MO data; 4 seasons; 80% of 2009 or 3, 5, 10, 20, 35 previous years	Vichy, MO, 4 seasons, 20% of 2009
	100	30	4	group data; 4 seasons; 1, 3, 5, 10, 20, 35 previous years (80% of 2009 for local data)	
Fig. 7	100	30	4	Group data, spring 1975..2009 (80% of 2009 for local data)	Vichy, MO, spring 20% of 2009
	1000	100	40	Group data, spring 1975..2009 (80% of 2009 for local data)	

Table 1: Forecast parameter settings in 4 experiments

The first experiment involves defining the best ELM configuration for prediction with and without clustering information. The parameters to be determined are  $F$ , the number of hidden nodes, and  $n$ , the number of past wind data samples from Vichy, MO used as ELM inputs. When predicting with clustering information, one additional parameter,  $m$ , the number of past samples from each of the locations in the same cluster as Vichy, MO, is used as extra ELM input and has to be estimated.

Fig. 6(a) shows the error of the forecast of wind speed in autumn. For prediction with only local data,  $F$  is given three representative values 10, 100 and 100 while  $n$  varies from 1 to 100. As Fig. 6(a) indicates, when the number of hidden neurons is small (i.e., about 10) the ELM prediction is unstable. In fact, the root mean square error (RMSE) increases with the number of samples used for prediction. However, as the number of hidden neurons increases, the performance of the prediction become stable. At  $F=100$ , the ELM predictor performs very well across a wide range of hidden nodes and the smallest error is obtained when  $n=30$ .

For prediction with clustering information,  $F$  is fixed at 100 while  $n$  and  $m$  vary from 1 to 20 and 1 to 100, respectively. Fig. 6(b) shows that the lowest error prediction is reached when  $m=4$  and  $n=30$ .

In the second experiment, the performance of the proposed approach is compared with that of other feed forward neural networks such Backpropagation, RBF, ANFIS, and the persistence method. Backpropagation and RBF are implemented in the MATLAB Neural Network Toolbox, and ANFIS is implemented in the MATLAB Fuzzy Logic Toolbox.

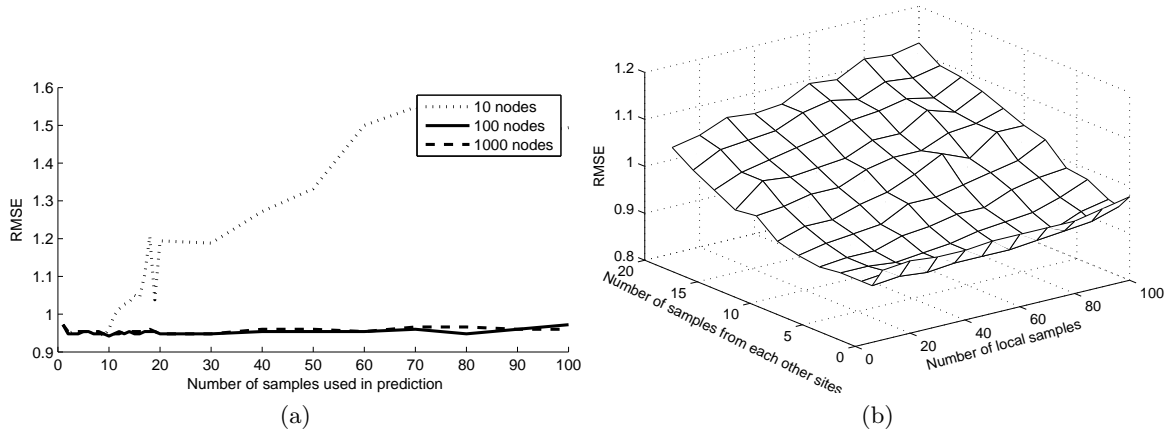


Figure 6: ELM forecasting configuration defined with various numbers of hidden nodes and input nodes. a) Prediction without clustering information. ELM performance is stable as long as  $F$  is larger than 100. b) Best number of past samples in the local site defined along with number of past samples in each other site in the same cluster for shared-cluster prediction.

	RMSE (m/s)	MAE (m/s)	MAPE	Training time (s)
Persistence (autumn 2009 data)	1.00	.76	16.27	N/A
ANFIS (2 inputs, autumn 2009 data)	1.05	.75	16.15	1.70
RBF (5 inputs, spread factor =10 with data from last 5 autumns)	.95	.72	15.94	165
Backpropagation (30 inputs, 100 hidden nodes)	1.59	1.14	24.32	6300
ELM local 30 inputs, 100 hidden nodes)	.95	.72	16.08	.24
ELM group (30 local inputs, 4 inputs from each of the sites in the same cluster, 100 hidden nodes )	.90	.70	15.20	.27

Table 2: Performance comparisons among RBF, backpropagation, ANFIS, persistence method, AR and ELM approach

As Table 2 indicates, the proposed approach performed best among the compared algorithms across all of the error performance indices: root mean square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE). The time requirement is also small, just .24s for local data and .27s for group data.

The performance of the ELM group approach is also tested in various seasons and years. Table 3 shows the wind forecast in spring, summer, autumn and winter of 2009. The number of years of data used to train the ELM was 1, 3, 5, 10, 20, and 35.

Different seasons affects the performance of ELM prediction differently. It performs best in spring, for which the RMSE error is lowest at .85 with clustering information. The prediction with clustering information performs better than without it, across all seasons.

RMSE	Years	1	3	5	10	20	35
Autumn	Vichy, MO data	.96	.95	.95	.94	.95	.95
	Group data	.94	.90	.89	.89	.90	.90
Summer	Vichy, MO data	1.08	1.06	1.06	1.06	1.06	1.06
	Group data	1.08	1.02	1.03	1.01	1.01	1.01
Spring	Vichy, MO data	.92	.92	.91	.90	.90	.90
	Group data	.90	.86	.86	.86	.86	.86
Winter	Vichy, MO data	1.40	1.37	1.37	1.36	1.36	1.36
	Group data	1.35	1.29	1.28	1.29	1.28	1.29

Table 3: Forecast performance for various seasons and years.

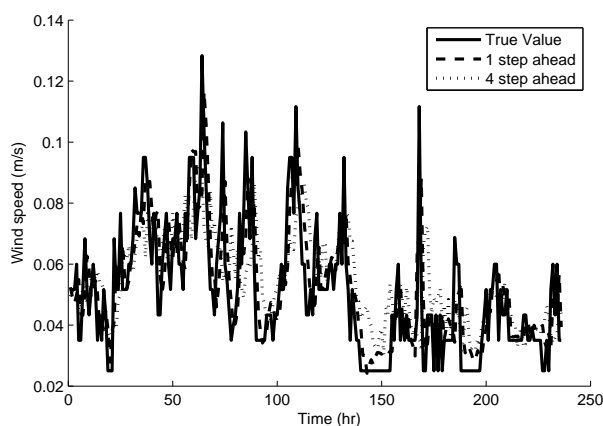


Figure 7: Comparison of forecasting and ground truth over 240-hour duration.

Finally, the wind speed is forecast and compared with the true value. Forecasting is performed for both a single step ahead and up to four steps ahead with clustering information. Multiple step ahead prediction can be performed by either of two methods: In the direct method,  $n$  past samples were used to directly predict the wind speed 4 hours ahead while in the indirect method,  $n$  past samples were used to predict the wind speed one step ahead, and this prediction is recursively used

to predict the next step until prediction had been made up to 4 hours ahead. Multiple simulations are performed using two methods; the direct method performed better and therefore is used in this paper.

To increase the performance,  $F$  was set at 1000. Similarly to one step ahead prediction, experiments are conducted with different numbers of past samples from the local site and clustering sites to determine that the ELM with  $n = 100$  and  $m = 40$  is the optimal configuration. The wind speed is forecast using this ELM configuration.

Fig. 7 depicts the 1 hour and 4 hour ahead forecast values and compares with the true value in 240 hour observation. The RMSE of 4 step ahead prediction is 1.11 m/s while that of 1 step ahead is 0.86 m/s.

One potentially fruitful avenue for future investigation would be to extend the presented technique to online learning. Clustering is widely and correctly considered as a tool for online learning. However, in this application the relationships between clusters vary slowly over time. Therefore, the ELM and its rapid adaptation capability make online learning an intriguing possibility.

## 6. CONCLUSION

This paper presents a new method for forecasting wind time series data. By combining clustering techniques in HMM and the fast but accurate ELM regression, the proposed method successfully improves forecasting accuracy. The method can handle a large amount of data by leveraging the power of the HTK engine and the analytic solution for training the ELM. In the future, HMM processing will be accelerated using GPU clusters to further reduce the amount of time required for learning HMM parameters.

## REFERENCES

- [1] G. Venayagamoorthy, K. Rohrig, and I. Erlich, "One step ahead: short-term wind power forecasting and intelligent predictive control based on data analytics," *Power and Energy Magazine, IEEE*, vol. 10, no. 5, pp. 70–78, 2012.
- [2] B. Brown, R. Katz, and A. Murphy, "Time series models to simulate and forecast windspeed and wind power," *Journal of Climate and Applied Meteorology*, vol. 23, pp. 1184–1195, 1984.
- [3] S. Li, T. A. Haskew, K. A. Williams, and R. P. Swatloski, "Control of dfig wind turbine with direct-current vector control configuration," *Sustainable Energy, IEEE Transactions on*, vol. 3, no. 1, pp. 1–11, 2012.
- [4] G. Box and G. Jenkins, *Time series analysis, forecasting and control*. 1976.
- [5] S. S. Soman, H. Zareipour, O. Malik, and P. Mandal, "A review of wind power and wind speed forecasting methods with different time horizons," in *North American Power Symposium (NAPS)*, pp. 1–8, 2010.
- [6] P. Werbos, "Brain-like prediction: New statistical foundations for prediction in the face of real world complexity." IEEE Latin American Summer School on Computational Intelligence lecture, 2009.
- [7] E. Saad, D. Prokhorov, and I. Wunsch, D.C., "Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks," *Neural Networks, IEEE Transactions on*, vol. 9, no. 6, pp. 1456–1470, 1998.

- [8] A. Sfetsos, "A comparison of various forecasting techniques applied to mean hourly wind speed time series," *Renewable Energy*, vol. 21, no. 1, pp. 23–35, 2000.
- [9] S. Li, D. C. Wunsch, E. A. O'Hair, and M. G. Giesselmann, "Using neural networks to estimate wind turbine power generation," *Energy conversion, IEEE transactions on*, vol. 16, no. 3, pp. 276–282, 2001.
- [10] K. Rohrig and B. Lange, "Application of wind power prediction tools for power system operations," in *Power Engineering Society General Meeting, IEEE*, 2006.
- [11] C. W. Potter and M. Negnevitsky, "Very short-term wind forecasting for Tasmanian power generation," *Power Systems, IEEE Transactions on*, vol. 21, no. 2, pp. 965–972, 2006.
- [12] P. Ailliot and V. Monbet, "Markov-switching autoregressive models for wind time series," *Environmental Modelling & Software*, vol. 30, pp. 92–101, Apr. 2012.
- [13] "Htk speech recognition toolkit," in <http://htk.eng.cam.ac.uk/>.
- [14] H. Akaike, "A new look at the statistical model identification," *Automatic Control, IEEE Transactions on*, vol. 19, no. 6, pp. 716–723, 1974.
- [15] G. E. Schwarz, "Estimating the dimension of a model," *Annals of Statistics*, pp. 461–464, 1978.
- [16] C. Li and B. Gautam, "A Bayesian approach to temporal data clustering using hidden Markov models," in *International Conference on Machine Learning*, pp. 543–550, 2000.
- [17] A. Biem, J.-Y. Ha, and J. Subrahmonia, "A Bayesian model selection criterion for hmm topology optimization," in *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, vol. 1, pp. 989–992, May 2002.
- [18] J. Hu and B. Ray, "In interleaved HMM/DTW approach to robust time series clustering," tech. rep., IBM, 2006.
- [19] R. Xu and D. Wunsch, *Clustering*. Wiley-IEEE Press, 2009.
- [20] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 42, no. 2, pp. 513–529, 2012.
- [21] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, 2006.
- [22] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
- [23] "Sphinx," in <http://cmusphinx.sourceforge.net/>.
- [24] P. Smyth, "Clustering sequences with hidden Markov models," in *Advances in Neural Information Processing Systems*, pp. 648–654, MIT Press, 1997.
- [25] A. Panuccio, M. Bicego, and V. Murino, "A hidden Markov model-based approach to sequential data clustering," pp. 734–742, Springer, 2002.
- [26] M. Bicego, V. Murino, and M. A. Figueiredo, "Similarity-based classification of sequences using hidden Markov models," *Pattern Recognition*, vol. 37, no. 12, pp. 2281–2291, 2004.



- [27] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: analysis and an algorithm," in *NIPS*, pp. 849–856, 2001.
- [28] R. Schlueter, G. Sigari, and A. Costi, "Wind array power prediction for improved operating economics and reliability," *Power Systems, IEEE Transactions on*, vol. 1, no. 1, pp. 137–142, 1986.

*Received on November 20 - 2014*

*Revised on November 30 - 2014*